Data Science Desktop Survival Guide

Graham Williams
Togaware
Graham.Williams@togaware.com

2021-06-24

Preface

20210308

"The enjoyment of one's tools is an essential ingredient of successful work." *Donald E. Knuth*

This book is undergoing conversion from LaTeX to Bookdown. It is a work in progress and there remains numerous glitches. Please bare with us.

Welcome to the world of Artificial Intelligence (AI), Machine Learning, and Data Science. Data Science has come to describe many different activities that are driven by data utilising technological advances in AI and Machine Learning, amongst others. Indeed, we can think of Data Science as an ecosystem of both technology and experiences, shared through the freedom of open source software that implements tools for AI and Machine Learning. Open source software gives us the freedom to do as we want with the software, with very few, and ideally no, restrictions, except the requirement to maintain our freedoms.

The aim of this book is to gently guide the novice along the pathway to Data Science, from data processing through Machine Learning and to AI. I hope I can share the excitement of a fun and productive environment for exploring data with a focus on the R language as our platform of choice. R remains the most flexible and powerful language developed specifically for the analysis of data.

This book provides a guide to the many different regions of the R platform, with a focus on doing what is required of the Data Scientist. It is comprehensive, beginning with basic support for the novice Data Scientist, moving into recipes for the variety of analyses we may find ourselves needing.

On completing an install of R (which may take only a few minutes) you are ready to explore your data and beyond. All of the different types of data analyses are covered in this book, including basic data ingestion, data cleaning and wrangling, data visualisation, modelling and evaluation in order to discover new knowledge from our data. Tools for developers of systems to be deployed are well represented.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

Data Science Desktop Survival Guide

Graham Williams
Togaware
Graham.Williams@togaware.com

2021-06-24

Preface

20210308

"The enjoyment of one's tools is an essential ingredient of successful work." *Donald E. Knuth*

This book is undergoing conversion from LaTeX to Bookdown. It is a work in progress and there remains numerous glitches. Please bare with us.

Welcome to the world of Artificial Intelligence (AI), Machine Learning, and Data Science. Data Science has come to describe many different activities that are driven by data utilising technological advances in AI and Machine Learning, amongst others. Indeed, we can think of Data Science as an ecosystem of both technology and experiences, shared through the freedom of open source software that implements tools for AI and Machine Learning. Open source software gives us the freedom to do as we want with the software, with very few, and ideally no, restrictions, except the requirement to maintain our freedoms.

The aim of this book is to gently guide the novice along the pathway to Data Science, from data processing through Machine Learning and to AI. I hope I can share the excitement of a fun and productive environment for exploring data with a focus on the R language as our platform of choice. R remains the most flexible and powerful language developed specifically for the analysis of data.

This book provides a guide to the many different regions of the R platform, with a focus on doing what is required of the Data Scientist. It is comprehensive, beginning with basic support for the novice Data Scientist, moving into recipes for the variety of analyses we may find ourselves needing.

On completing an install of R (which may take only a few minutes) you are ready to explore your data and beyond. All of the different types of data analyses are covered in this book, including basic data ingestion, data cleaning and wrangling, data visualisation, modelling and evaluation in order to discover new knowledge from our data. Tools for developers of systems to be deployed are well represented.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

About this Book

20210314 This book has been a work in progress since 1995, just as Data Science continues to develop and expand into our lives. Every section (eventually) will begin with a date that indicates the currency of the section—when it was last reviewed and/or updated.

Since beginning the survival guide books in 1995 they have grown in all kinds of directions. My original aim was to capture useful notes for the varied and many common tasks we find ourselves doing as data scientists (or *data miners* back then). I structured the book as one page nuggets of information. That is, each section within a chapter targeted a single printed page, and focused on a single task. This was the origin of my OnePageR Desktop Survival Guides. It seems to have worked well over the years, from my personal use and your feedback. This material has also lead to the publication of two books.

Readers are invited to send corrections, comments, suggestions, and updates to me at Graham.Williams@togaware.com. Your feedback is most welcome and will be acknowledged within the book.

A **pdf** version of this book is available for a small financial donation which goes towards supporting the development and availability of the book. Please visit Togaware for the details. The **html** version contains the same material and remains freely available from Togaware

Production

This book is produced using bookdown. Emacs is used to edit the text. Many will be using RStudio to edit their bookdown documents, which is a generally more friendly environment and is the environment of choice for bookdown support. I've used Emacs since 1985 and as a fully extensible "kitchen-sink" type of editor, it has served me well for over 35 years, despite numerous flirtations with "better" editors over my career. RStudio and Visual Studio Code come close.

Bookdown is an rmarkdown based platform for intermixing text with executable code (like Python, R and Shell code blocks). Rmarkdown itself utilises the simple markdown syntax to markup the sections of a document. After running knitr over the rmarkdown material a markdown document is produced.

Pandoc is then utilised to produce **html** which is published on the Web. For the **pdf** output pandoc utilises LaTeX, converting the markdown into LaTeX markup, with xetex used to then convert that to **pdf**.

All these tools are open source software and available on multiple platforms.

Many books are today being written using bookdown. Examples include Data Science at the Command Line (github); Efficient R Programming (github).

What's In A Name

GNU/Linux refers to the GNU environment and the GNU and other applications running in that environment on top of the Linux operating system kernel.

Ubuntu and its underlying base distribution *Debian* are complete repository based distributions which include many applications pre-built for the particular choice of operating system kernel. The repositories house pre-built packages ready to be installed.

X Window System is the common windowing system used in Ubuntu and is a separate complementary component to the operating system itself.

Microsoft Windows (or *MS/Windows* and less informatively just *Windows*) usually refers to the whole of the popular operating system, from kernel to applications, irrespective of which version of Microsoft Windows is being run, unless the version is important. Microsoft Windows is one of many windowing systems and came on to the screen rather later than the pioneering Apple Macintosh windowing system and the Unix windowing systems. We will refer to MS/Windows version 10 as the *last* release of this Microsoft operating system, which going forward has snapshot releases rather than new versions.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

Organisation of the Book

20200218 Individual chapters aim to be a standalone reference using a one-pager style whereby each individual page aims to be a standalone guide on a specific point or topic. This one-page concept comes from my OnePageR book for data science, artificial intelligence, and machine learning and has been quite successful there.

So sit back and enjoy the freedom and liberty that comes with free software. Also, please consider becoming an active part of the community that is making computers and the applications they run and the data they collect a benefit to society world wide, rather than a privilege of the few.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

Acknowledgements

20210223 There are many people to thank for sharing these tools, their knowledge, and their encouragement in many different ways. Indeed, the open source and especially the R and now the tidyverse communities are chracterised by their willingness to share for the good of us all, and many folk have also contributed directly and indirectly to this book through their sharing. Their contributions are acknowledged throughout the book, but there are always gaps. To all who share openly, thank you. I have learned so much from this community over more than 30 years.

Your support for maintenance of this book is always welcome. Financial support is used to contribute toward the costs of running the servers used to make this book available. Donations can be made through PayPal at https://onepager.togaware.com/

The following are acknowledged for their recent support of the book: David Montgomery, Arivarignan Gunaseelan, Clemens Kielhauser, and Ricardo Scotta.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

Waiver and Copyright

20210314 Writing of this book began in 1995 and continues to be updated to this day. It is copyright by Togaware and licensed under the Creative Commons Attribution-ShareAlike 4.0 license. It is made freely available to serve as a useful resource for users of Free and Open Source Software in the hope that it serves as a useful resource.

The procedures and applications presented in this book have been included for their instructional value. They have been tested at various times over the years but are not guaranteed for any particular purpose. We also note that functionality of different packages can change over time and whilst we make an effort to update the material the sheer volume presents a challenge. The publisher, togaware.com, does not offer any warranties or representations, nor does it accept any liabilities with respect to the programs and applications.

Copyright © Togaware Pty Ltd

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

1 Data Science

20200104

> Science is analytic description, philosophy is synthetic interpretation. Science wishes to resolve the whole into parts, the organism into organs, the obscure into the known. (Durant 1926)

Today we live in a *data* rich, *information* driven, *knowledge* strained, and *wisdom* scant world. Data surrounds us every where we look. Data describes every facet of everything we know and do. Today we are capturing and storing more data electronically (i.e., digitising the data) at a rate we humans have never before been capable of. We have so much data available and even more yet to be digitised from the world around us, and so much of the digitised data yet to be analysed. Most of our work today is about data, and perhaps it always has been.

Data science is a broad church capturing the endeavour of analysing *data* and *information* by appropriately applying an ever changing and vast collection of techniques and technology to deliver *knowledge* to be synthesised into *wisdom*. The role of a data scientist is to perform the transformations that make sense of it all in an evidence based endeavour delivering the knowledge deployed with wisdom.

The data scientist acts with humanity and philosophy to synthesise knowledge into wisdom whereby we thrive to *resolve the obscure into the known*. It is this synthesis that delivers the real benefit of the science—whether that benefit be for business, industry, government, environment, but always for humanity.

A data scientist brings together a suite of skills to solve problems in a data driven way. It is not the only way to solve problems, and indeed may not even be the best way to solve problems, but it is the current best way to deal with many real world use cases today. In the future we will see more focus on knowledge representation and causal reasoning, but for now we can do a lot with data.

References

Durant, Will. 1926. *The Story of Philosophy*. 2012th ed. Simon; Schuster.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

1.1 Data Wrangling

Data wrangling is defined by Wikipedia (http://en.wikipedia.org/wiki/Data_wrangling) as *the process of manually converting or mapping data from one "raw" form into another format that allows for more convenient consumption of the data with the help of semi-automated tools.*

Those who do this are data wranglers.

In this introductory chapter we present the concepts of data science, Analytics, and the role and toolkit of the data scientist. We identify the progression from data Technician, through data analyst and data miner, to data scientist. The remaining chapters of this book then provide a guide to the key tools for doing data science in a hands-on manner through the most powerful software system for doing data science called R (R Core Team 2021). R is today supported by all of the major vendors and is the backbone of the emerging data science platforms in the cloud.

References

R Core Team. 2021. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

1.2 Data Scientist

A data scientist is an exceptional person. They bring to a task a specific collection of computer skills using a variety of tools. They also have particularly strong intuitions about how to resolve the whole into its components. They can then explore, visualise, analyse, and model the components to then synthesise new understandings into a new whole. With a desire and hunger for continually learning the data scientist will always be on the lookout for opportunities to improve how things are done and to change the current state of the world for the better.

Finding all the requisite technical skills and motivation in one person is rare and so true data scientists are scarce and the demand for their services continues to grow as we find ourselves with more data every day.

As these skills become critical to all human activity we will find new platforms emerging that will support the data scientist. Platforms that provide open and extensible support of data ingestion, data fusion, data cleansing, data wrangling, data visualisation, data analysis, and model building will begin to appear and will allow data scientists to be more efficient and productive. Such frameworks will also serve as the pathway for new data scientists into the world of data science.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

1.3 Data Science Through Doing

This book takes a seriously hands on approach to doing data science. Concepts are introduced but usually with an example of the concept with some code. We use R as our language to share these concepts and demonstrate how we execute the concepts in practice. Through this guide new R commands will be introduced. The reader is encouraged to review the command's documentation and understand what the command does. Help is obtained using the `?` command as in:

```
?read_csv
```

Documentation on a particular package can be obtained using the `help=` option of `library()`:

```
library(help=tidyverse)
```

To learn effectively you are encouraged to run R (e.g., RStudio or Emacs with ESS mode) and to replicate the commands. Check that the output you get is the same and that you understand how it is generated. Try some variations. Explore.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

2 Introducing R

Data scientists write programs to ingest, fuse, clean, wrangle, visualise, analyse, and model data. Programming over data is a core task for the data scientist. We will primarily use R (R Core Team 2021) and in particular the tidyverse as our programming language and assume basic familiarity of R as may be gained from the many resources available on the Intranet, particularly from https://cran.r-project.org/manuals.html.

The development of the tidyverse has been instrumental in bringing R into the modern data science era and the resources provided by RStudio and the tidyverse community are extensive. In particular, as you develop your data analyses, be sure to have the RStudio cheatsheets for the tidyverse in front of you. You will find them invaluable. Visit https://rstudio.com/resources/cheatsheets/.

Programmers of data develop sentences or *code*. Code instructs a computer to perform specific tasks. A collection of sentences written in a language is what we might call a *program*. Through *programming by example* and *learning by immersion* we will share programs to deliver insights and outcomes from our data.

R is a large and complex ecosystem for the practice of data science. There is much freely available information on the Internet from which we can continually learn and borrow useful code segments that illustrate almost any task we might think of. We introduce here the basics for getting started with R, libraries and packages which extend the language, and the concepts of functions, commands, and operators.

References

R Core Team. 2021. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of*

Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

2.1 Tooling For R Programming

20210103 We will first need to have available to us the computer software that interprets our programs that we write as sentences in the R language. Usually we will install this software, called the R Interpreter, on our own computer and instructions for doing so are readily available on the Internet. The R Project is a good place to start. Most GNU/Linux distributions freely provide R packages from their application stores and a commercial offering is available from Microsoft who also provide direct access within SQL Server and through Cortana Analytics on the Azure cloud. Now is a good time to install R on your own computer if that is required.

The open source and freely available RStudio software is recommended as a modern integrated development environment (IDE) for writing R programs. It can be installed on common desktop computers or servers running GNU/Linux, Microsoft/Windows, or Apple/MacOS.

A browser version of RStudio also allows a desktop to access R running on a back-end cloud server. This version of RStudio runs on a remote computer (e.g., a server in the cloud) and provides a graphical interface presented within a web browser on our own desktop. The interface is much the same as the desktop version but all of the commands are run on the server rather than on our own desktop. Such a setup is useful when we require a powerful server on which to analyse very large datasets. We can then control the analyses from our own desktop with the results sent from the server back to our desktop whilst all the computation is performed on the server itself which may be a considerably more powerful computer than our usually less capable personal computers.

An alternative cloud server setup is to connect to a remote cloud server using X2Go as a remote desktop protocol.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

2.2 Introducing RStudio

20210103 The RStudio interface includes an R `Console` on the left through which we directly communicate with the R software. The window pane on the top right provides access to the `Environment` within which R is operating. This includes data and datasets that are defined as we interact with R. Initially it is empty. A `History` of all the commands we have asked R to run is available on another tab within the top right pane.

The bottom right pane provides direct access to an extensive collection of `Help`. Another tab within this same pane provides access to our local `Files`. Other tabs provide access to `Plots`, `Packages`, and a `Viewer` to view documents that we might generate.

In the `Help` tab displayed in the bottom right pane of Figure @ref(intro:fig:rstudio_startup), for example, we can see links to the R documentation. The manual titled An Introduction to R within the `Help` tab is a good place to start if you are unfamiliar with R. There are also links to learning more about RStudio—these are recommended if you are new to RStudio.

{#intro:fig:rstudio_startup} *The initial layout of the RStudio window panes showing the* `Console` *pane on the left with the* `Environment` *and* `Help` *panes on the right.*

Your donation will support ongoing development and give you access to the ***PDF version of this book****. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

2.3 Editing Code

Often we will be interacting with R by writing code and sending that code to the R Interpreter so that it can be run (locally or remotely). It is always good practice to store this code into a file so that we have a record of what we have done and are able to replicate the work at a later time. Such a file is called an R Script.

We can create a new R Script by clicking on the first icon on the RStudio toolbar and choosing the first item in the resulting menu as in Figure @ref(intro:fig:rstudio_startup_editor). A keyboard shortcut is also available to do this: `Ctrl+Shift+N` (hold the `Ctrl` and the `Shift` keys down and press the `N` key).

A sophisticated file editor is presented as the top left pane within RStudio. The tab will initially be named `Untitled1` until we actually save the script to a file on the disk. When we do so we will be asked to provide a name for the file.

The editor is where we write our R code and compose the programs that instruct the computer to perform particular tasks. The editor provides numerous features that are expected in a modern program editor. These include syntax colouring, automatic indentation to improve layout, automatic command completion, interactive command documentation, and the ability to send specific commands to the R Console to have them run by the R Interpreter.

{#intro:fig:rstudio_startup_editor} *Ready to edit R scripts in RStudio using the* `Editor` *pane on the top left created when we choose to add a new R Script from the toolbar menu displayed by clicking the icon highlighted in red.*

Your donation will support ongoing development and give you access to the ***PDF version of this book****. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

2.4 Executing Code

REVIEW To send R commands to the R Console we can use the appropriate toolbar buttons. One line or a highlighted region can be sent using the `Run` button. Having opened a new R script file we can enter the commands below. The four commands make up a program which begins with an install.packages() to ensure we have installed two requisite software packages for R. The second and third commands use library() to load those packages into the current R session. The fourth command produces a plot of the weatherAUS dataset. This dataset contains daily observations of weather related variables across some 50 weather stations in Australia over multiple years.

```
install.packages(c("ggplot2", "rattle"))
library(ggplot2)
library(rattle)
qplot(data=weatherAUS, x=MinTemp, y=MaxTemp)
```

qplot() allows us to quickly code a plot. Using the weatherAUS dataset we plot the minimum daily temperature (`MinTemp`) on the x-axis against the maximum daily temperature (`MaxTemp`) on the y-axis. The resulting plot is a scatter plot as we see the plot in the lower right pane.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

2.5 RStudio Review

REVIEW Figure @ref(fig:intror:rstudio_weather_scatterplot) shows the RStudio editor as it appears after we have typed the above commands into the R `Script` file in the top left pane. We have sent the commands to the R `Console` to have it run by R. We have done this by ensuring the cursor within the R `Script` editor is on the same line as the command to be run and then clicking the `Run` button. We will notice that the command is sent to the R `Console` in the bottom left pane and the cursor advances to the next line within the R `Script` editor.

After each command is run any text output by the command is displayed in the R `Console` whilst graphic output is displayed in the `Plots` tab of the bottom right pane.

This is now our first program in R. We can now provide our first observations of the data. It is not too hard to see from the plot that there appears to be quite a strong relationship between the minimum temperature and the maximum temperature: with higher values of the minimum temperature recorded on any particular day we see higher values of the maximum temperature. There is also a clear lower boundary that might suggest, as logic would dictate, that the maximum temperature can not be less than the minimum temperature. If we were to observe data points below this line then we would begin to explore issues with the quality of the data.

As data scientists we have begun our observation and understanding of the data, taking our first steps toward immersing ourselves in and thereby beginning to understand the data.

Running R commands in RStudio. The R programming code is written into a file using the editor in the top left pane. With the cursor on the line containing the code we click the `Run` *button to pass the code on to the R* `Console` *to have it run by the R Interpreter to produce the plot we see in the bottom right pane.*

{#fig:intror:rstudio_weather_scatterplot}

Your donation will support ongoing development and give you access to the ***PDF version of this book****. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

2.6 Packages and Libraries

REVIEW The power of the R ecosystem comes from the ability to extend the language by its nature as open source software. Anyone is able to contribute to the R ecosystem and by following stringent guidelines they can have their contributions included in the comprehensive R archive network, abbreviated as CRAN. Such contributions are collected by an author into what is called a *package*. Over the decades many researchers and developers have contributed very many *package*s to CRAN.

A *package* is how R collects together *command*s for a specific collection of tasks. A *command* is a verb in the computer language used to tell the computer to do something. There are over 17,000 packages available for R from almost as many different authors. Hence there are very many verbs available to build our sentences to command R appropriately. With so many packages there is bound to be a package or two covering essentially any kind of processing we could imagine though we will also find packages offering the same or similar commands (verbs) perhaps even with very different meanings.

Most chapters will list at the beginning of the chapter the R *package*s that are required for us to be able to replicate the examples presented in that chapter. Packages are installed from the Internet (from the securely managed CRAN package repository) into a local *library* on our own computer. A *library* is a folder on our computer's storage which contains sub-folders corresponding to each of the installed packages.

To install a package from the Internet we can use the command install.packages() and provide to it as an *argument* the name of the package to install. The package name is provided as a *string* of *characters* within quotes and supplied as the *pkgs= argument* to the command as in the following code. Here we choose to install a package called `dplyr` —a very useful package for data manipulations.

```
# Install a package from a CRAN repository.

install.packages(pkgs="dplyr")
```

Once a package is installed we can access the commands provided by that package by prefixing the command name with the package name as in ggplot2::qplot(). This is to say that qplot() is provided by the ggplot2 (Wickham et al. 2020) package.

References

Wickham, Hadley, Winston Chang, Lionel Henry, Thomas Lin Pedersen, Kohske Takahashi, Claus Wilke, Kara Woo, Hiroaki Yutani, and Dewey Dunnington. 2020. *Ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics*. https://CRAN.R-project.org/package=ggplot2.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

2.7 A Glimpse of the Dataset

REVIEW Another example of a useful command that we will find ourselves using often is the glimpse() command from the dplyr (Wickham et al. 2021) package. This command can be accessed in the R console as dplyr::glimpse() once the dplyr (Wickham et al. 2021) package has been installed. This particular command accepts an *argument x*= which names the dataset we wish to glimpse. In the following R example we use the weatherAUS dataset from the rattle (G. Williams 2020) package.

```
# Review the dataset.


dplyr::glimpse(x=rattle::weatherAUS)
```

```
## Rows: 176,747
## Columns: 24
## $ Date          <date> 2008-12-01, 2008-12-02, 2008-12-03, 2008-12-04, 2008-12…
## $ Location      <chr> "Albury", "Albury", "Albury", "Albury", "Albury", "Albur…
## $ MinTemp       <dbl> 13.4, 7.4, 12.9, 9.2, 17.5, 14.6, 14.3, 7.7, 9.7, 13.1, …
## $ MaxTemp       <dbl> 22.9, 25.1, 25.7, 28.0, 32.3, 29.7, 25.0, 26.7, 31.9, 30…
## $ Rainfall      <dbl> 0.6, 0.0, 0.0, 0.0, 1.0, 0.2, 0.0, 0.0, 0.0, 1.4, 0.0, 2…
## $ Evaporation   <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …
## $ Sunshine      <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …
## $ WindGustDir   <ord> W, WNW, WSW, NE, W, WNW, W, W, NNW, W, N, NNE, W, SW, NA…
## $ WindGustSpeed <dbl> 44, 44, 46, 24, 41, 56, 50, 35, 80, 28, 30, 31, 61, 44, …
## $ WindDir9am    <ord> W, NNW, W, SE, ENE, W, SW, SSE, SE, S, SSE, NE, NNW, W, …
## $ WindDir3pm    <ord> WNW, WSW, WSW, E, NW, W, W, W, NW, SSE, ESE, ENE, NNW, S…
## $ WindSpeed9am  <dbl> 20, 4, 19, 11, 7, 19, 20, 6, 7, 15, 17, 15, 28, 24, 4, N…
## $ WindSpeed3pm  <dbl> 24, 22, 26, 9, 20, 24, 24, 17, 28, 11, 6, 13, 28, 20, 30…
## $ Humidity9am   <int> 71, 44, 38, 45, 82, 55, 49, 48, 42, 58, 48, 89, 76, 65, …
## $ Humidity3pm   <int> 22, 25, 30, 16, 33, 23, 19, 19, 9, 27, 22, 91, 93, 43, 3…
## $ Pressure9am   <dbl> 1007.7, 1010.6, 1007.6, 1017.6, 1010.8, 1009.2, 1009.6, …
## $ Pressure3pm   <dbl> 1007.1, 1007.8, 1008.7, 1012.8, 1006.0, 1005.4, 1008.2, …
## $ Cloud9am      <int> 8, NA, NA, NA, 7, NA, 1, NA, NA, NA, NA, 8, 8, NA, NA, 0…
## $ Cloud3pm      <int> NA, NA, 2, NA, 8, NA, NA, NA, NA, NA, NA, 8, 8, 7, NA, N…
## $ Temp9am       <dbl> 16.9, 17.2, 21.0, 18.1, 17.8, 20.6, 18.1, 16.3, 18.3, 20…
## $ Temp3pm       <dbl> 21.8, 24.3, 23.2, 26.5, 29.7, 28.9, 24.6, 25.5, 30.2, 28…
## $ RainToday     <fct> No, No, No, No, No, No, No, No, No, Yes, No, Yes, Yes, Y…
## $ RISK_MM       <dbl> 0.0, 0.0, 0.0, 1.0, 0.2, 0.0, 0.0, 0.0, 1.4, 0.0, 2.2, 1…
## $ RainTomorrow  <fct> No, No, No, No, No, No, No, No, Yes, No, Yes, Yes, Yes, …
```

We can see here the command that was run and the output from running that command. As a convention used in this book the output from running R commands is prefixed with `##`. The `#` introduces a comment in an R script file and tells R to ignore everything that follows on that line. We use the `##' convention throughout the book to clearly identify output produced by R. When we run these commands ourselves in R this prefix is not displayed.

Long lines of output are also truncated for our presentation here. The `...` at the end of the lines and the `....` at the end of the output indicate that the output has been truncated for the sake of keeping our example output to a minimum.

This is a data frame, which is the basic data structure used to store a dataset within R, enhanced by the tidyverse to add functionality that improves our interactions with the data frame.

References

Wickham, Hadley, Romain François, Lionel Henry, and Kirill Müller. 2021. *Dplyr: A Grammar of Data Manipulation*. https://CRAN.R-project.org/package=dplyr.
Williams, Graham. 2020. *Rattle: Graphical User Interface for Data Science in r*. https://rattle.togaware.com/.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

2.8 Attaching a Package

REVIEW We can *attach* a package to our *current* R session from our local *library* for added convenience. This will make the command available during this specific R session without the requirement to specify the package name each time we use the command. Attaching a package tells the R software to look within that package (and then to look within any other attached packages) when it needs to find out what the command should do (the *definition* of the command).

We can see which packages are currently attached using stringi::search(). The order in which they are listed here corresponds to the order in which R searches for the definition of a command.

```
##  [1] ".GlobalEnv"        "package:stats"
##  [3] "package:graphics"  "package:grDevices"
##  [5] "package:utils"     "package:datasets"
##  [7] "package:methods"   "Autoloads"
##  [9] "package:base"
```

Notice that a collection of *packages* are installed by default among a couple of other special objects (`.GlobalEnv` and `Autoloads`).

A package is attached using the base::library() command. The base::library() command takes an argument to identify the *package*= we wish to *attach*.

```
# Load packages from the local library into the R session.


library(package=dplyr)
library(package=rattle)
```

Running these two commands will affect the search path by placing these packages early within the path.

```
##  [1] ".GlobalEnv"        "package:rattle"
##  [3] "package:dplyr"     "package:stats"
##  [5] "package:graphics"  "package:grDevices"
##  [7] "package:utils"     "package:datasets"
##  [9] "package:methods"   "Autoloads"
## [11] "package:base"
```

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

2.9 Simplifying Commands

REVIEW By attaching the dplyr (Wickham et al. 2021) package we can drop the package name prefix for any commands from the package. Similarly by attaching rattle (G. Williams 2020) we can drop the package name prefix from the name of the dataset. Our previous pillar::glimpse() command can be simplified.

```
# Review the dataset.

glimpse(x=weatherAUS)
```

```
## Rows: 176,747
## Columns: 24
## $ Date          <date> 2008-12-01, 2008-12-02, 2008-12-03, 2008-12-04, 2008-12…
## $ Location      <chr> "Albury", "Albury", "Albury", "Albury", "Albury", "Albur…
## $ MinTemp       <dbl> 13.4, 7.4, 12.9, 9.2, 17.5, 14.6, 14.3, 7.7, 9.7, 13.1, …
## $ MaxTemp       <dbl> 22.9, 25.1, 25.7, 28.0, 32.3, 29.7, 25.0, 26.7, 31.9, 30…
## $ Rainfall      <dbl> 0.6, 0.0, 0.0, 0.0, 1.0, 0.2, 0.0, 0.0, 0.0, 1.4, 0.0, 2…
## $ Evaporation   <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …
## $ Sunshine      <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …
## $ WindGustDir   <ord> W, WNW, WSW, NE, W, WNW, W, W, NNW, W, N, NNE, W, SW, NA…
## $ WindGustSpeed <dbl> 44, 44, 46, 24, 41, 56, 50, 35, 80, 28, 30, 31, 61, 44, …
## $ WindDir9am    <ord> W, NNW, W, SE, ENE, W, SW, SSE, SE, S, SSE, NE, NNW, W, …
## $ WindDir3pm    <ord> WNW, WSW, WSW, E, NW, W, W, W, NW, SSE, ESE, ENE, NNW, S…
## $ WindSpeed9am  <dbl> 20, 4, 19, 11, 7, 19, 20, 6, 7, 15, 17, 15, 28, 24, 4, N…
## $ WindSpeed3pm  <dbl> 24, 22, 26, 9, 20, 24, 24, 17, 28, 11, 6, 13, 28, 20, 30…
## $ Humidity9am   <int> 71, 44, 38, 45, 82, 55, 49, 48, 42, 58, 48, 89, 76, 65, …
## $ Humidity3pm   <int> 22, 25, 30, 16, 33, 23, 19, 19, 9, 27, 22, 91, 93, 43, 3…
## $ Pressure9am   <dbl> 1007.7, 1010.6, 1007.6, 1017.6, 1010.8, 1009.2, 1009.6, …
## $ Pressure3pm   <dbl> 1007.1, 1007.8, 1008.7, 1012.8, 1006.0, 1005.4, 1008.2, …
## $ Cloud9am      <int> 8, NA, NA, NA, 7, NA, 1, NA, NA, NA, NA, 8, 8, NA, NA, 0…
## $ Cloud3pm      <int> NA, NA, 2, NA, 8, NA, NA, NA, NA, NA, NA, 8, 8, 7, NA, N…
## $ Temp9am       <dbl> 16.9, 17.2, 21.0, 18.1, 17.8, 20.6, 18.1, 16.3, 18.3, 20…
## $ Temp3pm       <dbl> 21.8, 24.3, 23.2, 26.5, 29.7, 28.9, 24.6, 25.5, 30.2, 28…
## $ RainToday     <fct> No, No, No, No, No, No, No, No, No, Yes, No, Yes, Yes, Y…
## $ RISK_MM       <dbl> 0.0, 0.0, 0.0, 1.0, 0.2, 0.0, 0.0, 0.0, 1.4, 0.0, 2.2, 1…
## $ RainTomorrow  <fct> No, No, No, No, No, No, No, No, Yes, No, Yes, Yes, Yes, …
```

We can actually simplify this a little more. Often for a command we don't have to explicitly name all of the arguments. In the following example we drop the *package*= and the *x*= arguments as the commands themselves know what to expect implicitly.

```r
# Load packages from the local library into the R session.

library(dplyr)
library(rattle)

# Review the dataset.

glimpse(weatherAUS)
```

```
## Rows: 176,747
## Columns: 24
## $ Date          <date> 2008-12-01, 2008-12-02, 2008-12-03, 2008-12-04, 2008-12…
## $ Location      <chr> "Albury", "Albury", "Albury", "Albury", "Albury", "Albur…
## $ MinTemp       <dbl> 13.4, 7.4, 12.9, 9.2, 17.5, 14.6, 14.3, 7.7, 9.7, 13.1, …
## $ MaxTemp       <dbl> 22.9, 25.1, 25.7, 28.0, 32.3, 29.7, 25.0, 26.7, 31.9, 30…
## $ Rainfall      <dbl> 0.6, 0.0, 0.0, 0.0, 1.0, 0.2, 0.0, 0.0, 0.0, 1.4, 0.0, 2…
## $ Evaporation   <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …
## $ Sunshine      <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …
## $ WindGustDir   <ord> W, WNW, WSW, NE, W, WNW, W, W, NNW, W, N, NNE, W, SW, NA…
## $ WindGustSpeed <dbl> 44, 44, 46, 24, 41, 56, 50, 35, 80, 28, 30, 31, 61, 44, …
## $ WindDir9am    <ord> W, NNW, W, SE, ENE, W, SW, SSE, SE, S, SSE, NE, NNW, W, …
## $ WindDir3pm    <ord> WNW, WSW, WSW, E, NW, W, W, W, NW, SSE, ESE, ENE, NNW, S…
## $ WindSpeed9am  <dbl> 20, 4, 19, 11, 7, 19, 20, 6, 7, 15, 17, 15, 28, 24, 4, N…
## $ WindSpeed3pm  <dbl> 24, 22, 26, 9, 20, 24, 24, 17, 28, 11, 6, 13, 28, 20, 30…
## $ Humidity9am   <int> 71, 44, 38, 45, 82, 55, 49, 48, 42, 58, 48, 89, 76, 65, …
## $ Humidity3pm   <int> 22, 25, 30, 16, 33, 23, 19, 19, 9, 27, 22, 91, 93, 43, 3…
## $ Pressure9am   <dbl> 1007.7, 1010.6, 1007.6, 1017.6, 1010.8, 1009.2, 1009.6, …
## $ Pressure3pm   <dbl> 1007.1, 1007.8, 1008.7, 1012.8, 1006.0, 1005.4, 1008.2, …
## $ Cloud9am      <int> 8, NA, NA, NA, 7, NA, 1, NA, NA, NA, NA, 8, 8, NA, NA, 0…
## $ Cloud3pm      <int> NA, NA, 2, NA, 8, NA, NA, NA, NA, NA, NA, 8, 8, 7, NA, N…
## $ Temp9am       <dbl> 16.9, 17.2, 21.0, 18.1, 17.8, 20.6, 18.1, 16.3, 18.3, 20…
## $ Temp3pm       <dbl> 21.8, 24.3, 23.2, 26.5, 29.7, 28.9, 24.6, 25.5, 30.2, 28…
## $ RainToday     <fct> No, No, No, No, No, No, No, No, No, Yes, No, Yes, Yes, Y…
## $ RISK_MM       <dbl> 0.0, 0.0, 0.0, 1.0, 0.2, 0.0, 0.0, 0.0, 1.4, 0.0, 2.2, 1…
## $ RainTomorrow  <fct> No, No, No, No, No, No, No, No, Yes, No, Yes, Yes, Yes, …
```

% “start up” is a verb, “start-up” is an adjective and “startup” is a noun.

A number of packages are automatically attached when R starts. The first stringi::search() command above returned a vector of packages and since we had yet to attach any further packages those listed are the ones automatically attached. One of those is the base (R Core Team 2021) package which provides the base::library() command.

References

R Core Team. 2021. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Wickham, Hadley, Romain François, Lionel Henry, and Kirill Müller. 2021. *Dplyr: A Grammar of Data Manipulation*. https://CRAN.R-project.org/package=dplyr.

Williams, Graham. 2020. *Rattle: Graphical User Interface for Data Science in r*. https://rattle.togaware.com/.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*

Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

2.10 Working with the Library

REVIEW To summarise then, when we interact with R we can usually drop the package prefix for those commands that can be found in one (or more) of the attached packages. Throughout the running text in this book we retain the package prefix to clarify where each command comes from. However, within the code we tend to drop the package name prefix.

A prefix can still be useful in larger programs to ensure we are using the correct command and to communicate to the human reader where the command comes from. Some packages do implement commands with the same names as commands defined differently and found in other packages and the prefix notation is then useful to specify which command we are referring to.

Each of the following chapters will begin with a list of *packages* to *attach* from the *library* into the R session. Below is an example of attaching five common packages for our work. Attaching the listed packages will allow the examples presented within the chapter to be replicated. In the code below take note of the use of the hash (`#`) to introduce a comment which is ignored by R—R will not attempt to understand the comments as commands. Comments are there to assist the human reader in understanding our programs.

```
# Load packages required for this script.

library(rattle)   # The weatherAUS dataset and normVarNames().
library(readr)    # Efficient reading of CSV data.
library(dplyr)    # Data wrangling and glimpse().
library(ggplot2)  # Visualise data.
library(magrittr) # Pipes %>%, %<>%, %T>%, %$%.
```

In starting up an R session (for example, by starting up RStudio) we can enter the above library() commands into an R script file created using the `New RScript File` menu option in RStudio and then ask RStudio to `Run` the commands. RStudio will send each command to the R `Console` which sends the command on to the R software. It is the R software that then runs the commands. If R responds that the package is not available then the package will need to be installed, which we can do from RStudio's `Tools` menu or by directly using utils::install.packages() as we saw above.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

2.11 Getting Help

REVIEW A key skill of any programmer, including those programming over data, is the ability to identify how to access the full power of our tools. The breadth and depth of the capabilities of R means that there is much to learn around both the basics of R programming and the multitude of packages that support the data scientist.

R provides extensive in-built documentation with many introductory manuals and resources accessible through the RStudio `Help` tab. These are a good adjunct to our very brief introduction here to R. Further we can use RStudio's search facility for documentation on any action and we will find manual pages that provide an understanding of the purpose, arguments, and return value of *function*s, *command*s, and *operator*s. We can also ask for help using the utils::? operator as in:

```
# Ask for documentation on using the library command.

? library
```

The package which provides this operator is another package that is attached by default into R. Thus we can drop its prefix as we did here when we run the command in the R Console.

RStudio provides a search box which can be used to find specific topics within the vast collection of R documentation. Now would be a good time to check the documentation for the base::library() command.

Beginning with this chapter we will also list at the conclusion of each chapter the functions, commands, and operators introduced within the chapter together with a brief synopsis of the purpose. The following section reviews all of the functions introduced so far.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open*

source software since 1984.

Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

3 R Constructs

20210103 Here we introduce the language constructs of R including the powerful concept of pipes as a foundation for building sophisticated data processing pipelines in R.

To illustrate the R constructs we will take a copy of the rattle::weatherAUS dataset (daily weather observations from Australia) and save it as a variable named `ds`. The column names in the dataset are also normalised as we would normally do as data scientists. The operations here will become familiar as we progress through the book. To aid our understanding for now though, we can ready the code as:

Obtain the weatherAUS dataset from the rattle package; clean up the column names using the clean_names function from the janitor package, choosing to align numerals to the right; save the resulting dataset into an R variable named ds.

```
rattle::weatherAUS %>%
  janitor::clean_names(numerals="right") ->
ds
```

For reference the resulting column names (variables) are:

```
names(ds)
```

```
##  [1] "date"            "location"        "min_temp"        "max_temp"
##  [5] "rainfall"        "evaporation"     "sunshine"        "wind_gust_dir"
##  [9] "wind_gust_speed" "wind_dir_9am"    "wind_dir_3pm"    "wind_speed_9am"
## [13] "wind_speed_3pm"  "humidity_9am"    "humidity_3pm"    "pressure_9am"
## [17] "pressure_3pm"    "cloud_9am"       "cloud_3pm"       "temp_9am"
## [21] "temp_3pm"        "rain_today"      "risk_mm"         "rain_tomorrow"
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

3.1 A Data Frame as a Dataset

20210103 A data frame is essentially a rectangular table (or matrix) of data consisting of rows (*observations*) and columns (*variables*). We can base::print.data.frame() to view a table, here choosing the first 10 observations of the first 6 variables of the `ds` dataset.

```
# Display the table structure of the ingested dataset.

ds[1:10,1:6] %>% print.data.frame()
```

```
##          date location min_temp max_temp rainfall evaporation
## 1  2008-12-01   Albury     13.4     22.9      0.6          NA
## 2  2008-12-02   Albury      7.4     25.1      0.0          NA
## 3  2008-12-03   Albury     12.9     25.7      0.0          NA
## 4  2008-12-04   Albury      9.2     28.0      0.0          NA
## 5  2008-12-05   Albury     17.5     32.3      1.0          NA
## 6  2008-12-06   Albury     14.6     29.7      0.2          NA
## 7  2008-12-07   Albury     14.3     25.0      0.0          NA
## 8  2008-12-08   Albury      7.7     26.7      0.0          NA
## 9  2008-12-09   Albury      9.7     31.9      0.0          NA
## 10 2008-12-10   Albury     13.1     30.1      1.4          NA
```

Alternatively we might sample 10 random observations (dplyr::sample_n()) of 5 random variables (dplyr::select()):

```
# Display a random selection of observations and variables.

ds %>%
  sample_n(10) %>%
  select(sample(1:ncol(ds), 5)) %>%
  print.data.frame()
```

```
##    wind_gust_speed max_temp rainfall wind_speed_3pm rain_today
## 1               30     25.1      0.2             11         No
## 2               72     30.7      1.0             33         No
## 3               56     14.9      1.6             20        Yes
## 4               33     28.8      0.0             20         No
## 5               37     31.3      0.0             20         No
## 6               35     35.7      0.0             15         No
## 7               24     15.5      0.0             15         No
## 8               22     22.5      0.0              6         No
## 9               31     20.0      0.4              4         No
## 10              35     21.9      0.0             17         No
```

This tabular form (i.e., it has rows and columns) is common for data science and we refer to it as our dataset.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

3.2 Assignment

20210103 The assignment operator in R base::<- is used to save a result into the computer's memory and give it a name that we can refer to later. We can use this operator to save the result of the sequence of operations from a pipeline (see Section 3.7).

```
# Select variables/observations and save the result.

rainy_days <-
  ds %>%
  select(min_temp, max_temp, rainfall, sunshine) %>%
  filter(rainfall >= 1)
```

An alternative that makes sense within the pipe paradigm that we use is to use the forward assignment operator base::-> to save the resulting data into a variable in a logically forward flowing way.

```
# Demonstrate use of the forward assignment operator.

ds %>%
  select(min_temp, max_temp, rainfall, sunshine) %>%
  filter(rainfall >= 1) ->
  rainy_days
```

Using the forward assignment operator tends to hide the final variable into which the assignment is made. This important side effect of the series of commands, the assignment to `rainy_days`, could easily be missed unlike our use of the backward assignment earlier.

When using the forward assignment be sure to un-indent the final variable name so that the assignment is clearer and the un-indented variable name serves as the final *bracket* for the command pipeline.

```
# Best to unindent the final assigned variable for clarity.

ds %>%
  select(min_temp, max_temp, rainfall, sunshine) %>%
  filter(rainfall >= 1) ->
rainy_days
```

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

3.3 Case Statement

20200421 Many languages support a case or switch statement. R did not originally have a case statement but it was introduced through tidyr as dplyr::case_when():

```
library(dplyr)

with(ds %>% sample_n(100),
     case_when(max_temp > 35 ~ "hot",
               max_temp > 25 ~ "warm",
               max_temp > 18 ~ "mild",
               max_temp > 12 ~ "cool",
               max_temp >  0 ~ "cold",
               TRUE          ~ "freezing"))
```

```
##   [1] "mild"     "mild"     "hot"      "warm"     "mild"     "mild"
##   [7] "cool"     "warm"     "mild"     "cool"     "mild"     "cool"
##  [13] "warm"     "cool"     "warm"     "warm"     "mild"     "warm"
##  [19] "warm"     "mild"     "cool"     "warm"     "mild"     "warm"
##  [25] "cool"     "warm"     "warm"     "mild"     "cool"     "cold"
##  [31] "cool"     "mild"     "warm"     "warm"     "cool"     "warm"
##  [37] "warm"     "cool"     "mild"     "mild"     "mild"     "mild"
##  [43] "mild"     "warm"     "mild"     "cool"     "warm"     "cool"
##  [49] "mild"     "warm"     "warm"     "mild"     "warm"     "warm"
##  [55] "cool"     "mild"     "warm"     "warm"     "mild"     "cool"
##  [61] "cool"     "mild"     "warm"     "warm"     "mild"     "mild"
##  [67] "mild"     "cool"     "warm"     "warm"     "warm"     "cool"
##  [73] "warm"     "warm"     "cool"     "mild"     "mild"     "mild"
##  [79] "warm"     "warm"     "warm"     "mild"     "warm"     "mild"
##  [85] "warm"     "mild"     "warm"     "cool"     "cool"     "cool"
##  [91] "warm"     "warm"     "warm"     "cool"     "mild"     "cool"
##  [97] "freezing" "warm"     "cool"     "warm"
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

3.4 Commands

20200501 A *command* is a function (Section 3.5) for which we are generally more interested in the actions or side-effects that are performed by the command rather than the value returned by the command. For example, the base::library() command is run to attach a package from the library (the action) and consequently modifies the search path that R uses to find commands (the side-effect).

```
# Load package from the local library into the R session.

library(rattle)
```

A *command* is also a function and does in fact return a value. In the above example we do not see any value printed. Functions in R can be implemented to return values invisibly. This is the case for base::library(). We can ask R to print the value when the returned result is invisible using the function base::print().

```
# Demonstrate that library() returns a value invisibly.

print(library(rattle))
```

```
##  [1] "readr"     "rattle"    "bitops"    "tibble"    "tidyr"     "stringr"
##  [7] "ggplot2"   "dplyr"     "magrittr"  "stats"     "graphics"  "grDevices"
## [13] "utils"     "datasets"  "methods"   "base"
```

We see that the value returned by base::library() is a vector of character strings. Each character string names a package that has been attached during this session of R. Notice that the vector begins with item `[1]` and then item `[5]` continues on the second line, and so on. We can save the resulting vector into a variable (Section 3.18) and then index the variable to identify packages at specific locations within the vector.

```r
# Load a package and save the value returned.

l <- library(rattle)

# Review one of the returned values.

l[7]
```

```
## [1] "ggplot2"
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

3.5 Functions

20210103 When using R we write *command*s that instruct the R software to perform particular actions. Formally commands are actually *function*s which mathematically operate on a set of inputs and produces an output. R provides a wealth of functions and these form the verbs of the language we use to construct sentences. The verbs are the doing words! Although technically everything is a function in R we can generally think of each verb as either a function, a command (Section 3.4) or an operator (Section 3.6).

All functions in R take arguments and return values. When we invoke a function (as distinct from what we will identify as a command) we are interested in the value that it returns. Below is a simple example base::sum()ing two numbers.

```
# Add two numbers.

sum(1, 2)
```

```
## [1] 3
```

Here the function has returned a single value, though it is technically returned as a vector of length 1 containing just the item `3`. The result can be stored into a variable as discussed in Section 3.18.

We can define out own functions:

```
mysum <- function(x) x + 1
```

A shorthand for function() introduced in 2021 is:

```
mysum <- \(x) x + 1
```

This syntax is most useful for anonymous functions inline.

```
sapply(1:10, \(x) x + 1)
```

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

3.6 Operators

20210103 The third type of function is the *operator*. We use *operator*s in the flow of an expression. Technically they are called *infix* operators.

In the following example we use the base::+ infix operator.

```
# Add two numbers using an infix operator.

1 + 2
```

```
## [1] 3
```

Internally R actually converts the operator into a functional prefix form to be run in the same way as any other function.

```
# Add two numbers using the equivalent functional form.

`+`(1, 2)
```

```
## [1] 3
```

Note that R also supports using quotes and double quotes instead of the backtick we used in the code above. The backtick is preferred as per page 142 of Wickham (2014).

The key thing to remember is that all commands and operators are functions and we instruct R to take action by running functions.

References

Wickham, Hadley. 2014. *Advanced R*. Chapman & Hall/CRC the r Series. Chapman & Hall.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

3.7 Pipe Operator

20210103 A function (Section 3.5) performs an action on input data and returns the results of those actions as the output from the function. They are the verbs of the language—the action words of our sentences.

As we learn new functions we will construct longer sentences that string together a sequence of verbs to deliver the outcomes. This is a powerful programming concept, combining dedicated, well designed and implemented functions, each focused on achieving a specific outcome.

To combine single focus functions into more complex operations we use the powerful concept of *pipe*s. Pipes will be familiar to command line users of Unix and Linux. The idea is to pass the output of one function on as the input to another function. Each function does one task very well, very accurately, and very simply from a user's point of view. We can pipe together many such specialist functions to deliver very complex and quite sophisticated data transformations in an easily accessible manner.

Pipes were introduced in R through the magrittr (Bache and Wickham 2020) package (`%>%`) and became part of base R in 2021 with version 4.1 (`|>`).

To illustrate the concept of pipes recall the contents of the dataset (rattle::weatherAUS):

```
# Review the dataset of weather observations.

ds
```

```
## # A tibble: 176,747 x 24
##    date       location min_temp max_temp rainfall evaporation sunshine
##    <date>     <chr>       <dbl>    <dbl>    <dbl>       <dbl>    <dbl>
##  1 2008-12-01 Albury       13.4     22.9      0.6          NA       NA
##  2 2008-12-02 Albury        7.4     25.1      0            NA       NA
##  3 2008-12-03 Albury       12.9     25.7      0            NA       NA
##  4 2008-12-04 Albury        9.2     28        0            NA       NA
##  5 2008-12-05 Albury       17.5     32.3      1            NA       NA
##  6 2008-12-06 Albury       14.6     29.7      0.2          NA       NA
##  7 2008-12-07 Albury       14.3     25        0            NA       NA
##  8 2008-12-08 Albury        7.7     26.7      0            NA       NA
##  9 2008-12-09 Albury        9.7     31.9      0            NA       NA
## 10 2008-12-10 Albury       13.1     30.1      1.4          NA       NA
## # … with 176,737 more rows, and 17 more variables: wind_gust_dir <ord>,
## #   wind_gust_speed <dbl>, wind_dir_9am <ord>, wind_dir_3pm <ord>,
## #   wind_speed_9am <dbl>, wind_speed_3pm <dbl>, humidity_9am <int>,
## #   humidity_3pm <int>, pressure_9am <dbl>, pressure_3pm <dbl>,
## #   cloud_9am <int>, cloud_3pm <int>, temp_9am <dbl>, temp_3pm <dbl>,
## #   rain_today <fct>, risk_mm <dbl>, rain_tomorrow <fct>
```

We might be interested in the distribution of specific numeric variables. For that we will dplyr::select() a few numeric variables using a *pipe*.

```
# Select variables from the dataset.

ds %>%
  select(min_temp, max_temp, rainfall, sunshine)
```

```
## # A tibble: 176,747 x 4
##    min_temp max_temp rainfall sunshine
##       <dbl>    <dbl>    <dbl>    <dbl>
##  1     13.4     22.9      0.6       NA
##  2      7.4     25.1      0         NA
##  3     12.9     25.7      0         NA
##  4      9.2     28        0         NA
##  5     17.5     32.3      1         NA
##  6     14.6     29.7      0.2       NA
##  7     14.3     25        0         NA
##  8      7.7     26.7      0         NA
##  9      9.7     31.9      0         NA
## 10     13.1     30.1      1.4       NA
## # … with 176,737 more rows
```

Typing `ds` by itself lists the whole dataset. Piping the whole dataset to dplyr::select() using the *pipe* `%>%` selects the named variables. The end result returned as the output of the pipeline is a subset of the original dataset containing just the named columns.

References

Bache, Stefan Milton, and Hadley Wickham. 2020. *Magrittr: A Forward-Pipe Operator for r*. https://CRAN.R-project.org/package=magrittr.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

3.8 Pipeline

20210103 Suppose we want to produce a base::summary() of a selection of numeric variables. We can *pipe* the output of dplyr::select() into base::summary().

```
# Select variables from the dataset and summarise the result.

ds %>%
  select(min_temp, max_temp, rainfall, sunshine) %>%
  summary()
```

```
##     min_temp        max_temp        rainfall          sunshine
##  Min.   :-8.70   Min.   :-4.10   Min.   :  0.000   Min.   : 0.00
##  1st Qu.: 7.50   1st Qu.:18.10   1st Qu.:  0.000   1st Qu.: 4.90
##  Median :12.00   Median :22.80   Median :  0.000   Median : 8.50
##  Mean   :12.15   Mean   :23.36   Mean   :  2.241   Mean   : 7.66
##  3rd Qu.:16.90   3rd Qu.:28.40   3rd Qu.:  0.600   3rd Qu.:10.60
##  Max.   :33.90   Max.   :48.90   Max.   :474.000   Max.   :14.50
##  NA's   :2349    NA's   :2105    NA's   :4318      NA's   :93859
```

Perhaps we would like to review only those observations where there is more than a little rain on the day of the observation. To do so we stats::filter() the observations.

```
# Select specific variables and observations from the dataset.

ds %>%
  select(min_temp, max_temp, rainfall, sunshine) %>%
  filter(rainfall >= 1)
```

```
## # A tibble: 39,122 x 4
##    min_temp max_temp rainfall sunshine
##       <dbl>    <dbl>    <dbl>    <dbl>
##  1     17.5     32.3      1         NA
##  2     13.1     30.1      1.4       NA
##  3     15.9     21.7      2.2       NA
##  4     15.9     18.6     15.6       NA
##  5     12.6     21        3.6       NA
##  6     13.5     22.9     16.8       NA
##  7     11.2     22.5     10.6       NA
##  8     12.5     24.2      1.2       NA
##  9     18.8     35.2      6.4       NA
## 10     14.6     29        3         NA
## # … with 39,112 more rows
```

This sequence of functions operating on the original rattle::weatherAUS dataset returns a subset of that dataset where all observations have at least 1mm of rain.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

3.9 Pipeline Construction

20210103 Continuing with our pipeline example, we might want a base::summary() of the dataset.

```
# Summarise subset of variables for observations with rainfall.

ds %>%
  select(min_temp, max_temp, rainfall, sunshine) %>%
  filter(rainfall >= 1) %>%
  summary()
```

```
##     min_temp        max_temp        rainfall          sunshine
##  Min.   :-8.50   Min.   :-4.10   Min.   :  1.000   Min.   : 0.000
##  1st Qu.: 8.40   1st Qu.:15.50   1st Qu.:  2.200   1st Qu.: 2.400
##  Median :12.20   Median :19.30   Median :  4.600   Median : 5.500
##  Mean   :12.72   Mean   :20.19   Mean   :  9.681   Mean   : 5.379
##  3rd Qu.:17.20   3rd Qu.:24.40   3rd Qu.: 10.800   3rd Qu.: 8.100
##  Max.   :28.90   Max.   :46.30   Max.   :474.000   Max.   :14.200
##  NA's   :127     NA's   :176                       NA's   :20056
```

It could be useful to contrast this with a base::summary() of those observations where there was little or no rain.

```
# Summarise observations with little or no rainfall.

ds %>%
  select(min_temp, max_temp, rainfall, sunshine) %>%
  filter(rainfall < 1) %>%
  summary()
```

```
##     min_temp        max_temp       rainfall          sunshine
##  Min.   :-8.70   Min.   :-2.1   Min.   :0.00000   Min.   : 0.00
##  1st Qu.: 7.20   1st Qu.:19.1   1st Qu.:0.00000   1st Qu.: 6.20
##  Median :11.90   Median :23.8   Median :0.00000   Median : 9.30
##  Mean   :11.97   Mean   :24.3   Mean   :0.05825   Mean   : 8.37
##  3rd Qu.:16.70   3rd Qu.:29.3   3rd Qu.:0.00000   3rd Qu.:11.00
##  Max.   :33.90   Max.   :48.9   Max.   :0.90000   Max.   :14.50
##  NA's   :569     NA's   :612                      NA's   :70700
```

Any number of functions can be included in a *pipeline* to achieve the results we desire. In the following chapters we will see many examples and some will string together ten or more functions. Each step along the way is of itself generally easily understandable. The power is in what we can achieve by stringing together many simple steps to produce something more complex.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

3.10 Pipeline Identity Operator

REVIEW A handy trick when building a pipeline is to use what is effectively the identity operator. An identity operator simply takes the data being communicated through the pipeline, and without changing it passes it on to the next operator in the pipeline. An effective identity operator is constructed with the syntax `{.}`. In R terms this is a compound statement containing just the period whereby the period represents the data. Effectively this is an operator that passes the data through without processing it—an identity operator.

Why is this useful? Whilst we are building our pipeline, one line at a time, we will be wanting to put a pipe at the end of each line, but of cause we can not do so if there is no following operator. Also, whilst debugging a pipeline, we may want to execute only a part of it, and so the identity operator is handy there too.

As a typical scenario we might be in the process of building a pipeline as here and find that including the `%>%` at the end of the line of the dplyr::select() operation:

```
ds %>%
  select(rainfall, min_temp, max_temp, sunshine) %>%
  {.}
```

```
## # A tibble: 176,747 x 4
##    rainfall min_temp max_temp sunshine
##       <dbl>    <dbl>    <dbl>    <dbl>
##  1      0.6     13.4     22.9       NA
##  2      0        7.4     25.1       NA
##  3      0       12.9     25.7       NA
##  4      0        9.2     28         NA
##  5      1       17.5     32.3       NA
##  6      0.2     14.6     29.7       NA
##  7      0       14.3     25         NA
##  8      0        7.7     26.7       NA
##  9      0        9.7     31.9       NA
## 10      1.4     13.1     30.1       NA
## # … with 176,737 more rows
```

We then add the next operation into the pipeline without having to modify any of the code already present:

```
ds %>%
  select(rainfall, min_temp, max_temp, sunshine) %>%
  summary() %>%
  {.}
```

```
##     rainfall          min_temp        max_temp        sunshine
##  Min.   :  0.000   Min.   :-8.70   Min.   :-4.10   Min.   : 0.00
##  1st Qu.:  0.000   1st Qu.: 7.50   1st Qu.:18.10   1st Qu.: 4.90
##  Median :  0.000   Median :12.00   Median :22.80   Median : 8.50
##  Mean   :  2.241   Mean   :12.15   Mean   :23.36   Mean   : 7.66
##  3rd Qu.:  0.600   3rd Qu.:16.90   3rd Qu.:28.40   3rd Qu.:10.60
##  Max.   :474.000   Max.   :33.90   Max.   :48.90   Max.   :14.50
##  NA's   :4318      NA's   :2349    NA's   :2105    NA's   :93859
```

And so on. Whilst it appears quite a minor convenience, over time as we build more pipelines, this becomes quite a handy trick.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

3.11 Pipeline Syntactic Sugar

20210103 For the technically minded we note that what is actually happening here is that the syntax (i.e., how we write the sentences) is changed by R from the traditional functional expression, increasing the ease with which we can read the code. This is important as we keep in mind that we write our code for others (and ourselves later on) to read.

The pipeline below combines a series of commands to operate on the dataset.

```
# Summarise observations with little or no rainfall.

ds %>%
  select(min_temp, max_temp, rainfall, sunshine) %>%
  filter(rainfall < 1) %>%
  summary()
```

```
##     min_temp        max_temp       rainfall          sunshine
##  Min.   :-8.70   Min.   :-2.1   Min.   :0.00000   Min.   : 0.00
##  1st Qu.: 7.20   1st Qu.:19.1   1st Qu.:0.00000   1st Qu.: 6.20
##  Median :11.90   Median :23.8   Median :0.00000   Median : 9.30
##  Mean   :11.97   Mean   :24.3   Mean   :0.05825   Mean   : 8.37
##  3rd Qu.:16.70   3rd Qu.:29.3   3rd Qu.:0.00000   3rd Qu.:11.00
##  Max.   :33.90   Max.   :48.9   Max.   :0.90000   Max.   :14.50
##  NA's   :569     NA's   :612                      NA's   :70700
```

Contrast this with how it is mapped by R into the functional construct below, which is how we might have traditionally written it. For many of us it will take quite a bit of effort to parse this traditional functional form of the expression, and so to understand what it is doing. The pipeline alternative above provides a clearer narrative.

```r
# Functional form equivalent to the pipeline above.

summary(filter(select(ds,
                      min_temp, max_temp, rainfall, sunshine),
               rainfall < 1))
```

```
##     min_temp        max_temp       rainfall          sunshine
##  Min.   :-8.70   Min.   :-2.1   Min.   :0.00000   Min.   : 0.00
##  1st Qu.: 7.20   1st Qu.:19.1   1st Qu.:0.00000   1st Qu.: 6.20
##  Median :11.90   Median :23.8   Median :0.00000   Median : 9.30
##  Mean   :11.97   Mean   :24.3   Mean   :0.05825   Mean   : 8.37
##  3rd Qu.:16.70   3rd Qu.:29.3   3rd Qu.:0.00000   3rd Qu.:11.00
##  Max.   :33.90   Max.   :48.9   Max.   :0.90000   Max.   :14.50
##  NA's   :569     NA's   :612                      NA's   :70700
```

Anything that improves the readability of our code is useful. Computers are quite capable of doing the hard work of transforming a simpler sentence into this much more complex looking sentence for its own purposes. For our purposes, let's keep it simple for others to follow.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

3.12 Pipes: Assignment Pipe

20210103 The are several variations of the pipe operator available. A particularly handy operator implemented by magrittr (Bache and Wickham 2020) is the assignment pipe `%<>%`. This operator should be the left most pipe of any sequence of pipes. In addition to piping the dataset named on the left onto the function on the right the result coming out at the end of the right-hand pipeline is piped back to the original dataset overwriting its original contents in memory with the results from the pipeline.

A simple example is to replace a dataset with the same dataset after removing some observations (rows) and variables (columns). In the example below we stats::filter() and dplyr::select() the dataset to reduce it to just those observations and variables of interest. The result is piped backwards to the original dataset and thus overwrites the original data (which may or may not be a good thing). We do this on a temporary copy of the dataset and use the base::dim() function to report on the dimensions (rows and columns) of the resulting datasets.

```
ods <- ds
```

```
# Report on the dimensions of the dataset.

dim(ds)
```

```
## [1] 176747     24
```

```
# Demonstrate an assignment pipeline.

ds %<>%
  filter(rainfall==0) %>%
  select(min_temp, max_temp, sunshine)

# Confirm that the dataset has changed.

dim(ds)
```

```
## [1] 112447      3
```

Once again this is so-called *syntactic sugar*. The core assignment command is effectively translated by the computer into the following code (noting we did a little more that just the assignment in the above cod block).

```
# Functional form equivalent to the pipeline above.

ds <- select(filter(ds, rainfall==0), min_temp, max_temp, sunshine)
```

```
ds <- ods
```

References

Bache, Stefan Milton, and Hadley Wickham. 2020. *Magrittr: A Forward-Pipe Operator for r*. https://CRAN.R-project.org/package=magrittr.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

3.13 Pipes: Exposition Pipe

20210103 The exposition pipe `%$%`. is useful for commands that do not take a dataset as their argument but instead operates on vectors (such as columns from the dataset). An exposition pipe evaluates the following function within the context of the dataset passed to it, so that the variables of the dataset become available without the need to quote them. In our example we determine the correntlation between two columns, using stats::cor():

```
ds %>%
  filter(rainfall==0) %>%
  na.omit() %$%
  cor(min_temp, max_temp)
```

```
## [1] 0.7525428
```

Without an exposition pipe we might otherwise use base::with():

```
ds %>%
  filter(rainfall==0) %>%
  na.omit() %>%
  with(cor(min_temp, max_temp))
```

```
## [1] 0.7525428
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

3.14 Pipes: Tee Pipe

20210103 Another useful operation is the tee-pipe `%T>%` which causes the command that follows to be run as a side-pipe whilst piping the same data into that command and also into the then following command and the rest of the pipeline. The output from the first command is ignored, except for its side-effect, which might be to base::print() the intermediate results as below or to store the intermediate results before further processing, as in Section @ref(sec:pipe_tee_save).

A common use case is to whilst continuing on to assign the dataset itself to a variable. We will often see the following example.

```
# Demonstrate usage of a tee-pipe.

no_rain <-
  ds %>%
  filter(rainfall==0) %T>%
  print()
```

```
## # A tibble: 112,447 x 24
##    date       location min_temp max_temp rainfall evaporation sunshine
##    <date>     <chr>       <dbl>    <dbl>    <dbl>       <dbl>    <dbl>
##  1 2008-12-02 Albury        7.4     25.1        0          NA       NA
##  2 2008-12-03 Albury       12.9     25.7        0          NA       NA
##  3 2008-12-04 Albury        9.2     28          0          NA       NA
##  4 2008-12-07 Albury       14.3     25          0          NA       NA
##  5 2008-12-08 Albury        7.7     26.7        0          NA       NA
##  6 2008-12-09 Albury        9.7     31.9        0          NA       NA
##  7 2008-12-11 Albury       13.4     30.4        0          NA       NA
##  8 2008-12-15 Albury        8.4     24.6        0          NA       NA
##  9 2008-12-17 Albury       14.1     20.9        0          NA       NA
## 10 2008-12-20 Albury        9.8     25.6        0          NA       NA
## # … with 112,437 more rows, and 17 more variables: wind_gust_dir <ord>,
## #   wind_gust_speed <dbl>, wind_dir_9am <ord>, wind_dir_3pm <ord>,
## #   wind_speed_9am <dbl>, wind_speed_3pm <dbl>, humidity_9am <int>,
## #   humidity_3pm <int>, pressure_9am <dbl>, pressure_3pm <dbl>,
## #   cloud_9am <int>, cloud_3pm <int>, temp_9am <dbl>, temp_3pm <dbl>,
## #   rain_today <fct>, risk_mm <dbl>, rain_tomorrow <fct>
```

The tee-pipe processes the transformed dataset in two ways—once with base::print(), then continuing on with dplyr::select() and base::summary(). The tee-pipe splits the flow in two directions. The second flow continues the sequence of the pipeline.

```
ds %>%
  select(rainfall, min_temp, max_temp, sunshine) %>%
  filter(rainfall==0) %T>%
  print()  %>%
  select(min_temp, max_temp, sunshine) %>%
  summary()
```

```
## # A tibble: 112,447 x 4
##    rainfall min_temp max_temp sunshine
##       <dbl>    <dbl>    <dbl>    <dbl>
##  1        0      7.4     25.1       NA
##  2        0     12.9     25.7       NA
##  3        0      9.2     28         NA
##  4        0     14.3     25         NA
##  5        0      7.7     26.7       NA
##  6        0      9.7     31.9       NA
##  7        0     13.4     30.4       NA
##  8        0      8.4     24.6       NA
##  9        0     14.1     20.9       NA
## 10        0      9.8     25.6       NA
## # … with 112,437 more rows
```

```
##     min_temp        max_temp        sunshine
##  Min.   :-8.70   Min.   :-2.10   Min.   : 0.00
##  1st Qu.: 7.40   1st Qu.:19.80   1st Qu.: 6.90
##  Median :12.10   Median :24.60   Median : 9.70
##  Mean   :12.15   Mean   :24.97   Mean   : 8.72
##  3rd Qu.:16.90   3rd Qu.:30.00   3rd Qu.:11.10
##  Max.   :33.90   Max.   :48.90   Max.   :14.50
##  NA's   :518     NA's   :537     NA's   :59508
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

3.15 Pipes: Tee Pipe Intermediate Results

20210103 A use of the tee-pipe that can come in handy is to base::assign() intermediate results into a variable without breaking the main flow of the pipeline. The variable `tmp` in the following tee-pipe is assigned to within the base::globalenv() so that we can access it from anywhere in the R session.

```
ds %>%
  select(rainfall, min_temp, max_temp, sunshine) %>%
  filter(rainfall==0) %T>%
  {assign("tmp", ., globalenv())}  %>%
  select(min_temp, max_temp, sunshine) %>%
  summary()
```

```
##     min_temp        max_temp        sunshine
##  Min.   :-8.70   Min.   :-2.10   Min.   : 0.00
##  1st Qu.: 7.40   1st Qu.:19.80   1st Qu.: 6.90
##  Median :12.10   Median :24.60   Median : 9.70
##  Mean   :12.15   Mean   :24.97   Mean   : 8.72
##  3rd Qu.:16.90   3rd Qu.:30.00   3rd Qu.:11.10
##  Max.   :33.90   Max.   :48.90   Max.   :14.50
##  NA's   :518     NA's   :537     NA's   :59508
```

We can now access the variable `tmp`:

```
print(tmp)
```

```
## # A tibble: 112,447 x 4
##    rainfall min_temp max_temp sunshine
##       <dbl>    <dbl>    <dbl>    <dbl>
##  1        0      7.4     25.1       NA
##  2        0     12.9     25.7       NA
##  3        0      9.2     28         NA
##  4        0     14.3     25         NA
##  5        0      7.7     26.7       NA
##  6        0      9.7     31.9       NA
##  7        0     13.4     30.4       NA
##  8        0      8.4     24.6       NA
##  9        0     14.1     20.9       NA
## 10        0      9.8     25.6       NA
## # … with 112,437 more rows
```

Notice the use of the curly braces for the first path of the tee-pipe. This allows us to reference the data from the pipeline as the period (`.`).

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

3.16 Pipes: Tee Pipe Load CSV Files

20210103 This use case loads all `.csv.gz` files in the data folder into a single data frame and prints a message for each file loaded to monitor progress.

```
fpath <- "data"
files <- dir(fpath, "*.csv.gz")
ds    <- data.frame()
for (i in seq_len(length(files)))
{
  fpath %>%
    file.path(files[i]) %T>%
    cat("\n") %>%
    readr::read_csv() %>%
    rbind(ds, .) ->
  ds
}
```

This can be useful when combined with a sub pipeline which is introduced with curly braces. A global assignment operator saves the result for later. The example is a little contrived though illustrative. The sub-pipeline calculates the order of wind gust directions based on the maximum temperature for any day within each group defined by the wind direction. This is then used to order the wind directions, saving it in a global variable `lvls`. That variable is then used to mutate the original data frame (note the use of the tee pipe) to reorder the levels of the `wind_gust_dir` variable. This is typically done within a pipeline that feeds into a plot where we want to reorder the levels so that there is some meaning to the order of the bars in a bar plot, for example.

```
levels(ds$wind_gust_dir)
```

```
##  [1] "N"   "NNE" "NE"  "ENE" "E"   "ESE" "SE"  "SSE" "S"   "SSW" "SW"  "WSW"
## [13] "W"   "WNW" "NW"  "NNW"
```

```
ds %>%
  filter(rainfall>0) %T>%
  {
    select(., wind_gust_dir, max_temp) %>%
      group_by(wind_gust_dir) %>%
      summarise(max_max_temp=max(max_temp), .groups="drop") %>%
      arrange(max_max_temp) %>%
      pull(wind_gust_dir) ->>
    lvls
  } %>%
  mutate(wind_gust_dir=factor(wind_gust_dir, levels=lvls)) %$%
  levels(wind_gust_dir)
```

```
##  [1] "SW"  "NNE" "N"   "NE"  "ENE" "E"   "ESE" "SE"  "SSE" "S"   "SSW" "WSW"
## [13] "W"   "WNW" "NW"  "NNW"
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

3.17 Pipes and Plots

REVIEW A common scenario for pipeline processing is to prepare data for plotting. Indeed, plotting itself has a pipeline type concept where we build a plot by adding layers to it.

Below the rattle::weatherAUS dataset is stats::filter()ed for observations from four Australian cities. We stats::filter() observations that have missing values for the variable `Temp3pm` using an embedded pipeline. The embedded pipeline pipes the `Temp3pm` data through the base::is.na() function which tests if the value is missing. These results are then piped to magrittr::not() which inverts the true/false values so that we include those that are not missing.

A plot is generated using ggplot2::ggplot() into which we pipe the processed dataset. We add a geometric layer using ggplot2::geom_density() which consists of a density plot with transparency specified through the argument. We also add a title and label the axes using ggplot2::labs().

```
cities <- c("Canberra", "Darwin", "Melbourne", "Sydney")


ds %>%
  filter(location %in% cities) %>%
  filter(temp_3pm %>% is.na() %>% not()) %>%
  ggplot(aes(x=temp_3pm, colour=location, fill=location)) +
  geom_density(alpha=0.55) +
  labs(title = "Density Distributions of the 3pm Temperature",
       x     = "Temperature Recorded at 3pm",
       y     = "Density")
```

We now observe and tell a story from the plot. Our narrative will begin with the observation that Darwin has quite a different and warmer pattern of temperatures at 3pm than Canberra, Melbourne and Sydney. Canberra is on the colder side with Sydney generally warmer than Melbourne!

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

3.18 Variables

20200501 We can store values, such as the resulting value from invoking a function (Section 3.5), into a *variable*. A *variable* is a name that we can use to refer to some location in the computer's memory. Into this location we store the data while our programs are running so that we can refer to that data by the specific variable name later on. We use the *assignment operator* base::<- to do so:

```
# Add two numbers and assign the result into a variable.

v <- sum(1, 2)
```

We can access the value stored in the variable (or actually stored in the computer's memory that the variable name refers to) simply by requesting through the R `Console` that R `` `run'' the command `v` . We do so below with a line containing just `v` . In fact the R software actually runs the [base::print()] (https://www.rdocumentation.org/packages/base/topics/print){ target="_blank" } function to display the contents of the computer's memory that `v` refers to—this is a convenience that the R software provides for us.

```
# Print the value stored in a variable.

v
```

```
## [1] 3
```

```
print(v)
```

```
## [1] 3
```

By indexing the variable with `[1]` we can ask for the first item referred to by `v`. Noting that the variable `v` is a vector with only a single item when we try to index a second item we get an `NA` (meaning not available or a missing value).

```
# Access a particular value from a vector of data.

v[1]
```

```
## [1] 3
```

```
v[2]
```

```
## [1] NA
```

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

4 R Tasks

In this chapter we demonstrate different tasks that are generally language constructs that are not specifically data scientist activities but useful in coding.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

4.1 Create Directory

```
library(glue)
library(magrittr)

DEST <- glue("./backup/{Sys.Date() %>% format('%Y%m%d')}/")
dir.create(DEST, recursive=TRUE, showWarnings=FALSE)
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

4.2 File Exists

```
MYFILE <- "weatherAUS.csv"

if ( ! file.exists(MYFILE))
{
  system("wget https://rattle.togaware.com/weatherAUS.csv")
}
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

4.3 Format Date

```
library(magrittr)

Sys.Date() %>% format("%Y%m%d")
```

```
## [1] "20210624"
```

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

5 Dates

20210318 Wrangling dates is something we will find ourselves doing often as data scientists. R provides a comprehensive set of tools for handling and processing dates. In this chapter we review the functionality provided by R for managing and manipulating dates.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

5.1 Dates Setup

```
##
## Attaching package: 'scales'
```

```
## The following object is masked from 'package:readr':
##
##     col_factor
```

20210318 Packages used in this chapter include lubridate (Spinu, Grolemund, and Wickham 2021), magrittr (Bache and Wickham 2020), and rattle (G. Williams 2020).

Packages are loaded into the currently running R session from your local library directories on disk. Missing packages can be installed using utils::install.packages() within R. On Ubuntu, for example, R packages can also be installed using `$ wajig install r-cran-<pkgname>`.

```
# Load required packages from local library into the R session.

library(lubridate)    # Tidy dates and time.
```

The rattle::weatherAUS dataset is loaded into the template variable `ds` and further template variables are setup as introduced by Graham J. Williams (2017). See Chapter 8 for details.

```
dsname <- "weatherAUS"
ds     <- get(dsname)

nobs   <- nrow(ds)

vnames <- names(ds)
ds    %<>% clean_names(numerals="right")
names(vnames) <- names(ds)

vars   <- names(ds)
target <- "rain_tomorrow"
vars   <- c(target, vars) %>% unique() %>% rev()
```

References

Bache, Stefan Milton, and Hadley Wickham. 2020. *Magrittr: A Forward-Pipe Operator for r*. https://CRAN.R-project.org/package=magrittr.

Spinu, Vitalie, Garrett Grolemund, and Hadley Wickham. 2021. *Lubridate: Make Dealing with Dates a Little Easier*. https://CRAN.R-project.org/package=lubridate.

Williams, Graham. 2020. *Rattle: Graphical User Interface for Data Science in r*. https://rattle.togaware.com/.

Williams, Graham J. 2017. *The Essentials of Data Science: Knowledge Discovery Using r*. The r Series. CRC Press.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*

Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

5.2 Extract Year, Month and Day

```
td <- lubridate::today() %T>% print()
```

```
## [1] "2021-06-24"
```

```
y  <- lubridate::year(td) %T>% print()
```

```
## [1] 2021
```

```
m  <- lubridate::month(td) %T>% print()
```

```
## [1] 6
```

```
m  <- lubridate::month(td, label=TRUE, abbr=TRUE) %T>% print()
```

```
## [1] Jun
## 12 Levels: Jan < Feb < Mar < Apr < May < Jun < Jul < Aug < Sep < ... < Dec
```

```
d  <- lubridate::day(td) %T>% print()
```

```
## [1] 24
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

5.3 Format Dates

20210412 The function base::strptime() supports the date format strings. For example, based on *Thursday, 24 June 2021, 16:00:37 PM*:

- `%b` $\longrightarrow$ Jun
- `%Y` $\longrightarrow$ 2021

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

6 Strings

20200421 Wrangling strings of characters is something we will find ourselves doing often as data scientists. R provides a comprehensive set of tools for handling and processing strings. In this chapter we review the functionality provided by R for managing and manipulating strings.

Karthik Bharadwaj is a co-author of this chapter.

A freely available ebook by Gaston Sanchez, Handling and Processing Strings in R (2013) provides an extensive treatise on using strings in R.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

6.1 Strings Setup

20180720 Packages used in this chapter include dplyr (Wickham et al. 2021), glue (Hester 2020), magrittr (Bache and Wickham 2020), readr (Wickham and Hester 2020), stringr (Wickham 2019b), stringi (Gagolewski et al. 2021), scales (Wickham and Seidel 2020), and rattle (G. Williams 2020).

Packages are loaded into the currently running R session from your local library directories on disk. Missing packages can be installed using utils::install.packages() within R. On Ubuntu, for example, R packages can also be installed using `$ wajig install r-cran-<pkgname>`.

```
# Load required packages from local library into the R session.

library(dplyr)        # Wrangling: mutate().
library(stringi)      # The string concat operator %s+%.
library(stringr)      # String manipulation.
library(glue)         # Format strings.
library(magrittr)     # Pipelines for data processing: %>% %T>% %<>%.
library(rattle)       # Weather dataset.
library(readr)        # Read/write: read_csv().
library(scales)       # commas(), percent().
```

After loading the required packages into the library we access the rattle::weatherAUS dataset and save it into the template dataset named `ds`, as per the template based approach introduced by Graham J. Williams (2017). The dataset is modestly large and is used extensively in this book to illustrate the capabilities of R for the Data Scientist.

```
# Initialise the dataset as per the template.
dsname <- "weatherAUS"
ds     <- get(dsname)
names(ds) %<>% normVarNames()

ds %>% sample_frac()
```

```
## # A tibble: 176,747 x 24
##    date       location      min_temp max_temp rainfall evaporation sunshine
##    <date>     <chr>            <dbl>    <dbl>    <dbl>       <dbl>    <dbl>
##  1 2014-08-27 MountGinini       -1.3      4.9      0.6        NA       NA
##  2 2019-03-20 Walpole           18       19.9      0.2        NA       NA
##  3 2009-02-17 Newcastle         18.8     23.4      0          NA       NA
##  4 2014-09-29 Tuggeranong        8.2     25.4      0          NA       NA
##  5 2007-11-11 Canberra           9.1     25.2      0           4.2     11.9
##  6 2019-05-10 Woomera            4.6     16.2      3          NA       NA
##  7 2011-01-18 Portland          12.5     17.3      0.2         3.2      0.9
##  8 2015-11-08 Watsonia           6.2     25.7      0           4.2     10.9
##  9 2012-06-03 Mildura            3.1     14.8      0           1.6      8.7
## 10 2015-06-26 BadgerysCreek      7.2     18.1      1          NA       NA
## # … with 176,737 more rows, and 17 more variables: wind_gust_dir <ord>,
## #   wind_gust_speed <dbl>, wind_dir_9am <ord>, wind_dir_3pm <ord>,
## #   wind_speed_9am <dbl>, wind_speed_3pm <dbl>, humidity_9am <int>,
## #   humidity_3pm <int>, pressure_9am <dbl>, pressure_3pm <dbl>,
## #   cloud_9am <int>, cloud_3pm <int>, temp_9am <dbl>, temp_3pm <dbl>,
## #   rain_today <fct>, risk_mm <dbl>, rain_tomorrow <fct>
```

References

Bache, Stefan Milton, and Hadley Wickham. 2020. *Magrittr: A Forward-Pipe Operator for r*. https://CRAN.R-project.org/package=magrittr.

Gagolewski, Marek, Bartek Tartanus, and others; IBM, Unicode, Inc., and others. 2021. *Stringi: Character String Processing Facilities*. https://CRAN.R-project.org/package=stringi.

Hester, Jim. 2020. *Glue: Interpreted String Literals*. https://CRAN.R-project.org/package=glue.

———. 2019b. *Stringr: Simple, Consistent Wrappers for Common String Operations*. https://CRAN.R-project.org/package=stringr.

Wickham, Hadley, Romain François, Lionel Henry, and Kirill Müller. 2021. *Dplyr: A Grammar of Data Manipulation*. https://CRAN.R-project.org/package=dplyr.

Wickham, Hadley, and Jim Hester. 2020. *Readr: Read Rectangular Text Data*. https://CRAN.R-project.org/package=readr.

Wickham, Hadley, and Dana Seidel. 2020. *Scales: Scale Functions for Visualization*. https://CRAN.R-project.org/package=scales.

Williams, Graham. 2020. *Rattle: Graphical User Interface for Data Science in r*. https://rattle.togaware.com/.

Williams, Graham J. 2017. *The Essentials of Data Science: Knowledge Discovery Using r*. The r Series. CRC Press.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*

Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

6.2 Case Conversion

20180606 Often during data transformations strings have to be converted from one case to the other. These simple transformations can be achieved by base::tolower() and base::toupper(). The base::casefold() function can also be used as a wrapper to the two functions.

```
toupper("String Manipulation")
```

```
## [1] "STRING MANIPULATION"
```

```
tolower("String Manipulation")
```

```
## [1] "string manipulation"
```

```
casefold("String Manipulation")
```

```
## [1] "string manipulation"
```

```
casefold("String Manipulation", upper=TRUE)
```

```
## [1] "STRING MANIPULATION"
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open*

source software since 1984.

Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

6.3 Concatenate Strings

20180720 One of the most basic operations in string manipulation is the concatenate operation. R provides alternatives for doing so but a modern favourite is the `%s+` operator.

```
"abc" %s+% "def" %s+% "ghi"
```

```
## [1] "abcdefghi"
```

```
c("abc", "def", "ghi", "jkl") %s+% c("mno")
```

```
## [1] "abcmno" "defmno" "ghimno" "jklmno"
```

```
c("abc", "def", "ghi", "jkl") %s+% c("mno", "pqr")
```

```
## [1] "abcmno" "defpqr" "ghimno" "jklpqr"
```

```
c("abc", "def", "ghi", "jkl") %s+% c("mno", "pqr", "stu", "vwx")
```

```
## [1] "abcmno" "defpqr" "ghistu" "jklvwx"
```

The tidy function for concatenating strings is stringr::str_c(). A `sep=` can be used to specify a separator for the concatenated strings.

```
str_c("hello", "world")
```

```
## [1] "helloworld"
```

```
str_c("hello", "world", sep=" ")
```

```
## [1] "hello world"
```

We can also concatenate strings using glue::glue().

```
glue("hello", "world")
```

```
## helloworld
```

The traditional base::cat() function returns the concatenation of the supplied strings. Numeric and other complex objects are converted into character strings.

```
cat("hello", "world")
```

```
## hello world
```

```
cat ("hello", 123, "world")
```

```
## hello 123 world
```

Yet another alternative (and there are many) is the function base::paste(). Notice that it separates the concatenated strings with a space.

```
paste("hello", "world")
```

```
## [1] "hello world"
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

6.4 Concatenate NULL

20180720 Each operator/functions treats NULL differently. Note the convenience for base::cat() to add a space between the strings, and that base::paste() treats NULL as a zero length string, and thus there are two spaces between the words concatenated.

```
"hello" %s+% NULL %s+% "world"
```

```
## character(0)
```

```
str_c("hello", NULL, "world")
```

```
## [1] "helloworld"
```

```
glue("hello", NULL, "world")
```

```
cat("hello", NULL, "world")
```

```
## hello world
```

```
paste("hello", NULL, "world")
```

```
## [1] "hello  world"
```

NA tends to be treated differently too.

```r
"hello" %s+% NA %s+% "world"
```

```
## [1] NA
```

```r
str_c("hello", NA, "world")
```

```
## [1] NA
```

```r
glue("hello", NA, "world")
```

```
## helloNAworld
```

```r
cat("hello", NA, "world")
```

```
## hello NA world
```

```r
paste("hello", NA, "world")
```

```
## [1] "hello NA world"
```

The examples becomes more interesting in the context that the arguments to the functions might be string returning functions. If that function returns NULL or NA, purposely or accidentally then it is useful to know the consequences.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of*

Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

6.5 Glue Strings Together

20180729 The glue (Hester 2020) package provides a mechanism for building output strings from a collection of strings and variables. The basic use of glue::glue() will concatenate its string arguments with variable substitution identified using curly braces. In this exmaple we use the rattle::weatherAUS and format large numbers using scales::comma().

```
dsname <- "weatherAUS"
nobs   <- nrow(weatherAUS)
starts <- min(weatherAUS$Date)
glue("The {dsname} dataset",
     " has just less than {comma(nobs + 1)} observations,",
     " starting from {format(starts, '%-d %B %Y')}.")
```

```
## The weatherAUS dataset has just less than 176,748 observations, starting from 1 N
```

We can manually wrap the sentence.

```
glue("
     The {dsname} dataset has just
     less than {comma(nobs + 1)} observations
     starting from {format(starts, '%-d %B %Y')}.
     ")
```

```
## The weatherAUS dataset has just
## less than 176,748 observations
## starting from 1 November 2007.
```

Notice how the initial and last empty lines are handled as expected'', and the line split is maintained.

Named arguments within the function call can be used to assign values to variables that only exist in the scope of the function call.

```
glue("
     The {dsname} dataset has just
         less than {comma(nobs + 1)} observations
     starting from {format(starts, '%-d %B %Y')}.
     ",
     dsname = "weather",
     nobs   = nrow(weather),
     starts = min(weather$Date))
```

```
## The weather dataset has just
##     less than 367 observations
## starting from 1 November 2007.
```

We can also see the effect of indenting lines in this example, where the indentation is retained.

References

Hester, Jim. 2020. *Glue: Interpreted String Literals*. https://CRAN.R-project.org/package=glue.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

6.6 Glue Pipelines

20180729 We can use glue::glue_data() within pipes and operate over the rows of the data that is piped into the operator.

```
weatherAUS %>%
  sample_n(6) %>%
  glue_data("Observation",
            " {rownames(.) %>% as.integer() %>% comma() %>% sprintf('%7s', .)}",
            " location {Location %>% sprintf('%-14s', .)}",
            " max temp {MaxTemp %>% sprintf('%5.1f', .)}")
```

```
## Observation     1.0 location Perth          max temp  24.9
## Observation     2.0 location SydneyAirport  max temp  22.9
## Observation     3.0 location Canberra       max temp  44.0
## Observation     4.0 location Williamtown    max temp  21.0
## Observation     5.0 location Hobart         max temp  18.5
## Observation     6.0 location SydneyAirport  max temp  21.8
```

It can also be useful with the tidy verse work flow.

```
weatherAUS %>%
  sample_n(6) %>%
  mutate(TempRange = glue("{MinTemp}-{MaxTemp}")) %>%
  glue_data("Observed temperature range at {Location} of {TempRange}")
```

```
## Observed temperature range at MountGambier of 9-15.7
## Observed temperature range at Nuriootpa of 18.5-38.4
## Observed temperature range at Watsonia of 5.9-14.8
## Observed temperature range at Ballarat of 4.9-18.5
## Observed temperature range at Hobart of 11.6-17.7
## Observed temperature range at Watsonia of 8.6-32.5
```

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

6.7 Last Character of a String

```
str_sub("Hello World!", -1)
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

6.8 Length of String

20180606 The tidy way to get the length of a string is stringr::str_length().

```
str_length("hello world")
```

```
## [1] 11
```

```
str_length(c("hello", "world"))
```

```
## [1] 5 5
```

```
str_length(NULL)
```

```
## integer(0)
```

```
str_length(NA)
```

```
## [1] NA
```

The function base::nchar() is the traditional approach.

```
nchar("hello world")
```

```
## [1] 11
```

```r
nchar(c("hello", "world"))
```

```
## [1] 5 5
```

```r
nchar(NULL)
```

```
## integer(0)
```

```r
nchar(NA)
```

```
## [1] NA
```

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

6.9 Random Strings

20180604 It is sometimes very useful to be able to test out some code using some test data. A simple way to generate test strings us with stringi::stri_rand_lipsum().

```
stri_rand_lipsum(20)
```

```
##  [1] "Lorem ipsum dolor sit amet, tempor, donec eu arcu montes montes nisl. Sed t
##  [2] "Proin nisl curae primis tincidunt at. Eleifend, sed tortor amet ex, bibendu
##  [3] "Mollis eros venenatis tortor sit taciti suspendisse tortor tristique curae
##  [4] "Donec turpis a ut. Ultrices, magna praesent tristique, ac a ridiculus. Non
##  [5] "Egestas mus posuere class tempus nunc bibendum dignissim. Amet suspendisse
##  [6] "Tellus, vel in ligula, sed interdum, eleifend vitae. Class leo et varius, v
##  [7] "Nulla tortor ex ullamcorper, mus pulvinar enim leo. Pellentesque hendrerit
##  [8] "Facilisis eleifend montes. Sollicitudin quis mauris tortor sapien, in et se
##  [9] "Rhoncus lorem sed sagittis velit. Consectetur quis sem sed urna vitae vehic
## [10] "Accumsan ac curabitur tempus a leo dapibus sed erat. Suscipit tempor litora
## [11] "Lobortis sem non sodales interdum vehicula litora ut vivamus vivamus. Amet
## [12] "Ridiculus consectetur sem ultricies consequat sapien nulla enim leo dictums
## [13] "Scelerisque sed. In amet venenatis, magnis, hendrerit, donec etiam nulla. T
## [14] "Natoque facilisi, neque ornare maecenas eu dapibus nunc cum commodo, tristi
## [15] "Tincidunt ac nostra pulvinar urna turpis. Lacinia orci porttitor nisi rutru
## [16] "Ridiculus pulvinar vel ut inceptos praesent, commodo. Congue lobortis torqu
## [17] "Nostra libero quis taciti convallis adipiscing elit interdum. Nec lobortis

## [18] "At nascetur quam ut nulla ligula at accumsan nisi iaculis. Primis tristique
## [19] "Tincidunt pretium lacinia senectus nibh lacus arcu. Ac posuere leo volutpat
## [20] "Ac rhoncus nisi purus praesent blandit venenatis. At consequat ultricies nu
```

```
stri_rand_lipsum(2)
```

```
## [1] "Lorem ipsum dolor sit amet, phasellus ex ut augue. Velit class nunc eget ap
## [2] "Quam ut curae vitae dictumst. Est sodales ac. Gravida aliquam arcu sem ac, 
```

```
sapply(stri_rand_lipsum(10), nchar, USE.NAMES=FALSE)
```

```
##  [1] 852 477 402 566 754 430 718 384 729 554
```

```
sapply(stri_rand_lipsum(10), nchar, USE.NAMES=FALSE)
```

```
##  [1] 741 566 424 440 647 939 585 787 432 490
```

The strings generated are of different lengths and each call generates different strings.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

6.10 Regexp Pattern Matching

20180608 One of the most powerful string processing concepts is the concept of regular expressions. A regular expression is a sequence of characters that describe a pattern. The concept was formalised by the American mathematician Stephen Cole Kleene. A regular expression pattern can contain a combination of alphanumeric and special characters. It is a complex topic and we take an introductory look at it here to craft regular expressions in R.

Visit RegExr, by G Skinner to explore regular expressions interactively.

An important concept is that of metacharacters which have special meaning within a regular expression. Unlike other characters that are used to match themselves, metacharacters have a specific meaning beyond the character they represent. The following table contains a list of common metacharacters used in regular expressions.

% latex table generated in R 4.1.0 by xtable 1.8-4 package % Thu Jun 24 16:00:38 2021
Such metacharacters are used to match different patterns which can be found using base::grep(). According to <gnu.org/software/grep> `g/re/p` is a command from the command line tool `ed` to *g*et the *re*gular expression and *p*rint it.

```
s <- c("hands", "data", "on", "data$cience", "handsondata$cience", "handson")
grep(pattern="^data", s, value=TRUE)
```

```
## [1] "data"        "data$cience"
```

```
grep(pattern="on$", s, value=TRUE)
```

```
## [1] "on"      "handson"
```

```
grep(pattern="(nd)..(nd)", s, value=TRUE)
```

```
## [1] "handsondata$cience"
```

In order to match a metacharacter in R we need to escape it with \\ (double backslash).

```
grep(pattern="\\$", s, value=TRUE)
```

```
## [1] "data$cience"        "handsondata$cience"
```

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

6.11 Regexp Quantifiers

20180608 Quantifiers are used to match repitition of a pattern within a string. The following table shows a list of quantifiers.

% latex table generated in R 4.1.0 by xtable 1.8-4 package % Thu Jun 24 16:00:39 2021
Some examples will illustrate.

```
s <- c("aaab", "abb", "bc", "abbcd", "bbbc", "abab", "caa")
grep(pattern="ab*b", s, value=TRUE)
```

```
## [1] "aaab"  "abb"   "abbcd" "abab"
```

```
grep(pattern="abbc?", s, value=TRUE)
```

```
## [1] "abb"   "abbcd"
```

```
grep(pattern="b{2,}?", s, value=TRUE)
```

```
## [1] "abb"   "abbcd" "bbbc"
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

6.12 Regexp Character Classes

20180608 A character class is a collection of characters that are in some way grouped together. We enclose the characters to be grouped within square backets []. The pattern then matches any one of the characters in the set. For example, the character class [0-9] matches any of the digits from 0 to 9.

% latex table generated in R 4.1.0 by xtable 1.8-4 package % Thu Jun 24 16:00:39 2021

```
s <- c("abc12", "@#$", "345", "ABcd")
grep(pattern="[0-9]+", s, value=TRUE)
```

```
## [1] "abc12" "345"
```

```
grep(pattern="[A-Z]+", s, value=TRUE)
```

```
## [1] "ABcd"
```

```
grep(pattern="[^@#$]+", s, value=TRUE)
```

```
## [1] "abc12" "345"   "ABcd"
```

R also supports the use of POSIX character classes which are represented within [[]] (double braces).

```
grep(pattern="[[:alpha:]]", s, value=TRUE)
```

```
## [1] "abc12" "ABcd"
```

```r
grep(pattern="[[:upper:]]", s, value=TRUE)
```

```
## [1] "ABcd"
```

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

6.13 Sub Strings in Base R

20210330 The base function base::substr() can be used to extract and replace parts of a string similar to stringr::str_sub() in the tidyverse. Note however that it does not handle negative values and that string replacement only replaces the same length as the replacement string, without changing the length of the original string.

```
s <- "string manipulation"
substr(s, start=3, stop=6)
```

```
## [1] "ring"
```

```
substr(s, 3, 6)
```

```
## [1] "ring"
```

```
substr(s, 1, 12) <- "stip"
s
```

```
## [1] "stipng manipulation"
```

The base::substring() function performs similarly though uses *last*= rather than *stop*=.

```
s <- "string manipulation"
substring(s, first=3, last=6)
```

```
## [1] "ring"
```

```r
x <- c("abcd", "aabcb", "babcc", "cabcd")
substring(x, 2, 4)
```

```
## [1] "bcd" "abc" "abc" "abc"
```

```r
substring(x, 2, 4) <- "AB"
x
```

```
## [1] "aABd"  "aABcb" "bABcc" "cABcd"
```

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

6.14 Sub Strings in the Tidyverse

20180607 We will find ourselves often wanting to extract or modify sub-strings within a string. The tidy way to do this is with the stringr::str_sub() function. We can specify the *start*= and the *end*= of the string. The indices of the string start from 1.

```
s <- "string manipulation"
str_sub(s, start=3, end=6)
```

```
## [1] "ring"
```

```
str_sub(s, 3, 6)
```

```
## [1] "ring"
```

A negative is used to count from the end of the string.

```
str_sub(s, 1, -8)
```

```
## [1] "string manip"
```

Replacing a sub-string with another string is straightforward using the assignment operator.

```
str_sub(s, 1, -8) <- "stip"
s
```

```
## [1] "stipulation"
```

The function also operates over a vector of strings.

```
v <- c("string", "manipulation", "always", "fascinating")
str_sub(v, -4, -1)
```

```
## [1] "ring" "tion" "ways" "ting"
```

```
str_sub(v, -4, -1) <- "RING"
v
```

```
## [1] "stRING"       "manipulaRING" "alRING"       "fascinaRING"
```

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

6.15 Substitute Strings

```
str_replace(v, '^..', '')
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

6.16 Trim and Pad

20180608 One of the major challenges of string parsing is removing and adding whitespaces and wrapping text.

Additional white space can be present on the left, right or both sides of the word. The stringr::str_trim() function offers an effective way to get rid of these whitespaces.

```
ws <- c(" abc",  "def ", " ghi ")
str_trim(ws)
```

```
## [1] "abc" "def" "ghi"
```

```
str_trim(ws, side="left")
```

```
## [1] "abc"  "def " "ghi "
```

```
str_trim(ws, side="right")
```

```
## [1] " abc" "def"  " ghi"
```

```
str_trim(ws, side="both")
```

```
## [1] "abc" "def" "ghi"
```

Conversely we can also pad a string with additional characters for up to a specified width using stringr::str_pad(). The default padding character is a space but we can override that.

```
str_pad("abc", width=7)
```

```
## [1] "    abc"
```

```
str_pad("abc", width=7, side="left")
```

```
## [1] "    abc"
```

```
str_pad("abc", width=7, side="right")
```

```
## [1] "abc    "
```

```
str_pad("abc", width=7, side="both", pad="#")
```

```
## [1] "##abc##"
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

6.17 Wrapping and Words

20180608 Formatting a text string into a neat paragraph of defined maximum width is another operation we often find ourselves wanting. The stringr::str_wrap() function will do this for us.

```
st <- "All the Worlds a stage, All men are merely players"
cat(str_wrap(st, width=25))
```

```
## All the Worlds a stage,
## All men are merely
## players
```

Words of course form the basis for wrapping a sentence. We may wish to extract words from a sentence ourselves for further processing. Here we us stringr::word() to do so.We specify the positions of the word to be extracted from the setence. The default separator value is space.

```
st <- c("The quick brown fox", "jumps on the brown dog")
word(st, start=1, end=2)
```

```
## [1] "The quick" "jumps on"
```

```
word(st, start=1, end=-2)
```

```
## [1] "The quick brown"    "jumps on the brown"
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of*

Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

7 R Read, Write, and Create

20200104 Data is available in an enormous variety of formats and stored in a diverse number of locations, including our own data stores, local disks, cloud stores, database systems, and from the Internet sites like , , and . A major task we face as Data Scientists is ingesting that data into R in order for us to perform our analyses. R provides extensive capabilities for *reading* (aka *importing* and *ingesting*) data and for *writing* (aka *exporting*) data.

Here we explore the options, including a simple and widely used format known as comma separated values (CSV) files. A variation of this is the tab separated values, or indeed other special characters used to separate the columns of the data.

But data comes in an amazing variety of formats, including many proprietary formats that need special effort to decipher. R supports almost every known format through many different packages. This chapter introduces the numerous options available.

Also included is a guide to creating your own random dataset for testing ideas and building systems where the actual data may not be so readily available due to privacy, for example.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

7.1 Data Creation Setup

20201022 Packages used in this chapter include dplyr (Wickham et al. 2021), magrittr (Bache and Wickham 2020), randomForest (Breiman et al. 2018), readxl (Wickham and Bryan 2019), wakefield (Rinker 2020b), xlsx (Dragulescu and Arendt 2020), and rattle (G. Williams 2020).

Packages are loaded into the currently running R session from your local library directories on disk. Missing packages can be installed using utils::install.packages() within R. On Ubuntu, for example, R packages can also be installed using `$ wajig install r-cran-<pkgname>`.

```
# Load required packages from local library into the R session.

library(dplyr)        # Wrangling: select() sample_frac().
library(magrittr)     # Data pipelines: %>% %<>% %T>% equals().
library(randomForest) # Model: randomForest() na.roughfix() for missing data.
library(readxl)       # Read Excel spreadsheets: read_excel().
library(wakefield)    # Generate random datasets.
library(xlsx)         # Write Excel spreadsheets: write.xlsx() saveWorkbook().
library(rattle)       # Dataset: weather.
```

After loading the required packages into the library we access the rattle::weatherAUS dataset and save it into the template dataset named `ds`, as per the template based approach introduced in Graham J. Williams (2017). The dataset is modestly large and is used extensively in this book to illustrate the capabilities of R for the Data Scientist.

```
# Initialise the dataset as per the template.

ds <- weatherAUS

ds %>% sample_frac()
```

```
## # A tibble: 176,747 x 24
##    Date       Location MinTemp MaxTemp Rainfall Evaporation Sunshine WindGustDir
##    <date>     <chr>      <dbl>   <dbl>    <dbl>       <dbl>    <dbl> <ord>
##  1 2015-01-21 SydneyA…    19.5    29.1     13           7.4     10.5 ENE
##  2 2014-09-22 AliceSp…     4.7    27.1      0           9       11.5 E
##  3 2014-01-09 Portland    13.2    24        0           7.6     13.3 E
##  4 2009-04-11 PerthAi…    14.1    31.1      0           5.6     10   SW
##  5 2016-01-25 Canberra    17      27        0          NA       NA   <NA>
##  6 2015-08-27 Penrith     10.4    20.6      4          NA       NA   NW
##  7 2020-03-09 GoldCoa…    20.8    23        5          NA       NA   SE
##  8 2012-04-26 Melbour…    10.3    17.2      2.2         0.2      5.6 WSW
##  9 2012-04-22 MountGa…    15      18.9      4.2         6.6      8   NNW
## 10 2020-04-10 Norfolk…    17.8    22.5      0          NA       NA   SSE
## # … with 176,737 more rows, and 16 more variables: WindGustSpeed <dbl>,
## #   WindDir9am <ord>, WindDir3pm <ord>, WindSpeed9am <dbl>, WindSpeed3pm <dbl>,
## #   Humidity9am <int>, Humidity3pm <int>, Pressure9am <dbl>, Pressure3pm <dbl>,
## #   Cloud9am <int>, Cloud3pm <int>, Temp9am <dbl>, Temp3pm <dbl>,
## #   RainToday <fct>, RISK_MM <dbl>, RainTomorrow <fct>
```

References

Bache, Stefan Milton, and Hadley Wickham. 2020. *Magrittr: A Forward-Pipe Operator for r*. https://CRAN.R-project.org/package=magrittr.

Breiman, Leo, Adele Cutler, Andy Liaw, and Matthew Wiener. 2018. *randomForest: Breiman and Cutler's Random Forests for Classification and Regression*. https://www.stat.berkeley.edu/~breiman/RandomForests/.

Dragulescu, Adrian, and Cole Arendt. 2020. *Xlsx: Read, Write, Format Excel 2007 and Excel 97/2000/XP/2003 Files*. https://github.com/colearendt/xlsx.

———. 2020b. *Wakefield: Generate Random Data Sets*. https://github.com/trinker/wakefield.

Wickham, Hadley, and Jennifer Bryan. 2019. *Readxl: Read Excel Files*. https://CRAN.R-project.org/package=readxl.

Wickham, Hadley, Romain François, Lionel Henry, and Kirill Müller. 2021. *Dplyr: A Grammar of Data Manipulation*. https://CRAN.R-project.org/package=dplyr.

Williams, Graham. 2020. *Rattle: Graphical User Interface for Data Science in r*. https://rattle.togaware.com/.

Williams, Graham J. 2017. *The Essentials of Data Science: Knowledge Discovery Using r*. The r Series. CRC Press.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*

Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

7.2 Clipboard Data

20200106 A useful trick to quickly ingest a small amount of data into R is to read the data from the clipboard. That is, from a selection that is made in some other application. For example, if a CSV file is being viewed within a text editor, then some number of rows and columns could be highlighted and copied (often by pressing Ctrl-C) to the clipboard. The data can then be read into R using the package readr (Wickham and Hester 2020). The function readr::read_csv() can have as its first argument, `file=` readr::clipboard(), which diverts input to come from the clipboard. Another use case is when a small data sample is shared on a web page. The data can be highlighted and copied to the clipboard to then be ingested into R.

Note the use below of `col_names=`. Set to `FALSE` it indicates that the first row is not the column names, as might typically be the case when selecting different ranges of rows from sample data. Names like `X1` and `X2` will be generated. If the first row does name the columns then simply remove the option or else set it to `TRUE`. It can also be a character vector of names to be used for the columns.

```
library(readr)        # Modern and efficient data reader.

ds <- read_csv(clipboard(), col_names=FALSE)

ds
```

```
## # A tibble: 11 x 2
##    X1                X2
##    <chr>          <dbl>
##  1 27 Jun 2020 -2149
##  2 09 Jun 2020    33.8
##  3 11 Jun 2020    56.3
##  4 11 Jun 2020    12.1
##  5 11 Jun 2020    10.6
##  6 11 Jun 2020    47.4
##  7 11 Jun 2020    41.7
##  8 12 Jun 2020    45.6
##  9 15 Jun 2020  1838.
## 10 15 Jun 2020    50.3
## 11 17 Jun 2020    58.8
```

Using base R a similar approach is available with the special string *clipboard* as the first argument in utils::read.table(), for example. This generally only works on Linux with X11.

```
ds <- read.table("clipboard")
```

References

Wickham, Hadley, and Jim Hester. 2020. *Readr: Read Rectangular Text Data*. https://CRAN.R-project.org/package=readr.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

7.3 CSV Data

20201022 One of the simplest and most common ways of sharing data today remains the Comma Separated Values (CSV) format. As a simple and even trivial format the use of **csv** files has become a standard file format used to exchange data between many different applications.

csv files can, for example, be exported and imported by numerous applications and spreadsheets and databases, including Rattle, LibreOffice Calc, Gnumeric, MS/Excel, SAS/Enterprise Miner, Teradata, Netezza, and many, many, other applications.

The downside of the **csv** format is that the file does not contain explicit metadata (i.e., data about the data), like the data types of the different columns. Typically we would like to know whether there is numeric or character data within the column. If numeric data then are they dates, or dollars, and if character are they categoric (factors). Without this metadata R commands for reading the data have to make a guess and will sometimes determine the wrong data type for a particular column. There are options available to reduce this possibility or to provide the metadata.

Reading **csv** files is straight forward using readr::read_csv().

```
library(readr)        # Modern and efficient data reader/writer.

ds <- read_csv("mydata.csv")
```

Column types can be specified using `col_types="inffcD"`. The string can contain any of c (character), i (integer), n (numeric), d (double), l (logical), f (factor), D (date), T (data and time), t (time), ? (guess), _, (ignore).

Writing a dataset to a **csv** file is straightforward using readr::write_csv():

```
library(rattle)       # Dataset: weatherAUS.
library(dplyr)        # Wrangling: select().
library(readr)        # Modern and efficient data reader/writer.

ds <- weatherAUS

fname  <- "temperatureAUS.csv"

ds %>%
  select(Date, Location, MinTemp, MaxTemp, Temp9am, Temp3pm) %>%
  write_csv(fname)
```

To turn off the messaging of the identified columns, set the option for the number of columns to report to 0:

```
options(readr.num_columns=0)
```

To turn it back on, set it to NULL.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

7.4 Excel Data Read

20200104 Microsoft Excel spreadsheets are supported by readxl (Wickham and Bryan 2019) which provides readxl::read_excel(). A common requirement is to `skip=` the first few lines of the spreadsheet which might be taken up with logos and file meta data. A specific sheet can be chosen with `sheet=``2` to select the second sheet or `sheet=``"expenses"` to select a named sheet. A specific range within a sheet is selected using `range=`. The package also provides readxl::excel_format() and readxl::excel_sheets().

Below we read data from the `Sydney` tab of the `weatherAUS.xlsx` spreadsheet created in Section @ref(ingestion:write_excel).

```
library(magrittr)     # Data pipelines: %>% %<>% %T>% equals().
library(glue)         # Format strings: glue().
library(readxl)       # Read Excel spreadsheets: read_excel().

dsname <- "weatherAUS"
dstype <- "xlsx"
fsep   <- .Platform$file.sep

getwd() %>%
  glue("{fsep}{dsname}.{dstype}") %T>%
  print() ->
dspath
```

```
## /home/gjw/git/bitbucket/kayontoga/onepager/weatherAUS.xlsx
```

```
dspath %>% excel_format()
```

```
## [1] "xlsx"
```

```
dspath %>% excel_sheets()
```

```
## [1] "Sydney"
```

```
dspath %>%
  read_excel(sheet="Sydney") %>%
  assign(dsname, ., globalenv())
```

References

Wickham, Hadley, and Jennifer Bryan. 2019. *Readxl: Read Excel Files*. https://CRAN.R-project.org/package=readxl.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

7.5 Excel Data Write

20200922 An Excel spreadsheet can be written using writexl (Ooms 2021). The basic functionality of writexl::write_xlsx() is to write a data frame to the named spreadsheet file. If a list of data frames is provided, each data frame populates a separate named sheet for each data frame.

Using rattle::weatherAUS we will create a spreadsheet by splitting the data frame into separate data frames, one per location, as a list. For each location a new sheet (tab) is created.

```
library(writexl)         # Write Excel spreadsheets: write_xlsx()


weatherAUS %>%
  group_by(Location) %>%
  group_split() %>%
  set_names(weatherAUS$Location %>% unique() %>% sort()) %>%
  write_xlsx('weatherAUS.xlsx')
```

See Section @ref(ingestion:read_excel) for an example of using this created spreadsheet.

References

Ooms, Jeroen. 2021. *Writexl: Export Data Frames to Excel Xlsx Format*. https://CRAN.R-project.org/package=writexl.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

7.6 MATLAB Data

```
library(R.matlab)

ds <- readMat("mydata.mat")
```

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

7.7 Random Dataset

20200421 Using wakefield::r_data_frame() it is straightforward to create simple random data datasets. More sophisticated sampling can be used if required though extra work is required to get correlated distributions across variables when looking to most accurately match some real world data. Nonetheless, creating random data quickly is the forte of wakefield (Rinker 2020b).

```
library(wakefield)    # Generate random datasets.

nobs <- 100 # Number of observatios.

r_data_frame(n=nobs,
             id=id_factor,
             givennames=name,
             surname=name,
             dob=dob(start=Sys.Date()-365*100, k=365*99),
             sex=sex,
             phone=r_sample(x=400000000:499999999),
             diabetes=answer,
             result=r_sample_factor(x=c("pending", "positive", "negative")),
             previous=answer(prob=c(0.8,0.2)),
             bps=r_sample(90:135),
             temp=normal(36.3, 1.2),
             gcs=r_sample(3:15, prob=c(rep(0.01, 12), 0.88)),
             crp=normal(2.5, 2, min=0, max=200)) ->
random_ds

glimpse(random_ds)
```

```
## Rows: 100
## Columns: 13
## $ id         <fct> 001, 002, 003, 004, 005, 006, 007, 008, 009, 010, 011, 012,…
## $ givennames <chr> "Kayjuan", "Deerica", "Annajean", "Janetta", "Eisha", "Leas…
## $ surname    <chr> "Namyia", "Pharyn", "Elissabeth", "Alitza", "Lauire", "Jany…
## $ dob        <date> 2014-01-14, 1961-02-05, 1925-10-30, 1933-01-12, 1949-10-11…
## $ sex        <fct> Male, Female, Female, Male, Female, Female, Female, Female,…
## $ phone      <int> 494438542, 437279128, 423569925, 467982485, 453331629, 4002…
## $ diabetes   <fct> Yes, No, No, No, No, No, Yes, Yes, Yes, Yes, No, Yes, Yes, …
## $ result     <fct> negative, pending, negative, negative, pending, negative, p…
## $ previous   <fct> No, Yes, Yes, No, Yes, No, No, No, Yes, No, No, No, No, No,…
## $ bps        <int> 90, 128, 135, 117, 124, 115, 114, 103, 126, 109, 115, 114, …
## $ temp       <dbl> 36.88249, 37.32358, 35.64939, 36.11359, 36.57237, 37.49851,…
## $ gcs        <int> 15, 15, 15, 15, 15, 15, 5, 15, 6, 15, 13, 15, 15, 15, 15, 1…
## $ crp        <dbl> 2.3517065, 1.4579896, 3.7510125, 3.2607771, 0.0000000, 0.00…
```

References

———. 2020b. *Wakefield: Generate Random Data Sets*. https://github.com/trinker/wakefield.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

7.8 Read Strings from a File

20180604 There may be occasions where we would like to load a dataset from a file as strings, one line as a string, returning a vector of strings. We achieve this using the function base::readLines(). In the following example we access the system file `weather.csv` that is provided by the rattle (G. Williams 2020) package.

```
library(glue)         # Format strings: glue().

dsname <- "weather"   # Dataset name.
ftype  <- "csv"       # Source dataset file type.
fname <- glue("{dsname}.{ftype}")

fname %T>%
  print() %>%
  system.file(ftype, ., package="rattle") %>%
  readLines() ->
ds
```

```
## weather.csv
```

Show the first few lines using utils::head():

```
head(ds)
```

```
## [1] "\"Date\",\"Location\",\"MinTemp\",\"MaxTemp\",\"Rainfall\",\"Evaporation\",\
## [2] "2007-11-01,\"Canberra\",8,24.3,0,3.4,6.3,\"NW\",30,\"SW\",\"NW\",6,20,68,29,
## [3] "2007-11-02,\"Canberra\",14,26.9,3.6,4.4,9.7,\"ENE\",39,\"E\",\"W\",4,17,80,3
## [4] "2007-11-03,\"Canberra\",13.7,23.4,3.6,5.8,3.3,\"NW\",85,\"N\",\"NNE\",6,6,82
## [5] "2007-11-04,\"Canberra\",13.3,15.5,39.8,7.2,9.1,\"NW\",54,\"WNW\",\"W\",30,24
## [6] "2007-11-05,\"Canberra\",7.6,16.1,2.8,5.6,10.6,\"SSE\",50,\"SSE\",\"ESE\",20,
```

Find those strings that contain a specific pattern using base::grep().

```r
grep("ENE", ds)
```

```
##  [1]   3  10  23  26  28  36  37  42  43  49  50  54  68  69  71  76  86  91  97
## [20] 101 103 106 108 109 110 118 129 132 133 135 138 145 160 171 176 215 222 278
## [39] 303 304 310 323 341 348 351 357 365
```

```r
grep("ENE", ds, value=TRUE)
```

```
##  [1] "2007-11-02,\"Canberra\",14,26.9,3.6,4.4,9.7,\"ENE\",39,\"E\",\"W\",4,17,80,
##  [2] "2007-11-09,\"Canberra\",8.8,19.5,0,4,4.1,\"S\",48,\"E\",\"ENE\",19,17,70,48
##  [3] "2007-11-22,\"Canberra\",16.4,19.4,0.4,9.2,0,\"E\",26,\"ENE\",\"E\",6,11,88,
##  [4] "2007-11-25,\"Canberra\",15.4,28.4,0,4.4,8.1,\"ENE\",33,\"SSE\",\"NE\",9,15,
##  [5] "2007-11-27,\"Canberra\",13.3,22.2,0.2,6.6,2.3,\"ENE\",39,\"E\",\"E\",20,17,
##  [6] "2007-12-05,\"Canberra\",14.5,21.8,0,8.4,9.8,\"ENE\",43,\"ESE\",\"E\",11,30,
##  [7] "2007-12-06,\"Canberra\",16.3,26.8,0,6,6.3,\"ENE\",39,\"ESE\",\"ESE\",13,9,7
##  [8] "2007-12-11,\"Canberra\",11.8,18.5,0.6,4.8,2.3,\"ENE\",35,\"ESE\",\"E\",9,15
##  [9] "2007-12-12,\"Canberra\",11.7,21.5,0,4.2,7.3,\"ENE\",41,\"ESE\",\"E\",15,24,
## [10] "2007-12-18,\"Canberra\",7.5,20.9,0,6.6,8.7,\"ENE\",39,\"SE\",\"E\",13,20,61
## [11] "2007-12-19,\"Canberra\",12.8,21,0,6.4,0.8,\"NE\",22,\"NE\",\"ENE\",7,6,72,5
## [12] "2007-12-23,\"Canberra\",9.2,20.4,17.4,7.8,10.2,\"ENE\",39,\"N\",\"N\",9,20,
## [13] "2008-01-06,\"Canberra\",14.3,34.1,0,6.6,10.5,\"ENE\",39,\"W\",\"NNW\",6,19,
## [14] "2008-01-07,\"Canberra\",16.5,33.9,0,9,12.6,\"ENE\",39,\"E\",\"NW\",11,11,60
## [15] "2008-01-09,\"Canberra\",17.5,29.9,0,6.6,8.8,\"E\",43,\"E\",\"ENE\",13,11,74
## [16] "2008-01-14,\"Canberra\",16,22.8,0,12.4,6,\"E\",50,\"E\",\"ENE\",13,19,70,57
## [17] "2008-01-24,\"Canberra\",10.3,27.8,0,6.4,9.9,\"ENE\",35,\"SE\",\"NE\",6,11,6
## [18] "2008-01-29,\"Canberra\",17.9,33.9,0,10.4,11.8,\"ENE\",46,\"S\",\"W\",6,11,7
## [19] "2008-02-04,\"Canberra\",18.2,22.6,1.8,8,0,\"ENE\",33,\"SSE\",\"ENE\",7,13,9
## [20] "2008-02-08,\"Canberra\",12.4,19.9,16.2,5.4,5.6,\"ENE\",41,\"ESE\",\"ESE\",7
## [21] "2008-02-10,\"Canberra\",9.1,23.1,0,5.8,9.6,\"ENE\",41,\"SSE\",\"W\",17,13,6
## [22] "2008-02-13,\"Canberra\",12.6,18.2,11,3.2,0.4,\"ENE\",30,\"SSE\",\"SSE\",13,
## [23] "2008-02-15,\"Canberra\",10.8,25.2,0,5.6,12.6,\"ENE\",35,\"SE\",\"E\",7,7,69
## [24] "2008-02-16,\"Canberra\",11.2,26.1,0,7.2,12.6,\"ENE\",39,\"SE\",\"E\",7,19,7
## [25] "2008-02-17,\"Canberra\",12.1,24.1,0,7.4,10.2,\"ENE\",46,\"ESE\",\"NNE\",9,1
## [26] "2008-02-25,\"Canberra\",11.5,25.9,0,5.2,10.2,\"ENE\",44,\"ENE\",\"WSW\",9,7
## [27] "2008-03-07,\"Canberra\",10.8,29.2,0,8.4,7.5,\"E\",50,NA,\"ENE\",NA,9,56,34,
## [28] "2008-03-10,\"Canberra\",12.5,31.7,0,6.6,11.2,\"WNW\",24,\"ENE\",\"W\",6,9,7
## [29] "2008-03-11,\"Canberra\",13.9,34.7,0,6.4,8.5,\"SSW\",46,\"SE\",\"ENE\",11,7,
## [30] "2008-03-13,\"Canberra\",13.2,33.1,0,8.6,9.7,\"ENE\",39,\"SE\",\"NNW\",7,7,9
## [31] "2008-03-16,\"Canberra\",11.3,32.3,0,9.4,11.4,\"NE\",28,\"ENE\",\"WNW\",4,6,
## [32] "2008-03-23,\"Canberra\",12.8,24.9,0,2.4,6.2,\"ENE\",30,\"NNW\",\"SW\",6,7,7
## [33] "2008-04-07,\"Canberra\",8.1,20.5,0,3.8,7.8,\"ENE\",31,\"ESE\",\"ENE\",11,11
## [34] "2008-04-18,\"Canberra\",7.9,19.7,0,3.2,8.3,\"ESE\",48,\"SSE\",\"ENE\",17,11
## [35] "2008-04-23,\"Canberra\",7.5,19,0,4,6.8,\"ENE\",26,NA,\"ESE\",0,7,74,43,1025
```

```
## [36] "2008-06-01,\"Canberra\",0.6,14,0,2.2,2.7,\"NE\",17,NA,\"ENE\",NA,7,99,67,1
## [37] "2008-06-08,\"Canberra\",4.3,14.5,0,2,3,\"E\",30,\"SSE\",\"ENE\",6,13,76,70,
## [38] "2008-08-03,\"Canberra\",2.3,12.8,0,2.2,9.6,\"WNW\",35,\"ENE\",\"WNW\",4,24,
## [39] "2008-08-28,\"Canberra\",-0.1,14.7,0,3.4,9.9,\"ENE\",30,\"SE\",\"NE\",6,15,
## [40] "2008-08-29,\"Canberra\",-0.2,16.2,0,3.4,5.9,\"E\",28,NA,\"ENE\",0,20,74,45,
## [41] "2008-09-04,\"Canberra\",5.4,11.3,0.2,2.2,0.6,\"ENE\",35,\"SE\",\"ESE\",11,
## [42] "2008-09-17,\"Canberra\",0.7,14.1,0,5.6,9,\"ENE\",20,\"SSW\",\"NNW\",6,7,69,
## [43] "2008-10-05,\"Canberra\",10.3,21.3,3,4.2,6.7,\"NNW\",43,\"ENE\",\"N\",7,19,
## [44] "2008-10-12,\"Canberra\",4.5,23.9,0,4.8,11.7,\"NW\",30,\"ENE\",\"NNW\",4,11,
## [45] "2008-10-15,\"Canberra\",9.2,19.6,0.6,3.4,10.4,\"ENE\",31,\"SSE\",\"NNW\",4,
## [46] "2008-10-21,\"Canberra\",9,20.6,0,9,6.2,\"ENE\",39,\"S\",\"SW\",11,11,54,28,
## [47] "2008-10-29,\"Canberra\",12.5,19.9,0,8.4,5.3,\"ESE\",43,\"ENE\",\"ENE\",11,
```

References

Williams, Graham. 2020. *Rattle: Graphical User Interface for Data Science in r*. https://rattle.togaware.com/.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*

Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

7.9 TSV Data

20200928 A common interchange format for sharing data across different applications is the Tab Separated Value (**tsv**) file format. Reading **tsv** files is straight forward using readr::read_tsv().

```
library(readr)        # Modern and efficient data reader/writer.
library(stringi)      # String concat operator: %s+%.

dsname <- "abc"
dspath <- dsname %s+% ".tsv"
abc    <- read_csv(dspath)
ds     <- get(dsname)
```

Writing a dataset to a **csv** file is straightforward using readr::write_csv():

```
library(rattle)       # Dataset: weatherAUS.
library(dplyr)        # Wrangling: select().
library(readr)        # Modern and efficient data reader/writer.

dsname <- "weatherAUS"
ds     <- get(dsname)

fname  <- "australian_temperatures.tsv"

ds %>%
  select(Date, Location, MinTemp, MaxTemp, Temp9am, Temp3pm) %>%
  write_tsv(fname)
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of*

Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

8 Data Template

REVIEW The *business understanding* phase of a data science project aims to understand the business problem and to then liaise with the business data technicians to identify the data available. This is followed by the *data understanding* phase where we work with the business data technicians to access and ingest the data into R. We are then in a position to initiate our journey of discovery driven by the data. By *living and breathing* the data in the context of the business problem we gain our bearings and feed our intuitions as we journey.

In this chapter we present the common series of steps that initialise the data phase of data science—the data setup. Through this chapter we extract the basic shape and characteristics of the dataset. We prepare the dataset for exploration and wrangling. At the end of this chapter we will have a template for the repeatable end-to-end processing of the data. As you become proficient with R and data science you will develop your own habits and idiosyncrasies which you will incorporate into your own template.

The template concept, developed extensively by Graham J. Williams (2017), consists of canonical programming codes that can be reused with little or no modification on a new dataset. The intention is that to get started with a new dataset only a few initial lines code within the template need to be modified. Only minimal change is then required for the remainder of the codes within the template. For the software engineer, the concept of a template is a stepping stone toward developing functions in R that are general and reusable. For us though, rather than delving into the intricacies of the R language we immerse ourselves into using R to achieve our outcomes, learning R as we proceed and moving into more sophisticated software engineering practices.

Rather than delving into the intricacies of the R language we immerse ourselves into using R to achieve our outcomes, learning more about R as we proceed.

The template consists of programming code that can be reused with little or no modification on a new dataset. The intention is that to get started with a new dataset only a few lines at the top of the template need to be modified. Minimal (if any) change is then required for the remainder of the code. In many respects the concept of a template is a stepping stone toward writing functions in R.

References

Williams, Graham J. 2017. *The Essentials of Data Science: Knowledge Discovery Using r*. The r Series. CRC Press.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

8.1 Dataset Setup

20200320 Packages used in this chapter include dplyr (Wickham et al. 2021), janitor (Firke 2021), magrittr (Bache and Wickham 2020), randomForest (Breiman et al. 2018), and rattle (G. Williams 2020).

Packages are loaded into the currently running R session from your local library directories on disk. Missing packages can be installed using utils::install.packages() within R. On Ubuntu, for example, R packages can also be installed using `$ wajig install r-cran-<pkgname>`.

```
# Load required packages from local library into the R session.

library(dplyr)        # Wrangling: select() sample_frac().
library(janitor)      # Cleanup: clean_names().
library(magrittr)     # Data pipelines: %>% %<>% %T>% equals().
library(randomForest) # Model: randomForest() na.roughfix() for missing data.
library(rattle)       # Dataset: weather.
```

After loading the required packages into the library we access the rattle::weatherAUS dataset and save it into the template dataset named `ds`, as per the template based approach introduced in Graham J. Williams (2017). The dataset is reasonably large (rows or observations by columns or variables) and is used extensively in this book to illustrate the capabilities of R for the Data Scientist.

```
# Initialise the dataset as per the template.
dsname <- "weatherAUS"
ds     <- get(dsname)

ds %>% sample_frac()
```

```
## # A tibble: 3,984 x 24
##    Date                Location MinTemp MaxTemp Rainfall Evaporation Sunshine
##    <dttm>              <chr>      <dbl>   <dbl>    <dbl>       <dbl>    <dbl>
##  1 2013-04-12 00:00:00 Sydney      15.6    26        0           4.4     10.3
##  2 2009-03-03 00:00:00 Sydney      20.8    26.9      0           6.8      6.9
##  3 2020-01-21 00:00:00 Sydney      19.8    32        2.6         6.6     12.9
##  4 2011-06-19 00:00:00 Sydney       8.3    18.4      0           2.8      9.7
##  5 2015-01-22 00:00:00 Sydney      22.5    28.3      0.2         8.4     11.7
##  6 2014-07-07 00:00:00 Sydney       8.3    19.5      0           3        9.2
##  7 2009-08-21 00:00:00 Sydney      11.1    25        0           4        4.8
##  8 2019-08-15 00:00:00 Sydney       6.4    21.1      0           4.8     10.4
##  9 2017-08-22 00:00:00 Sydney       8.7    20.2      0           1.4     10.6
## 10 2014-04-20 00:00:00 Sydney      13.2    22.9      0           5.2     10.4
## # … with 3,974 more rows, and 17 more variables: WindGustDir <chr>,
## #   WindGustSpeed <dbl>, WindDir9am <chr>, WindDir3pm <chr>,
## #   WindSpeed9am <dbl>, WindSpeed3pm <dbl>, Humidity9am <dbl>,
## #   Humidity3pm <dbl>, Pressure9am <dbl>, Pressure3pm <dbl>, Cloud9am <dbl>,
## #   Cloud3pm <dbl>, Temp9am <dbl>, Temp3pm <dbl>, RainToday <chr>,
## #   RISK_MM <dbl>, RainTomorrow <chr>
```

References

Bache, Stefan Milton, and Hadley Wickham. 2020. *Magrittr: A Forward-Pipe Operator for r*. https://CRAN.R-project.org/package=magrittr.

Breiman, Leo, Adele Cutler, Andy Liaw, and Matthew Wiener. 2018. *randomForest: Breiman and Cutler's Random Forests for Classification and Regression*. https://www.stat.berkeley.edu/~breiman/RandomForests/.

Firke, Sam. 2021. *Janitor: Simple Tools for Examining and Cleaning Dirty Data*. https://github.com/sfirke/janitor.

Wickham, Hadley, Romain François, Lionel Henry, and Kirill Müller. 2021. *Dplyr: A Grammar of Data Manipulation*. https://CRAN.R-project.org/package=dplyr.

Williams, Graham. 2020. *Rattle: Graphical User Interface for Data Science in r*. https://rattle.togaware.com/.

Williams, Graham J. 2017. *The Essentials of Data Science: Knowledge Discovery Using r*. The r Series. CRC Press.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

8.2 Data Glimpse

20200317 Another useful tool to quickly review the dataset is pillar::glimpse(). Data from all columns is presented, including the first observations of the individual columns.

```
glimpse(ds)
```

```
## Rows: 3,984
## Columns: 24
## $ Date          <dttm> 2008-02-01, 2008-02-02, 2008-02-03, 2008-02-04, 2008-02…
## $ Location      <chr> "Sydney", "Sydney", "Sydney", "Sydney", "Sydney", "Sydne…
## $ MinTemp       <dbl> 19.5, 19.5, 21.6, 20.2, 19.7, 20.2, 18.6, 17.2, 16.4, 14…
## $ MaxTemp       <dbl> 22.4, 25.6, 24.5, 22.8, 25.7, 27.2, 26.3, 22.3, 20.8, 24…
## $ Rainfall      <dbl> 15.6, 6.0, 6.6, 18.8, 77.4, 1.6, 6.2, 27.6, 12.6, 8.8, 0…
## $ Evaporation   <dbl> 6.2, 3.4, 2.4, 2.2, NA, 2.6, 5.2, 5.8, 4.8, 4.4, 6.4, 6.…
## $ Sunshine      <dbl> 0.0, 2.7, 0.1, 0.0, 0.0, 8.6, 5.2, 2.1, 3.0, 10.1, 8.0, …
## $ WindGustDir   <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …
## $ WindGustSpeed <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …
## $ WindDir9am    <chr> "S", "W", "ESE", "NNE", "NNE", "W", "W", "S", "SSW", "W"…
## $ WindDir3pm    <chr> "SSW", "E", "ESE", "E", "W", "ENE", "S", "SE", "W", "SSE…
## $ WindSpeed9am  <dbl> 17, 9, 17, 22, 11, 9, 15, 7, 19, 11, 9, 7, 24, 15, 19, 1…
## $ WindSpeed3pm  <dbl> 20, 13, 2, 20, 6, 22, 15, 15, 9, 20, 26, 24, 30, 19, 22,…
## $ Humidity9am   <dbl> 92, 83, 88, 83, 88, 69, 75, 77, 92, 80, 78, 68, 87, 81, …
## $ Humidity3pm   <dbl> 84, 73, 86, 90, 74, 62, 80, 61, 91, 53, 53, 67, 70, 51, …
## $ Pressure9am   <dbl> 1017.6, 1017.9, 1016.7, 1014.2, 1008.3, 1002.7, 999.0, 1…
## $ Pressure3pm   <dbl> 1017.4, 1016.4, 1015.6, 1011.8, 1004.8, 998.6, 1000.3, 1…
## $ Cloud9am      <dbl> 8, 7, 7, 8, 8, 6, 4, 7, 7, 4, 7, 7, 8, 7, 7, 7, 7, 7, 7,…
## $ Cloud3pm      <dbl> 8, 7, 8, 8, 8, 6, 7, 8, 7, 2, 8, 7, 7, 1, 3, 6, 7, 6, 7,…
## $ Temp9am       <dbl> 20.7, 22.4, 23.5, 21.4, 22.5, 23.8, 21.7, 18.9, 17.1, 17…
## $ Temp3pm       <dbl> 20.9, 24.8, 23.0, 20.9, 25.5, 26.0, 22.3, 21.1, 16.5, 23…
## $ RainToday     <chr> "Yes", "Yes", "Yes", "Yes", "Yes", "Yes", "Yes", "Yes", …
## $ RISK_MM       <dbl> 6.0, 6.6, 18.8, 77.4, 1.6, 6.2, 27.6, 12.6, 8.8, 0.0, 0.…
## $ RainTomorrow  <chr> "Yes", "Yes", "Yes", "Yes", "Yes", "Yes", "Yes", "Yes", …
```

Notice the style used for variable names here. Different datasets will use different styles. It is useful to convert the variable names (and the levels of a factor) to a canonical form across all of the dataset that we deal with and so avoid having to remember particular naming schemes. We do this next.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

8.3 Normalise Variable Names

Variable names are normalised so as to have some certainty in interacting with the data. The convenience function janitor::clean_names() can do this.

```
# Review the variables before normalising their names.

names(ds)
```

```
##  [1] "Date"          "Location"      "MinTemp"       "MaxTemp"
##  [5] "Rainfall"      "Evaporation"   "Sunshine"      "WindGustDir"
##  [9] "WindGustSpeed" "WindDir9am"    "WindDir3pm"    "WindSpeed9am"
## [13] "WindSpeed3pm"  "Humidity9am"   "Humidity3pm"   "Pressure9am"
## [17] "Pressure3pm"   "Cloud9am"      "Cloud3pm"      "Temp9am"
## [21] "Temp3pm"       "RainToday"     "RISK_MM"       "RainTomorrow"
```

```
# Capture the original variable names for use in plots.

vnames <- names(ds)

# Normalise the variable names.

ds %<>% clean_names(numerals="right")

# Confirm the results are as expected.

names(ds)
```

```
##  [1] "date"            "location"        "min_temp"        "max_temp"
##  [5] "rainfall"        "evaporation"     "sunshine"        "wind_gust_dir"
##  [9] "wind_gust_speed" "wind_dir_9am"    "wind_dir_3pm"    "wind_speed_9am"
## [13] "wind_speed_3pm"  "humidity_9am"    "humidity_3pm"    "pressure_9am"
## [17] "pressure_3pm"    "cloud_9am"       "cloud_3pm"       "temp_9am"
## [21] "temp_3pm"        "rain_today"      "risk_mm"         "rain_tomorrow"
```

```
# Index the original variable names by the new names.

names(vnames) <- names(ds)

vnames
```

```
##            date        location        min_temp        max_temp        rainfall
##          "Date"      "Location"       "MinTemp"       "MaxTemp"      "Rainfall"
##     evaporation        sunshine   wind_gust_dir wind_gust_speed    wind_dir_9am
##   "Evaporation"      "Sunshine"   "WindGustDir" "WindGustSpeed"    "WindDir9am"
##    wind_dir_3pm  wind_speed_9am  wind_speed_3pm    humidity_9am    humidity_3pm
##    "WindDir3pm"  "WindSpeed9am"  "WindSpeed3pm"   "Humidity9am"   "Humidity3pm"
##    pressure_9am    pressure_3pm       cloud_9am       cloud_3pm        temp_9am
##   "Pressure9am"   "Pressure3pm"      "Cloud9am"      "Cloud3pm"       "Temp9am"

##        temp_3pm      rain_today         risk_mm   rain_tomorrow
##       "Temp3pm"     "RainToday"       "RISK_MM"  "RainTomorrow"
```

Notice that we capture the original variable names in the variable for reference, and particularly when generating plots and wanting to use the original names.

The variable names now conform to our expectations of them and in accordance to our chosen style as documented in Chapter 27.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

8.4 Variables and Model Target

Reviewing the variables we will note the different roles each of the variables play. Again we make use of the template in identifying variable roles. Notice the choice to place the target variable at the end of the variable list which is common practice. Loading the dataset into rattle will then automatically identify this variable as the target variable. Reverse the list to generate a stats::formula().

```
# Note the available variables.

vars <- names(ds) %T>% print()
```

```
##  [1] "date"            "location"        "min_temp"        "max_temp"
##  [5] "rainfall"        "evaporation"     "sunshine"        "wind_gust_dir"
##  [9] "wind_gust_speed" "wind_dir_9am"    "wind_dir_3pm"    "wind_speed_9am"
## [13] "wind_speed_3pm"  "humidity_9am"    "humidity_3pm"    "pressure_9am"
## [17] "pressure_3pm"    "cloud_9am"       "cloud_3pm"       "temp_9am"
## [21] "temp_3pm"        "rain_today"      "risk_mm"         "rain_tomorrow"
```

```
# Note the target variable.

target <- "rain_tomorrow"

# Place the target variable at the beginning of the vars.

vars <- c(target, vars) %>% unique() %>% rev() %T>% print()
```

```
##  [1] "risk_mm"         "rain_today"      "temp_3pm"        "temp_9am"
##  [5] "cloud_3pm"       "cloud_9am"       "pressure_3pm"    "pressure_9am"
##  [9] "humidity_3pm"    "humidity_9am"    "wind_speed_3pm"  "wind_speed_9am"
## [13] "wind_dir_3pm"    "wind_dir_9am"    "wind_gust_speed" "wind_gust_dir"
## [17] "sunshine"        "evaporation"     "rainfall"        "max_temp"
## [21] "min_temp"        "location"        "date"            "rain_tomorrow"
```

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

8.5 Identify Numeric Variables

We identify the numeric variables of a dataset by purrr::map()ing the function base::is.numeric() to find base::which() are numeric. Their names are stored into the variable `numc`.

```
# Note which variables are numeric.

vars %>%
  extract(ds, .) %>%
  sapply(is.numeric) %>%
  which() %>%
  names() %T>%
  print() ->
numc
```

```
##  [1] "risk_mm"         "temp_3pm"        "temp_9am"        "cloud_3pm"
##  [5] "cloud_9am"       "pressure_3pm"    "pressure_9am"    "humidity_3pm"
##  [9] "humidity_9am"    "wind_speed_3pm"  "wind_speed_9am"  "wind_gust_speed"
## [13] "sunshine"        "evaporation"     "rainfall"        "max_temp"
## [17] "min_temp"
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

8.6 Normalise Factor Levels

20200607 The levels of factors are also normalised.

```
for (v in which(sapply(ds, is.factor))) levels(ds[[v]]) %<>% normVarNames()
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

8.7 Modelling Roles

```r
# Note the risk variable which measures the severity of the outcome.

risk <- "risk_mm"

# Note the identifiers.

id <- c("date", "location")

# Initialise ignored variables: identifiers.

ignore <- c(risk, id)

# Remove the variables to ignore.

vars <- setdiff(vars, ignore)

# Identify the input variables for modelling.

inputs <- setdiff(vars, target) %T>% print()
```

```
##  [1] "rain_today"      "temp_3pm"        "temp_9am"        "cloud_3pm"
##  [5] "cloud_9am"       "pressure_3pm"    "pressure_9am"    "humidity_3pm"
##  [9] "humidity_9am"    "wind_speed_3pm"  "wind_speed_9am"  "wind_dir_3pm"
## [13] "wind_dir_9am"    "wind_gust_speed" "wind_gust_dir"   "sunshine"
## [17] "evaporation"     "rainfall"        "max_temp"        "min_temp"
```

```
# Also record them by indicies.

inputi <-
  inputs %>%
  sapply(function(x) which(x == names(ds)), USE.NAMES=FALSE) %T>%
  print()
```

```
##  [1] 22 21 20 19 18 17 16 15 14 13 12 11 10  9  8  7  6  5  4  3
```

Also create the formula for modelling. Note that the target variable is the final column of the dataset. The stats::formula() function treats the first column as the target so reverse the list here to automatically generate the correct default formula.

```
form <- formula(ds[rev(vars)])
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

8.8 Variable Types

```
# Identify the numeric variables by index.

ds %>%
  sapply(is.numeric) %>%
  which() %>%
  intersect(inputi) %T>%
  print() ->
numi
```

```
##  [1]  3  4  5  6  7  9 12 13 14 15 16 17 18 19 20 21
```

```
# Identify the numeric variables by name.

ds %>%
  names() %>%
  magrittr::extract(numi) %T>%
  print() ->
numc
```

```
##  [1] "min_temp"        "max_temp"        "rainfall"        "evaporation"
##  [5] "sunshine"        "wind_gust_speed" "wind_speed_9am"  "wind_speed_3pm"
##  [9] "humidity_9am"    "humidity_3pm"    "pressure_9am"    "pressure_3pm"
## [13] "cloud_9am"       "cloud_3pm"       "temp_9am"        "temp_3pm"
```

```r
# Identify the categoric variables by index.

ds %>%
  sapply(is.factor) %>%
  which() %>%
  intersect(inputi) %T>%
  print() ->
cati
```

```
## integer(0)
```

```r
# Identify the categoric variables by name.

ds %>%
  names() %>%
  magrittr::extract(cati) %T>%
  print() ->
catc
```

```
## character(0)
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

8.9 Formula to Describe the Goal

20200607 In the context of supporting analytic modelling tasks we identify *formula* used to describe the model to be built. Typically we will model the target variable on the input variables, so that using any resulting model with a new set of values for the input variables we can predict the value of the target variable.

Using stats::formula() we can automatically construct the formula from the dataset itself if the first column of the dataset is the target variable and the remaining columns are the input variables. Our usual ordering of columns within a dataset place the target variable as the last variable rather than the first. A simple selection of the columns from `vars` in the reverse order, using base::rev(), will then lead to the right formula automatically.

```
form <- formula(ds[rev(vars)]) %T>% print()
```

```
## rain_tomorrow ~ min_temp + max_temp + rainfall + evaporation +
##     sunshine + wind_gust_dir + wind_gust_speed + wind_dir_9am +
##     wind_dir_3pm + wind_speed_9am + wind_speed_3pm + humidity_9am +
##     humidity_3pm + pressure_9am + pressure_3pm + cloud_9am +
##     cloud_3pm + temp_9am + temp_3pm + rain_today
```

The notation used to express the formula begins with the name of the target (`\Sexpr{latexTranslate(target)` }) followed by a tilde (` `) followed by the variables that will be used to model the target, each separated by a plus (`+`). The formula indicates that we will fit a model to predict `\Sexpr{latexTranslate(target)` } from the remaining input variables.

A shorthand for this same formulation is:

```
rain_tomorrow ~ .
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

8.10 Missing Values

The topic of missing values is a significant topic and is covered elsewhere. For purposes of demonstration missing values are removed from our dataset by imputing (i.e., making up) values for any missing value using randomForest::na.roughfix().

```
ds[vars] <- na.roughfix(ds[vars])
```

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

8.11 Random Seed

REVIEW In much of our modelling we will be randomly sampling datasets. In sampling datasets a random number sequence will be used. Such a sequence can be repeatable by initialising with a randomly'' selected seed. We do this so that we can replicate the examples presented throughout this book. We will shortly identify a random training dataset as a subset of the whole dataset. To ensure the same random subset is selected each time we initiate the random number generator with a specific seed using base::set.seed(). For no particular reason we choose a seed.

```
seed <- 42
set.seed(seed)
```

It is worth noting that many model builders use heuristics to search for a good model. The general approach is to search for a good model rather than the best model. This is often necessary because the computational requirements to find the best model will generally be prohibitive and can be as much as years of computer time. Searching for the best model involves searching through an enormous search space of all of the possible models. Our algorithms will reduce the computational requirements to something feasible using heuristics. Such heuristics often involve some level of random decision making in deciding which paths to follow in any search. By setting the seed for the random number generator to a known initial value will ensure we can replicate the model building later.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

8.12 Train, Tune, and Test Datasets

20200607 The final data wrangling is to partition the dataset into three separate datasets: training, tuning, and test datasets. The training dataset will be used to fit a model to the data. The tuning dataset is used to the tune the parameters for the model building process. Whilst these observations are not directly modelled they do guide the model fitting process. The test dataset is then only used to assess the performance of the finally fit and tuned model. This will provide an unbiased estimate of the performance of the final model on new observations.

To build the datasets we will use a random selection process. We are now ready to partition the dataset into two subsets. The first is a 70% random sample for building the model (the `train`ing dataset) and the second is the remainder, used to evaluate the performance of the model (the `test` dataset).

```
PTR <- 0.7   # Proportion for training
PTU <- 0.15  # Proportion for tuning
PTE <- 0.15  # Proportion for testing

tr <- sample(nobs, PTR*nobs) %T>%
  {length(.) %>% print()}
```

```
## [1] 70
```

```
tu <- nobs %>% seq_len() %>% setdiff(tr) %>% sample(PTU*nobs) %T>%
  {length(.) %>% print()}
```

```
## [1] 15
```

```
te <- nobs %>% seq_len() %>% setdiff(tr) %>% setdiff(tu) %T>%
  {length(.) %>% print()}
```

```
## [1] 15
```

```
head(tr)
```

```
## [1]  49  65  25  74  18 100
```

Any model building we do will be based on the 70% `train`ing dataset. Our model may then be quite good at predicting these observations. The model's performance on this data on which it was trained will be a very optimistic (or biased) estimate of the true performance of the model on other datasets. We might thus ask how will the model perform when we use it to predict outcomes for other, yet unseen, observations?

The testing dataset is a hold-out dataset in that it has not been used at all for building the model. When we apply the model to this dataset we would expect it to have a lesser performance (e.g., a higher error rate). This is what we will generally observe and we will see this in the following sections.

The overall error rate measured on the `train`ing dataset will be shown to be less than the error rate calculated on the `test` dataset. The error rate (or any performance measure in general) calculated on the `test` dataset is closer to what we will obtain in general when we begin to use the model. It is an unbiased estimate of the true performance of the model.

We also record the actual target values and the risks. These will be used in the evaluation of the performance of the models.

```
target.tr <- ds %>% slice(tr) %>% pull(target)
target.tu <- ds %>% slice(tu) %>% pull(target)
target.te <- ds %>% slice(te) %>% pull(target)

risk.tr   <- ds %>% slice(tr) %>% pull(risk)
risk.tu   <- ds %>% slice(tu) %>% pull(risk)
risk.te   <- ds %>% slice(te) %>% pull(risk)
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

8.13 Review the Dataset

The dataset is now ready to be analysed but before doing so it is useful to have another pillar::glimpse().

```
glimpse(ds)
```

```
## Rows: 3,984
## Columns: 24
## $ date            <dttm> 2008-02-01, 2008-02-02, 2008-02-03, 2008-02-04, 2008-…
## $ location        <chr> "Sydney", "Sydney", "Sydney", "Sydney", "Sydney", "Syd…
## $ min_temp        <dbl> 19.5, 19.5, 21.6, 20.2, 19.7, 20.2, 18.6, 17.2, 16.4, …
## $ max_temp        <dbl> 22.4, 25.6, 24.5, 22.8, 25.7, 27.2, 26.3, 22.3, 20.8, …
## $ rainfall        <dbl> 15.6, 6.0, 6.6, 18.8, 77.4, 1.6, 6.2, 27.6, 12.6, 8.8,…
## $ evaporation     <dbl> 6.2, 3.4, 2.4, 2.2, NA, 2.6, 5.2, 5.8, 4.8, 4.4, 6.4, …
## $ sunshine        <dbl> 0.0, 2.7, 0.1, 0.0, 0.0, 8.6, 5.2, 2.1, 3.0, 10.1, 8.0…
## $ wind_gust_dir   <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA…
## $ wind_gust_speed <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA…
## $ wind_dir_9am    <chr> "S", "W", "ESE", "NNE", "NNE", "W", "W", "S", "SSW", "…
## $ wind_dir_3pm    <chr> "SSW", "E", "ESE", "E", "W", "ENE", "S", "SE", "W", "S…
## $ wind_speed_9am  <dbl> 17, 9, 17, 22, 11, 9, 15, 7, 19, 11, 9, 7, 24, 15, 19,…
## $ wind_speed_3pm  <dbl> 20, 13, 2, 20, 6, 22, 15, 15, 9, 20, 26, 24, 30, 19, 2…
## $ humidity_9am    <dbl> 92, 83, 88, 83, 88, 69, 75, 77, 92, 80, 78, 68, 87, 81…
## $ humidity_3pm    <dbl> 84, 73, 86, 90, 74, 62, 80, 61, 91, 53, 53, 67, 70, 51…
## $ pressure_9am    <dbl> 1017.6, 1017.9, 1016.7, 1014.2, 1008.3, 1002.7, 999.0,…
## $ pressure_3pm    <dbl> 1017.4, 1016.4, 1015.6, 1011.8, 1004.8, 998.6, 1000.3,…
## $ cloud_9am       <dbl> 8, 7, 7, 8, 8, 6, 4, 7, 7, 4, 7, 7, 8, 7, 7, 7, 7, 7, …
## $ cloud_3pm       <dbl> 8, 7, 8, 8, 8, 6, 7, 8, 7, 2, 8, 7, 7, 1, 3, 6, 7, 6, …
## $ temp_9am        <dbl> 20.7, 22.4, 23.5, 21.4, 22.5, 23.8, 21.7, 18.9, 17.1, …
## $ temp_3pm        <dbl> 20.9, 24.8, 23.0, 20.9, 25.5, 26.0, 22.3, 21.1, 16.5, …
## $ rain_today      <chr> "Yes", "Yes", "Yes", "Yes", "Yes", "Yes", "Yes", "Yes"…
## $ risk_mm         <dbl> 6.0, 6.6, 18.8, 77.4, 1.6, 6.2, 27.6, 12.6, 8.8, 0.0, …
## $ rain_tomorrow   <chr> "Yes", "Yes", "Yes", "Yes", "Yes", "Yes", "Yes", "Yes"…
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

8.14 Data Template

```r
library(janitor)      # Cleanup: clean_names().
library(rattle)       # Dataset: weatherAUS.
library(magrittr)     # Pipelines: %<>%.
library(randomForest) # Imputation: na.roughfix().
library(dplyr)        # Wrangling: rename_all().


dsname <- "weatherAUS"
ds     <- get(dsname)
nobs   <- nrow(ds)

dim(ds)

vnames        <- names(ds)
ds           %<>% clean_names(numerals="right")
names(vnames) <- names(ds)

names(ds)

vars   <- names(ds)
target <- "rain_tomorrow"
vars   <- c(target, vars) %>% unique() %>% rev()

for (v in which(sapply(ds, is.factor))) levels(ds[[v]]) %<>% normVarNames()

risk   <- "risk_mm"
id     <- c("date", "location")
ignore <- c(risk, id)
vars   <- setdiff(vars, ignore)
inputs <- setdiff(vars, target)

form   <- formula(paste(target, "~ ."))

# ds[vars] %<>% na.roughfix()
```

```r
SPLIT <- c(0.7, 0.15, 0.15)

tr     <- sample(nobs, SPLIT[1]*nobs)
tu     <- nobs %>% seq_len() %>% setdiff(tr) %>% sample(SPLIT[2]*nobs)
te     <- nobs %>% seq_len() %>% setdiff(tr) %>% setdiff(tu)

target.tr <- ds %>% slice(tr) %>% pull(target)
target.tu <- ds %>% slice(tu) %>% pull(target)
target.te <- ds %>% slice(te) %>% pull(target)

if (!is.null(risk))
{
  risk.tr   <- ds %>% slice(tr) %>% pull(risk)
  risk.tu   <- ds %>% slice(tu) %>% pull(risk)
  risk.te   <- ds %>% slice(te) %>% pull(risk)
}
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

9 Data Exploration

20210309 This chapter is about exploring data. This includes simply extracting different bits of the data, and also statistical summaries of the data, such as the averages, skewness, etc. Exploration is contrasted with Wrangling where in Wrangling you are transforming the data and cleaning the data in various ways.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

9.1 Exploration Setup

20200317 Packages used in this chapter include dplyr (Wickham et al. 2021), magrittr (Bache and Wickham 2020), and rattle (G. Williams 2020).

Packages are loaded into the currently running R session from your local library directories on disk. Missing packages can be installed using utils::install.packages() within R. On Ubuntu, for example, R packages can also be installed using `$ wajig install r-cran-<pkgname>`.

```
# Load required packages from local library into the R session.

library(dplyr)        # Wrangling: select() sample_frac().
library(magrittr)     # Data pipelines: %>% %<>% %T>% equals().
library(rattle)       # normVarNames(). Dataset: weather.
```

The rattle::weatherAUS dataset is loaded into the template variable `ds` and further template variables are setup as introduced by Graham J. Williams (2017). See Chapter 8 for details.

```
dsname <- "weatherAUS"
ds     <- get(dsname)

nobs   <- nrow(ds)

vnames <- names(ds)
ds    %<>% clean_names(numerals="right")
names(vnames) <- names(ds)

vars   <- names(ds)
target <- "rain_tomorrow"
vars   <- c(target, vars) %>% unique() %>% rev()
```

A random sample of the dataset:

```r
ds %>% sample_frac()
```

```
## # A tibble: 176,747 x 24
##    date       location     min_temp max_temp rainfall evaporation sunshine
##    <date>     <chr>           <dbl>    <dbl>    <dbl>       <dbl>    <dbl>
##  1 2013-09-03 Penrith           7.8     24.6      0          NA       NA
##  2 2017-12-12 Cobar            20.7     37.2      0          NA       NA
##  3 2014-12-01 WaggaWagga       17.7     33.9      3.8         5.4     10.1
##  4 2012-10-09 SalmonGums        9.4     21.8      0.4        NA       NA
##  5 2012-06-10 WaggaWagga        0.7     16        0           1        9
##  6 2019-01-20 GoldCoast        21.8     29.7      0          NA       NA
##  7 2019-03-11 Uluru            23.7     44.8      0          NA       NA
##  8 2014-05-29 Penrith           7       22.1      0          NA       NA
##  9 2012-10-21 Woomera          15.5     28.7      0          15.8     11
## 10 2016-01-08 MountGambier     10.9     23.8      0           6.8     12
## # … with 176,737 more rows, and 17 more variables: wind_gust_dir <ord>,
## #   wind_gust_speed <dbl>, wind_dir_9am <ord>, wind_dir_3pm <ord>,
## #   wind_speed_9am <dbl>, wind_speed_3pm <dbl>, humidity_9am <int>,
## #   humidity_3pm <int>, pressure_9am <dbl>, pressure_3pm <dbl>,
## #   cloud_9am <int>, cloud_3pm <int>, temp_9am <dbl>, temp_3pm <dbl>,
## #   rain_today <fct>, risk_mm <dbl>, rain_tomorrow <fct>
```

References

Bache, Stefan Milton, and Hadley Wickham. 2020. *Magrittr: A Forward-Pipe Operator for r*. https://CRAN.R-project.org/package=magrittr.

Wickham, Hadley, Romain François, Lionel Henry, and Kirill Müller. 2021. *Dplyr: A Grammar of Data Manipulation*. https://CRAN.R-project.org/package=dplyr.

Williams, Graham. 2020. *Rattle: Graphical User Interface for Data Science in r*. https://rattle.togaware.com/.

Williams, Graham J. 2017. *The Essentials of Data Science: Knowledge Discovery Using r*. The r Series. CRC Press.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

9.2 Counting Groups

Groups can be counted using dplyr::count().

```
ds %>% count(wind_dir_3pm)
```

```
## # A tibble: 17 x 2
##    wind_dir_3pm     n
##    <ord>        <int>
##  1 N            10475
##  2 NNE           8002
##  3 NE           10092
##  4 ENE           9605
##  5 E            10123
##  6 ESE          10290
##  7 SE           12919
##  8 SSE          11089
##  9 S            11788
## 10 SSW           9902
## 11 SW           11166
## 12 WSW          11700
## 13 W            12411
## 14 WNW          10846
## 15 NW           10315
## 16 NNW           9358
## 17 <NA>          6666
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open*

source software since 1984.

Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

9.3 Random Sample

20200317

A common task is to randomly sample rows from a dataset. The dplyr::sample_frac() function will randomly choose a specified fraction (e.g. 20%) of the rows of the dataset:

```
ds %>% sample_frac(0.2)
```

```
## # A tibble: 35,349 x 24
##    date       location     min_temp max_temp rainfall evaporation sunshine
##    <date>     <chr>           <dbl>    <dbl>    <dbl>       <dbl>    <dbl>
##  1 2010-09-08 Cairns           23.5     30.2      0           7        7.8
##  2 2011-10-25 Mildura           8.7     21.5     13.4         4.4     12.2
##  3 2017-06-24 Launceston       -0.5     11.4      8.4        NA       NA
##  4 2014-09-26 Katherine        20.1     37.5      0          11       NA
##  5 2015-02-05 Nhil              8.1     32.8      0          NA       NA
##  6 2015-02-03 CoffsHarbour     18.1     22.3    104.         NA       NA
##  7 2009-05-20 Cobar            12.5     15.6     17.6         0        0
##  8 2010-01-19 Hobart           11.7     19.4      0.6         5        8.1
##  9 2018-12-31 Sydney           21.7     30.5      0           8        4.1
## 10 2011-12-19 Mildura          15.7     28.4      0           5.8     12.9
## # … with 35,339 more rows, and 17 more variables: wind_gust_dir <ord>,
## #   wind_gust_speed <dbl>, wind_dir_9am <ord>, wind_dir_3pm <ord>,
## #   wind_speed_9am <dbl>, wind_speed_3pm <dbl>, humidity_9am <int>,
## #   humidity_3pm <int>, pressure_9am <dbl>, pressure_3pm <dbl>,
## #   cloud_9am <int>, cloud_3pm <int>, temp_9am <dbl>, temp_3pm <dbl>,
## #   rain_today <fct>, risk_mm <dbl>, rain_tomorrow <fct>
```

The next time you randomly sample the dataset the resulting sample will be different:

```
ds %>% sample_frac(0.2)
```

```
## # A tibble: 35,349 x 24
##    date       location      min_temp max_temp rainfall evaporation sunshine
##    <date>     <chr>            <dbl>    <dbl>    <dbl>       <dbl>    <dbl>
##  1 2011-03-22 BadgerysCreek     19.2     32       13.2        NA       NA
##  2 2011-05-26 Newcastle          9.9     17.2      0          NA       NA
##  3 2013-07-04 AliceSprings       0.3     23.2      0           4       10.5
##  4 2017-06-23 Cairns            19.7     27.3      1          NA       NA
##  5 2019-01-16 Uluru             30.1     45.3      0          NA       NA
##  6 2013-06-12 Uluru              3.1     17.3      0.2        NA       NA
##  7 2019-09-02 Brisbane          11.6     29.3      0           5       10.1
##  8 2012-10-28 Richmond          13.9     21.9      0          NA       NA
##  9 2010-11-26 Melbourne         16       25.1      1.8         0.8      4.6
## 10 2012-03-09 NorfolkIsland     19.1     25.2      0.2         5.2      6.6
## # … with 35,339 more rows, and 17 more variables: wind_gust_dir <ord>,
## #   wind_gust_speed <dbl>, wind_dir_9am <ord>, wind_dir_3pm <ord>,
## #   wind_speed_9am <dbl>, wind_speed_3pm <dbl>, humidity_9am <int>,
## #   humidity_3pm <int>, pressure_9am <dbl>, pressure_3pm <dbl>,
## #   cloud_9am <int>, cloud_3pm <int>, temp_9am <dbl>, temp_3pm <dbl>,
## #   rain_today <fct>, risk_mm <dbl>, rain_tomorrow <fct>
```

To ensure the sample random sample each time use base::set.seed():

```r
set.seed(72346)
ds %>% sample_frac(0.2)
```

```
## # A tibble: 35,349 x 24
##    date       location    min_temp max_temp rainfall evaporation sunshine
##    <date>     <chr>          <dbl>    <dbl>    <dbl>       <dbl>    <dbl>
##  1 2010-07-20 Brisbane        13.4     20.7      1           0.8     NA
##  2 2015-08-24 Walpole          6       15.7      3.2        NA       NA
##  3 2012-08-02 Cobar            1.6     17.2      0           2.4     NA
##  4 2010-03-16 Canberra         9.9     26.6      0           4.6     11.5
##  5 2010-08-03 SalmonGums       0.8     20        0          NA       NA
##  6 2010-06-20 Portland         9.5     14.1      2.6         0.6      4
##  7 2014-01-11 Melbourne       17.8     22.7      0          12.2      6.7
##  8 2018-01-29 Uluru           26.1     40.8      0.2        NA       NA
##  9 2011-01-22 Williamtown     14.8     29.5      0           7.8     12.8
## 10 2012-07-14 Cairns          22.6     28.5      0.8         2        2.5
## # … with 35,339 more rows, and 17 more variables: wind_gust_dir <ord>,
## #   wind_gust_speed <dbl>, wind_dir_9am <ord>, wind_dir_3pm <ord>,
## #   wind_speed_9am <dbl>, wind_speed_3pm <dbl>, humidity_9am <int>,
## #   humidity_3pm <int>, pressure_9am <dbl>, pressure_3pm <dbl>,
## #   cloud_9am <int>, cloud_3pm <int>, temp_9am <dbl>, temp_3pm <dbl>,
## #   rain_today <fct>, risk_mm <dbl>, rain_tomorrow <fct>
```

```r
set.seed(72346)
ds %>% sample_frac(0.2)
```

```
## # A tibble: 35,349 x 24
##    date       location    min_temp max_temp rainfall evaporation sunshine
##    <date>     <chr>          <dbl>    <dbl>    <dbl>       <dbl>    <dbl>
##  1 2010-07-20 Brisbane        13.4     20.7      1           0.8     NA
##  2 2015-08-24 Walpole          6       15.7      3.2        NA       NA
##  3 2012-08-02 Cobar            1.6     17.2      0           2.4     NA
##  4 2010-03-16 Canberra         9.9     26.6      0           4.6     11.5
##  5 2010-08-03 SalmonGums       0.8     20        0          NA       NA
##  6 2010-06-20 Portland         9.5     14.1      2.6         0.6      4
##  7 2014-01-11 Melbourne       17.8     22.7      0          12.2      6.7
##  8 2018-01-29 Uluru           26.1     40.8      0.2        NA       NA
##  9 2011-01-22 Williamtown     14.8     29.5      0           7.8     12.8
## 10 2012-07-14 Cairns          22.6     28.5      0.8         2        2.5
## # … with 35,339 more rows, and 17 more variables: wind_gust_dir <ord>,
## #   wind_gust_speed <dbl>, wind_dir_9am <ord>, wind_dir_3pm <ord>,
## #   wind_speed_9am <dbl>, wind_speed_3pm <dbl>, humidity_9am <int>,
## #   humidity_3pm <int>, pressure_9am <dbl>, pressure_3pm <dbl>,
## #   cloud_9am <int>, cloud_3pm <int>, temp_9am <dbl>, temp_3pm <dbl>,
## #   rain_today <fct>, risk_mm <dbl>, rain_tomorrow <fct>
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

9.4 Constant Variables

20200814 You may wish and typically do ignore variables (dataset columns) with constant values as they add no extra information to the analysis.

```
# Identify variables that have a single value.

ds %>%
  select(all_of(vars)) %>%
  sapply(function(x) all(x == x[1L])) %>%
  which() %>%
  names() %T>%
  print() ->
constants
```

```
## character(0)
```

If you are accumulating a list of variables to ignore then add these constants to that list.

```
# Add to the variables to be ignored for modelling.

ignore %<>% union(constants)
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

9.5 Correlated Numeric Variables

20200814 It is often useful to identify highly correlated variables. Such variables will often record the same information but in different ways and often arise when we combine data from different sources.

The correlation is calculated by dplyr::select()ing the numeric columns from the dataset and passing that through to stats::cor(). This matrix of pairwise correlations is based on only the complete observations so that observations with missing values are ignored.

We set the upper triangle of the correlation matrix to `NA`'s as they are a mirror of the values in the lower triangle and thus redundant. We also set *diag=TRUE* to set the diagonals as `NA` since they will always be perfect correlations.

The processing continues by making all values positive using base::abs(). With conversion to base::data.frame() then to tidyr::as_tibble() the dataset column names need to be reset appropriately using magrittr::set_colnames(). We dplyr::mutate() the dataset with a new column using dplyr::mutate(), reshape the dataset using tidyr::pivot_longer() from tidyr (Wickham 2021) and then omit missing correlations using data.table::na.omit(). Finally the rows are dplyr::arrange()'d with the highest absolute correlations appearing first.

```r
# For the numeric variables generate a table of correlations

ds %>%
  select(all_of(numc)) %>%
  cor(use="complete.obs") %>%
  ifelse(upper.tri(., diag=TRUE), NA, .) %>%
  abs() %>%
  data.frame() %>%
  as_tibble() %>%
  set_colnames(numc) %>%
  mutate(var1=numc) %>%
  tidyr::pivot_longer(var2, cor, -var1) %>%
  na.omit() %>%
  arrange(-abs(cor)) %T>%
  print() ->
mc
```

That could do with some work and instead use corrr:

```
ds %>%
  correlate() %>%
  shave() %>%
  fashion()

ds %>%
  correlate() %>%
  rearrange() %>%
  rplot()

ds %>%
  corrr::correlate() %>%
  shave() %>%
  stretch() %>%
  filter(abs(r) > 0.90)

ds %>%
  correlate() %>%
  focus()

ds %>%
  correlate() %>%
  network_plot()    # Fail
```

References

———. 2021. *Tidyr: Tidy Messy Data*. https://CRAN.R-project.org/package=tidyr.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

9.6 Selecting Columns

20200317

Columns from a data frame can be dplyr::select()'ed simply by listing the column names. The columns in the resulting data frame will be in the same order as they are listed in the function call.

```
ds %>%
  select(date, location, min_temp, temp_9am, temp_3pm, max_temp)
```

```
## # A tibble: 176,747 x 6
##    date       location min_temp temp_9am temp_3pm max_temp
##    <date>     <chr>       <dbl>    <dbl>    <dbl>    <dbl>
##  1 2008-12-01 Albury       13.4     16.9     21.8     22.9
##  2 2008-12-02 Albury        7.4     17.2     24.3     25.1
##  3 2008-12-03 Albury       12.9     21       23.2     25.7
##  4 2008-12-04 Albury        9.2     18.1     26.5     28
##  5 2008-12-05 Albury       17.5     17.8     29.7     32.3
##  6 2008-12-06 Albury       14.6     20.6     28.9     29.7
##  7 2008-12-07 Albury       14.3     18.1     24.6     25
##  8 2008-12-08 Albury        7.7     16.3     25.5     26.7
##  9 2008-12-09 Albury        9.7     18.3     30.2     31.9
## 10 2008-12-10 Albury       13.1     20.1     28.2     30.1
## # … with 176,737 more rows
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

9.7 Selecting Rows

20200419 Rows from a data frame can be stats::filter()'ed using specific conditions. The rows in the resulting data frame will be those for which the condition is true.

```
ds %>%
  filter(max_temp >= mean(max_temp, na.rm=TRUE))
```

```
## # A tibble: 82,459 x 24
##    date       location min_temp max_temp rainfall evaporation sunshine
##    <date>     <chr>       <dbl>    <dbl>    <dbl>       <dbl>    <dbl>
##  1 2008-12-02 Albury        7.4     25.1      0            NA       NA
##  2 2008-12-03 Albury       12.9     25.7      0            NA       NA
##  3 2008-12-04 Albury        9.2     28        0            NA       NA
##  4 2008-12-05 Albury       17.5     32.3      1            NA       NA
##  5 2008-12-06 Albury       14.6     29.7      0.2          NA       NA
##  6 2008-12-07 Albury       14.3     25        0            NA       NA
##  7 2008-12-08 Albury        7.7     26.7      0            NA       NA
##  8 2008-12-09 Albury        9.7     31.9      0            NA       NA
##  9 2008-12-10 Albury       13.1     30.1      1.4          NA       NA
## 10 2008-12-11 Albury       13.4     30.4      0            NA       NA
## # … with 82,449 more rows, and 17 more variables: wind_gust_dir <ord>,
## #   wind_gust_speed <dbl>, wind_dir_9am <ord>, wind_dir_3pm <ord>,
## #   wind_speed_9am <dbl>, wind_speed_3pm <dbl>, humidity_9am <int>,
## #   humidity_3pm <int>, pressure_9am <dbl>, pressure_3pm <dbl>,
## #   cloud_9am <int>, cloud_3pm <int>, temp_9am <dbl>, temp_3pm <dbl>,
## #   rain_today <fct>, risk_mm <dbl>, rain_tomorrow <fct>
```

To select rows that have missing values, for example, use stats::filter() with purrr::pmap_lgl() to map base::is.na() for base::any() column in the row. In the following example we count the number with missing values using base::nrow(), formatted nicely using scales::comma():

```r
ds %>%
  select(-date) %>%
  filter(purrr::pmap_lgl(., ~any(is.na(c(...))))) %>%
  nrow() %>% scales::comma() %>% cat("rows have missing values.\n")
```

```
## 114,944 rows have missing values.
```

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

9.8 Shuffle Rows

Using dplyr::sample_frac() with `size=` 1 (the default) will randomly shuffle the rows of the dataset.

```
ds %>% sample_frac()
```

```
## # A tibble: 176,747 x 24
##    date       location     min_temp max_temp rainfall evaporation sunshine
##    <date>     <chr>           <dbl>    <dbl>    <dbl>       <dbl>    <dbl>
##  1 2019-07-12 Mildura           9.9     17.5      0           7        5.3
##  2 2019-10-16 Dartmoor          8.2     17.2      3.8        NA       NA
##  3 2017-09-15 Woomera           6.5     23.6      0          NA       NA
##  4 2018-12-24 Adelaide         15.6     37        0          NA       NA
##  5 2010-09-08 Brisbane         11.6     23.8      0           4       10.2
##  6 2018-01-09 Witchcliffe      12.7     26.7      0          NA       NA
##  7 2012-11-18 Brisbane         17.5     26.1     64          NA        2.9
##  8 2017-03-18 PerthAirport     16.2     20.5      0.4         5.4      0.1
##  9 2015-07-12 Moree             4.8     13       NA          NA       NA
## 10 2016-01-09 Bendigo          12       34.8     NA          NA       NA
## # … with 176,737 more rows, and 17 more variables: wind_gust_dir <ord>,
## #   wind_gust_speed <dbl>, wind_dir_9am <ord>, wind_dir_3pm <ord>,
## #   wind_speed_9am <dbl>, wind_speed_3pm <dbl>, humidity_9am <int>,
## #   humidity_3pm <int>, pressure_9am <dbl>, pressure_3pm <dbl>,
## #   cloud_9am <int>, cloud_3pm <int>, temp_9am <dbl>, temp_3pm <dbl>,
## #   rain_today <fct>, risk_mm <dbl>, rain_tomorrow <fct>
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open*

source software since 1984.

Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

10 Data Wrangling

20200607 Data will need to be transformed in many different ways to get it ready for visualisation and analysis. This chapter provides a comprehensive catalogue of different tasks useful for wrangling our data. The goal is to transform the acquired raw and untidy data into a tidy dataset.

Chapter 8 presented an approach to data wrangling based on a template that utilises many of the transformations presented here. The template provides the basic steps which in any project will be combined with transformations found in this current chapter. What is to be transformed is usually determined through our data exploration as presented in Chapter 9.

We will also find ourselves coming back to the data wrangling after visualising (Chapter 11) and building machine learning models from the data.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

10.1 Wrangling Setup

20180908 Packages used in this chapter include dplyr (Wickham et al. 2021), FSelector (Romanski, Kotthoff, and Schratz 2021), ggplot2 (Wickham et al. 2020), glue (Hester 2020), janitor (Firke 2021), lobstr (Wickham 2019a), lubridate (Spinu, Grolemund, and Wickham 2021), randomForest (Breiman et al. 2018), readr (Wickham and Hester 2020), stringi (Gagolewski et al. 2021), stringr (Wickham 2019b), tidyr (Wickham 2021), magrittr (Bache and Wickham 2020), and rattle (G. Williams 2020).

Packages are loaded into the currently running R session from your local library directories on disk. Missing packages can be installed using utils::install.packages() within R. On Ubuntu, for example, R packages can also be installed using `$ wajig install r-cran-<pkgname>`.

```
# Load required packages from local library into the R session.

library(rattle)       # weather dataset.
library(readr)        # Efficient reading of CSV data.
library(dplyr)        # Wrangling: glimpse().
library(lobstr)       # Inspect R data structures.
library(tidyr)        # Prepare a tidy dataset, gather().
library(magrittr)     # Pipes %>% and %T>% and equals().
library(glue)         # Format strings.
library(janitor)      # Cleanup: clean_names().
library(lubridate)    # Dates and time.
library(FSelector)    # Feature selection, information.gain().
library(stringi)      # String concat operator %s+%.
library(stringr)      # String operations.
library(randomForest) # Impute missing values with na.roughfix().
library(ggplot2)      # Visualise data.
library(purrr)        # simplify(), set_names()
```

The rattle::weatherAUS dataset is loaded into the template variable `ds` and further template variables are setup as introduced by Graham J. Williams (2017). See Chapter 8 for details.

```r
dsname <- "weatherAUS"
ds     <- get(dsname)

nobs   <- nrow(ds)

vnames <- names(ds)
ds    %<>% clean_names(numerals="right")
names(vnames) <- names(ds)

vars   <- names(ds)
target <- "rain_tomorrow"
vars   <- c(target, vars) %>% unique() %>% rev()
```

References

Bache, Stefan Milton, and Hadley Wickham. 2020. *Magrittr: A Forward-Pipe Operator for r*. https://CRAN.R-project.org/package=magrittr.

Breiman, Leo, Adele Cutler, Andy Liaw, and Matthew Wiener. 2018. *randomForest: Breiman and Cutler's Random Forests for Classification and Regression*. https://www.stat.berkeley.edu/~breiman/RandomForests/.

Firke, Sam. 2021. *Janitor: Simple Tools for Examining and Cleaning Dirty Data*. https://github.com/sfirke/janitor.

Gagolewski, Marek, Bartek Tartanus, and others; IBM, Unicode, Inc., and others. 2021. *Stringi: Character String Processing Facilities*. https://CRAN.R-project.org/package=stringi.

Hester, Jim. 2020. *Glue: Interpreted String Literals*. https://CRAN.R-project.org/package=glue.

Romanski, Piotr, Lars Kotthoff, and Patrick Schratz. 2021. *FSelector: Selecting Attributes*. https://github.com/larskotthoff/fselector.

Spinu, Vitalie, Garrett Grolemund, and Hadley Wickham. 2021. *Lubridate: Make Dealing with Dates a Little Easier*. https://CRAN.R-project.org/package=lubridate.

———. 2019a. *Lobstr: Visualize r Data Structures with Trees*. https://github.com/r-lib/lobstr.

———. 2019b. *Stringr: Simple, Consistent Wrappers for Common String Operations*. https://CRAN.R-project.org/package=stringr.

———. 2021. *Tidyr: Tidy Messy Data*. https://CRAN.R-project.org/package=tidyr.

Wickham, Hadley, Winston Chang, Lionel Henry, Thomas Lin Pedersen, Kohske Takahashi, Claus Wilke, Kara Woo, Hiroaki Yutani, and Dewey Dunnington. 2020. *Ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics*. https://CRAN.R-project.org/package=ggplot2.

Wickham, Hadley, Romain François, Lionel Henry, and Kirill Müller. 2021. *Dplyr: A Grammar of Data Manipulation*. https://CRAN.R-project.org/package=dplyr.

Wickham, Hadley, and Jim Hester. 2020. *Readr: Read Rectangular Text Data*. https://CRAN.R-project.org/package=readr.

Williams, Graham. 2020. *Rattle: Graphical User Interface for Data Science in r*. https://rattle.togaware.com/.

Williams, Graham J. 2017. *The Essentials of Data Science: Knowledge Discovery Using r*. The r Series. CRC Press.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*

Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

10.2 Wrangling Data Review

It is always useful to remind ourselves of the dataset with a random sample:

```
ds  %>% sample_frac() %>% select(date, location, sample(3:length(vars), 5))
```

```
## # A tibble: 176,747 x 7
##    date       location    humidity_9am cloud_9am max_temp temp_9am rain_tomorrow
##    <date>     <chr>              <int>     <int>    <dbl>    <dbl> <fct>
##  1 2013-07-10 Walpole               97        NA     17       11   No
##  2 2010-01-25 SydneyAirp…           80         8     26.9     22.3 No
##  3 2009-12-18 Sale                  64         2     22.4     14.9 No
##  4 2019-08-11 PerthAirpo…           62         0     22.4     13.9 No
##  5 2018-05-11 Albury                91         8      9.9      5.7 Yes
##  6 2013-04-22 Penrith               90        NA     24.4     16   No
##  7 2013-06-16 Dartmoor             100        NA     13.2      5.1 Yes
##  8 2017-08-11 CoffsHarbo…           29        NA     27.7     21.9 No
##  9 2019-10-26 Perth                 56         0     28.9     21.3 No
## 10 2013-10-25 WaggaWagga            60         0     19.3      9.6 No
## # … with 176,737 more rows
```

```
glimpse(ds)
```

```
## Rows: 176,747
## Columns: 24
## $ date            <date> 2008-12-01, 2008-12-02, 2008-12-03, 2008-12-04, 2008-…
## $ location        <chr> "Albury", "Albury", "Albury", "Albury", "Albury", "Alb…
## $ min_temp        <dbl> 13.4, 7.4, 12.9, 9.2, 17.5, 14.6, 14.3, 7.7, 9.7, 13.1…
## $ max_temp        <dbl> 22.9, 25.1, 25.7, 28.0, 32.3, 29.7, 25.0, 26.7, 31.9, …
## $ rainfall        <dbl> 0.6, 0.0, 0.0, 0.0, 1.0, 0.2, 0.0, 0.0, 0.0, 1.4, 0.0,…
## $ evaporation     <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA…
## $ sunshine        <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA…
## $ wind_gust_dir   <ord> W, WNW, WSW, NE, W, WNW, W, W, NNW, W, N, NNE, W, SW, …
## $ wind_gust_speed <dbl> 44, 44, 46, 24, 41, 56, 50, 35, 80, 28, 30, 31, 61, 44…
## $ wind_dir_9am    <ord> W, NNW, W, SE, ENE, W, SW, SSE, SE, S, SSE, NE, NNW, W…
## $ wind_dir_3pm    <ord> WNW, WSW, WSW, E, NW, W, W, W, NW, SSE, ESE, ENE, NNW,…
## $ wind_speed_9am  <dbl> 20, 4, 19, 11, 7, 19, 20, 6, 7, 15, 17, 15, 28, 24, 4,…
## $ wind_speed_3pm  <dbl> 24, 22, 26, 9, 20, 24, 24, 17, 28, 11, 6, 13, 28, 20, …
## $ humidity_9am    <int> 71, 44, 38, 45, 82, 55, 49, 48, 42, 58, 48, 89, 76, 65…
## $ humidity_3pm    <int> 22, 25, 30, 16, 33, 23, 19, 19, 9, 27, 22, 91, 93, 43,…
## $ pressure_9am    <dbl> 1007.7, 1010.6, 1007.6, 1017.6, 1010.8, 1009.2, 1009.6…
## $ pressure_3pm    <dbl> 1007.1, 1007.8, 1008.7, 1012.8, 1006.0, 1005.4, 1008.2…
## $ cloud_9am       <int> 8, NA, NA, NA, 7, NA, 1, NA, NA, NA, NA, 8, 8, NA, NA,…
## $ cloud_3pm       <int> NA, NA, 2, NA, 8, NA, NA, NA, NA, NA, NA, 8, 8, 7, NA,…
## $ temp_9am        <dbl> 16.9, 17.2, 21.0, 18.1, 17.8, 20.6, 18.1, 16.3, 18.3, …
## $ temp_3pm        <dbl> 21.8, 24.3, 23.2, 26.5, 29.7, 28.9, 24.6, 25.5, 30.2, …
## $ rain_today      <fct> No, No, No, No, No, No, No, No, No, Yes, No, Yes, Yes,…
## $ risk_mm         <dbl> 0.0, 0.0, 0.0, 1.0, 0.2, 0.0, 0.0, 0.0, 1.4, 0.0, 2.2,…
## $ rain_tomorrow   <fct> No, No, No, No, No, No, No, No, Yes, No, Yes, Yes, Yes…
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

10.3 Add Columns

20200814 Adding columns to a dataset is accomplished with dplyr::mutate(). Within a pipeline dplyr::mutate() will modify the data as it passes through.

In the example below we add two new columns to the dataset. A tee pipe is used to print a sample of the resulting dataset using dplyr::select() and dplyr::sample_frac() within the curly braces. The ongoing pipe assigns the result into a new variable.

```
ds %>%
  mutate(range_temp=max_temp-min_temp,
         describe_temp=case_when(max_temp > 30 ~ "hot",
                                 max_temp > 20 ~ "mild",
                                 max_temp >  0 ~ "cold",
                                 TRUE          ~ "freezing")) %T>%
  {
    select(., date, location, ends_with("_temp")) %>%
    sample_frac() %>%
    print()
  } ->
newds
```

```
## # A tibble: 176,747 x 6
##    date       location      min_temp max_temp range_temp describe_temp
##    <date>     <chr>            <dbl>    <dbl>      <dbl> <chr>
##  1 2015-06-28 SydneyAirport      6.9     17.6       10.7 cold
##  2 2013-06-15 MountGambier       7.5     15.3        7.8 cold
##  3 2010-01-23 Bendigo           14.2     27         12.8 mild
##  4 2011-07-06 Cairns            11.4     26.6       15.2 mild
##  5 2013-04-01 Tuggeranong        7.4     22.2       14.8 mild
##  6 2011-03-08 Adelaide          21.4     23.2        1.8 mild
##  7 2010-06-13 PearceRAAF         7.8     24.4       16.6 mild
##  8 2015-08-20 Ballarat           2.5     14         11.5 cold
##  9 2018-01-15 Canberra           8.3     26         17.7 mild
## 10 2018-08-17 Launceston         3.5     12.7        9.2 cold
## # … with 176,737 more rows
```

To overwrite the original dataset instead of saving it as a new dataset, replace the first pipe with an assignment pipe magrittr::%<>%.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

10.4 Add Columns Using Variables

20200814 New columns can be added to a dataset where the new names are generated or supplied as the value of a variable. The example below uses uses the non-standard evaluation constructs and is generally useful when defining functions and the new variable name is passed through the function call. The notation includes the double exclamations (bang bang) which, technically, unquotes the following variable to replace it with its actual value. The colon equals is required in order to support unquoting the left hand side.

```
newvar <- "mid_temp"


ds %>%
  mutate(  newvar  = (max_temp + min_temp)/2,
         !!newvar := (max_temp + min_temp)/2) %T>%
  {
    select(., date, location, ends_with("_temp"), newvar) %>%
    sample_frac() %>%
    print()
  } ->
newds
```

```
## # A tibble: 176,747 x 6
##    date       location         min_temp max_temp mid_temp newvar
##    <date>     <chr>               <dbl>    <dbl>    <dbl>  <dbl>
##  1 2014-09-05 Uluru                14.5     20.2     17.4   17.4
##  2 2009-05-14 Bendigo               5.1     17.4     11.2   11.2
##  3 2014-02-03 Adelaide             23.6     28.3     26.0   26.0
##  4 2017-03-31 Adelaide              8.8     20.1     14.4   14.4
##  5 2013-04-19 AliceSprings         13.1     32.6     22.8   22.8
##  6 2019-07-19 PerthAirport         10.7     19.6     15.2   15.2
##  7 2015-08-09 WaggaWagga            0.6     12.8      6.7    6.7
##  8 2018-06-17 Townsville           15.3     24.7     20     20
##  9 2018-09-07 MelbourneAirport      5.8     14.4     10.1   10.1
## 10 2018-09-09 NorahHead            10.7     22       16.4   16.4
## # … with 176,737 more rows
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

10.5 Add Counts

20200814 Using dplyr::add_count() a new column will be added to the dataset recording the size of groups. The column name will be `n` .

```
ds %<>%
  add_count(location) %T>%
  {
    select(., date, location, n) %>%
    sample_frac() %>%
    print()
  }
```

```
## # A tibble: 176,747 x 3
##    date       location             n
##    <date>     <chr>            <int>
##  1 2009-02-11 MountGinini       3680
##  2 2009-06-01 PerthAirport      3648
##  3 2010-05-12 MelbourneAirport  3649
##  4 2018-08-17 Sale              3649
##  5 2015-03-15 Hobart            3833
##  6 2017-07-19 Canberra          4076
##  7 2010-12-12 Brisbane          3833
##  8 2013-04-06 Richmond          3649
##  9 2008-10-11 Perth             3832
## 10 2014-12-16 NorfolkIsland     3649
## # … with 176,737 more rows
```

```
names(ds)
```

```
##  [1] "date"            "location"        "min_temp"        "max_temp"
##  [5] "rainfall"        "evaporation"     "sunshine"        "wind_gust_dir"
##  [9] "wind_gust_speed" "wind_dir_9am"    "wind_dir_3pm"    "wind_speed_9am"
## [13] "wind_speed_3pm"  "humidity_9am"    "humidity_3pm"    "pressure_9am"
## [17] "pressure_3pm"    "cloud_9am"       "cloud_3pm"       "temp_9am"
## [21] "temp_3pm"        "rain_today"      "risk_mm"         "rain_tomorrow"
## [25] "n"
```

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

10.6 Combine Rows

20201026 We can utilise dplyr::bind_rows() to combine the rows of two datasets with the same columns:

```
bind_rows(head(ds), tail(ds))
```

```
## # A tibble: 12 x 24
##    date       location min_temp max_temp rainfall evaporation sunshine
##    <date>     <chr>       <dbl>    <dbl>    <dbl>       <dbl>    <dbl>
##  1 2008-12-01 Albury       13.4     22.9      0.6          NA       NA
##  2 2008-12-02 Albury        7.4     25.1      0            NA       NA
##  3 2008-12-03 Albury       12.9     25.7      0            NA       NA
##  4 2008-12-04 Albury        9.2     28        0            NA       NA
##  5 2008-12-05 Albury       17.5     32.3      1            NA       NA
##  6 2008-12-06 Albury       14.6     29.7      0.2          NA       NA
##  7 2020-04-24 Uluru        23       36.7      0            NA       NA
##  8 2020-04-25 Uluru        18.4     37.4      0            NA       NA
##  9 2020-04-26 Uluru        21.4     32.7      0            NA       NA
## 10 2020-04-27 Uluru        19.4     32.2      0            NA       NA
## 11 2020-04-28 Uluru        16.6     32.6      0            NA       NA
## 12 2020-04-29 Uluru        16.7     25.7      0            NA       NA
## # … with 17 more variables: wind_gust_dir <ord>, wind_gust_speed <dbl>,
## #   wind_dir_9am <ord>, wind_dir_3pm <ord>, wind_speed_9am <dbl>,
## #   wind_speed_3pm <dbl>, humidity_9am <int>, humidity_3pm <int>,
## #   pressure_9am <dbl>, pressure_3pm <dbl>, cloud_9am <int>, cloud_3pm <int>,
## #   temp_9am <dbl>, temp_3pm <dbl>, rain_today <fct>, risk_mm <dbl>,
## #   rain_tomorrow <fct>
```

The operation performed by dplyr::bind_rows() is also useful to convert a named list (or vector if all values are of the same type) into a tibble:

```r
ds %>%
  head() %>%
  select(2:5) %>%
  bind_rows(list(location="Somewhere", min_temp=-10,
                 max_temp=30, rainfall=10))
```

```
## # A tibble: 7 x 4
##   location  min_temp max_temp rainfall
##   <chr>        <dbl>    <dbl>    <dbl>
## 1 Albury        13.4     22.9      0.6
## 2 Albury         7.4     25.1      0
## 3 Albury        12.9     25.7      0
## 4 Albury         9.2     28        0
## 5 Albury        17.5     32.3      1
## 6 Albury        14.6     29.7      0.2
## 7 Somewhere    -10       30       10
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

10.7 Counting Groups

```
ds %>%
  count(location) %>%
  sample_frac()
```

```
## # A tibble: 49 x 2
##    location             n
##    <chr>            <int>
##  1 Witchcliffe       3648
##  2 Launceston        3680
##  3 MelbourneAirport  3649
##  4 Tuggeranong       3679
##  5 Newcastle         3680
##  6 NorfolkIsland     3649
##  7 Canberra          4076
##  8 Perth             3832
##  9 Albury            3680
## 10 Watsonia          3649
## # … with 39 more rows
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

10.8 Dollar to Numeric Conversion

20200813 Dollar amounts ingested from other applications might be formatted as a string with a `\$` %$ prefix and commas. Use readr::parse_number() to convert to a numeric rather than a string value. This then allows us to perform calculations on the amounts.

In the example below we use readr::parse_number() within a pipeline. It begins with ingesting the data from a **csv** file using readr::read_csv(). We then normalise the variable names using janitor::clean_names(). A tee pipe is then used to send the data along two pipes. The first pipe (between the curly brackets) will print a sample from the column to be converted, for information.

It is the other pipe that continues along the pipeline to perform the actual conversion using dplyr::mutate(). Here readr::parse_number() is applied to the appropriate column to effect the transformation of the dollar amount from string to numeric.

Another tee pipe is then used to print the result, for information and confirmation, whilst also assigning the dataset into a variable.

```
IFILE <- "data/dollars.csv"


IFILE %>%
  read_csv() %>%
  clean_names(numerals="right") %T>%
  {pull(., income) %>% head(20) %>% print(); cat("\n")} %>%
  mutate(income=parse_number(income)) %T>%
  {pull(., income) %>% head(20) %>% print(); cat("\n")} ->
dollars
```

```
##
## ── Column specification ────────────────────────────────────────────────────────
## cols(
##   id = col_double(),
##   age = col_double(),
##   income = col_character()
## )
```

```
##  [1] "$81,838.00"  "$72,099.00"  "$154,676.74" "$27,743.82"  "$7,568.23"
##  [6] "$33,144.40"  "$43,391.17"  "$59,906.65"  "$126,888.91" "$52,466.49"
## [11] "$291,416.11" "$24,155.31"  "$143,254.86" "$120,554.81" "$34,919.16"
## [16] "$67,176.79"  "$9,608.48"   "$12,475.84"  "$32,963.39"  "$31,534.97"
##
##  [1]  81838.00  72099.00 154676.74  27743.82   7568.23  33144.40  43391.17
##  [8]  59906.65 126888.91  52466.49 291416.11  24155.31 143254.86 120554.81
## [15]  34919.16  67176.79   9608.48  12475.84  32963.39  31534.97
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

10.9 Extract Column as Vector

20200813 To extract a single column from a dataset and return it as a vector rather than as a dataset column we can use dplyr::pull().

In the example below all of the values of a particular column are pulled and we retain just the base::unique() values from the resulting vector.

Using a tee pipe, the first pipe (between the curly brackets) will print a number of the locations, whilst the continuing pipe flows the vector on to be assigned into a variable for later use.

```
ds %>%
  pull(location) %>%
  unique() %T>%
  {head(., 20) %>% print()} ->
locations
```

```
##  [1] "Albury"        "BadgerysCreek" "Cobar"         "CoffsHarbour"
##  [5] "Moree"         "Newcastle"     "NorahHead"     "NorfolkIsland"
##  [9] "Penrith"       "Richmond"      "Sydney"        "SydneyAirport"
## [13] "WaggaWagga"    "Williamtown"   "Wollongong"    "Canberra"
## [17] "Tuggeranong"   "MountGinini"   "Ballarat"      "Bendigo"
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

10.10 Filter Rows Having Missing Values

20201202 To select the rows from a dataset which have missing values in any of the columns across the dataset we stats::filter() dplyr::across() tidyr::everything() that base::is.na() and reduce it within the stats::filter() using the or operator. In the example we randomly sample a few rows and columns to show the result.

```
ds %>%
  filter(across(everything(), is.na) %>% reduce(`|`)) %>%
  sample_frac() %>%
  select(date, location, sample(3:length(vars), 4))
```

```
## # A tibble: 114,944 x 6
##    date       location      wind_gust_speed temp_9am sunshine temp_3pm
##    <date>     <chr>                   <dbl>    <dbl>    <dbl>    <dbl>
##  1 2015-01-20 NorahHead                  48     20.2     NA       23.1
##  2 2009-10-02 Canberra                   NA     12.6     NA       16.9
##  3 2014-07-13 Williamtown                59     10.3     NA       14.4
##  4 2014-12-05 BadgerysCreek              59     21.2     NA       30.5
##  5 2018-01-30 BadgerysCreek              41     24.1     NA       35.3
##  6 2009-02-21 Albury                     28     19.3     NA       29.8
##  7 2012-01-28 Woomera                    65     26.8     NA       31.8
##  8 2015-01-18 Williamtown                37     23.8     NA       26.8
##  9 2010-03-28 MountGinini                41     13.4     NA       18.1
## 10 2009-08-10 Albany                     NA     11.5      2.3     17.1
## # … with 114,934 more rows
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open*

source software since 1984.

Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

10.11 Missing Value Imputation

20201026 See Section 10.16 to replace missing values with specific values.

Missing value imputation is useful but must be done with care. It can be akin to inventing new data. We may be tempted to do so as a quick fix for avoiding warnings that would otherwise advise us of missing data when using , for example. We can utilise the imputation function randomForest::na.roughfix() to perform missing value imputation through the use of machine learning to fill in the gaps. This particular function operates on numeric and factor columns, thus we remove the first two columns from the dataset to be imputed (date and location),

```
# Count the number of missing values.

ds %>% is.na() %>% sum()
```

```
## [1] 464300
```

```
# No missing values in the first two columns (date and location)

ds[1:2] %>% is.na() %>% sum()
```

```
## [1] 0
```

```
# Impute missing values.

ds[3:24] %<>% na.roughfix()

# Confirm that no missing values remain.

ds %>% is.na() %>% sum()
```

```
## [1] 0
```

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

10.12 Modify Columns

```
ds %>%
  select(date, location, pressure_9am)
```

```
## # A tibble: 176,747 x 3
##    date       location pressure_9am
##    <date>     <chr>           <dbl>
##  1 2008-12-01 Albury          1008.
##  2 2008-12-02 Albury          1011.
##  3 2008-12-03 Albury          1008.
##  4 2008-12-04 Albury          1018.
##  5 2008-12-05 Albury          1011.
##  6 2008-12-06 Albury          1009.
##  7 2008-12-07 Albury          1010.
##  8 2008-12-08 Albury          1013.
##  9 2008-12-09 Albury          1009.
## 10 2008-12-10 Albury          1007
## # … with 176,737 more rows
```

```
ds %>%
  mutate(pressure_9am=round(pressure_9am, 0)) %>%
  select(date, location, pressure_9am)
```

```
## # A tibble: 176,747 x 3
##    date       location pressure_9am
##    <date>     <chr>           <dbl>
##  1 2008-12-01 Albury           1008
##  2 2008-12-02 Albury           1011
##  3 2008-12-03 Albury           1008
##  4 2008-12-04 Albury           1018
##  5 2008-12-05 Albury           1011
##  6 2008-12-06 Albury           1009
##  7 2008-12-07 Albury           1010
##  8 2008-12-08 Albury           1013
##  9 2008-12-09 Albury           1009
## 10 2008-12-10 Albury           1007
## # … with 176,737 more rows
```

To convert all character columns to be numeric:

```
ds %>%
  mutate(across(where(is.character), as.numeric))
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

10.13 Normalise Variables

20200912 To rename variables in a dataset we can use dplyr::rename_with() which can apply a function, like rattle::normVarNames(), to the variable names and replace those names with the result from the function. A tidy alternative is to use janitor::clean_names() with the option `numerals=``"right"` to replicate rattle::normVarNames().

The choice of variable naming style is suggested in Chapter 27. all variable names are lowercase with words separated by the underscore. This normalisation is useful when different upper/lower case conventions are intermixed inconsistently in names like `Incm_tax_PyBl`. Remembering how to capitalize when interactively exploring the data with thousands of such variables can be quite a cognitive load. Yet we often see such variable names arising in practise especially when we import data from databases which are often case insensitive.

The example below shows the transformation into the preferred normalised form.

```
# Normalise variable names.

library(janitor)      # Cleanup: clean_names().

names(ds)
```

```
##  [1] "Date"          "Location"      "MinTemp"       "MaxTemp"
##  [5] "Rainfall"      "Evaporation"   "Sunshine"      "WindGustDir"
##  [9] "WindGustSpeed" "WindDir9am"    "WindDir3pm"    "WindSpeed9am"
## [13] "WindSpeed3pm"  "Humidity9am"   "Humidity3pm"   "Pressure9am"
## [17] "Pressure3pm"   "Cloud9am"      "Cloud3pm"      "Temp9am"
## [21] "Temp3pm"       "RainToday"     "RISK_MM"       "RainTomorrow"
```

```
ds %<>%
  clean_names(numerals="right")

names(ds)
```

```
##  [1] "date"            "location"        "min_temp"        "max_temp"
##  [5] "rainfall"        "evaporation"     "sunshine"        "wind_gust_dir"
##  [9] "wind_gust_speed" "wind_dir_9am"    "wind_dir_3pm"    "wind_speed_9am"
## [13] "wind_speed_3pm"  "humidity_9am"    "humidity_3pm"    "pressure_9am"
## [17] "pressure_3pm"    "cloud_9am"       "cloud_3pm"       "temp_9am"
## [21] "temp_3pm"        "rain_today"      "risk_mm"         "rain_tomorrow"
```

Notice the use of the assignment pipe here as introduced in Chapter 2}. We will recall that the operator pipes the left-hand data to the function on the right-hand side and then returns the result to the left-hand side overwriting the original contents of the memory referred to on the left-hand side.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

10.14 Pivot Pairwise Binary Table

20210224 The use case is to turn a table with three columns, consisting of two columns of binary values and the third some measure, into a confusion type matrix.

```
tibble(a=c(0, 0, 1, 1),
       b=c(0, 1, 0, 1),
       c=c(8346, 275, 2736, 945)) %T>%
  print() %>%
  tidyr::pivot_wider(names_from=b, values_from=c)
```

```
## # A tibble: 4 x 3
##       a     b     c
##   <dbl> <dbl> <dbl>
## 1     0     0  8346
## 2     0     1   275
## 3     1     0  2736
## 4     1     1   945
```

```
## # A tibble: 2 x 3
##       a   `0`   `1`
##   <dbl> <dbl> <dbl>
## 1     0  8346   275
## 2     1  2736   945
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open*

source software since 1984.

Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

10.15 Rename Variables

20201202 We can rename variables within a pipeline. To normalise the variables automatically see Section 10.13. If we have a set of new names that we want to use instead of the original names or instead of the automatically normalised names, then use dplyr::rename() in a pipeline.

The original variable names are obtained using base::names():

```
names(ds)
```

```
##  [1] "Date"          "Location"      "MinTemp"       "MaxTemp"
##  [5] "Rainfall"      "Evaporation"   "Sunshine"      "WindGustDir"
##  [9] "WindGustSpeed" "WindDir9am"    "WindDir3pm"    "WindSpeed9am"
## [13] "WindSpeed3pm"  "Humidity9am"   "Humidity3pm"   "Pressure9am"
## [17] "Pressure3pm"   "Cloud9am"      "Cloud3pm"      "Temp9am"
## [21] "Temp3pm"       "RainToday"     "RISK_MM"       "RainTomorrow"
```

We can rename the variables based on their name using dplyr::rename():

```
# Rename the variables based on their names.

ds %>%
  rename(date=Date, location=Location, min_temp=MinTemp, max_temp=MaxTemp)
```

```
## # A tibble: 176,747 x 24
##    date       location min_temp max_temp Rainfall Evaporation Sunshine
##    <date>     <chr>       <dbl>    <dbl>    <dbl>       <dbl>    <dbl>
##  1 2008-12-01 Albury       13.4     22.9      0.6          NA       NA
##  2 2008-12-02 Albury        7.4     25.1      0            NA       NA
##  3 2008-12-03 Albury       12.9     25.7      0            NA       NA
##  4 2008-12-04 Albury        9.2     28        0            NA       NA
##  5 2008-12-05 Albury       17.5     32.3      1            NA       NA
##  6 2008-12-06 Albury       14.6     29.7      0.2          NA       NA
##  7 2008-12-07 Albury       14.3     25        0            NA       NA
##  8 2008-12-08 Albury        7.7     26.7      0            NA       NA
##  9 2008-12-09 Albury        9.7     31.9      0            NA       NA
## 10 2008-12-10 Albury       13.1     30.1      1.4          NA       NA
## # … with 176,737 more rows, and 17 more variables: WindGustDir <ord>,
## #   WindGustSpeed <dbl>, WindDir9am <ord>, WindDir3pm <ord>,
## #   WindSpeed9am <dbl>, WindSpeed3pm <dbl>, Humidity9am <int>,
## #   Humidity3pm <int>, Pressure9am <dbl>, Pressure3pm <dbl>, Cloud9am <int>,
## #   Cloud3pm <int>, Temp9am <dbl>, Temp3pm <dbl>, RainToday <fct>,
## #   RISK_MM <dbl>, RainTomorrow <fct>
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

10.16 Replacing Missing Values

20201026 See Section 10.11 to impute (guess) values to replace missing values.

To replace missing values (`NA`) in a data set with a specific default value, like 0 for numeric data, we can use tidyr::replace_na() within a pipeline. In the following example only the numeric columns of the dataset are considered dplyr::across() the dataset, by checking tidyselect::where() the data base::is.numeric().

```
ds %>%
  mutate(across(where(is.numeric), ~replace_na(.x, 0)))
```

```
## # A tibble: 176,747 x 24
##    date       location min_temp max_temp rainfall evaporation sunshine
##    <date>     <chr>       <dbl>    <dbl>    <dbl>       <dbl>    <dbl>
##  1 2008-12-01 Albury       13.4     22.9      0.6         4.8      8.5
##  2 2008-12-02 Albury        7.4     25.1      0           4.8      8.5
##  3 2008-12-03 Albury       12.9     25.7      0           4.8      8.5
##  4 2008-12-04 Albury        9.2     28        0           4.8      8.5
##  5 2008-12-05 Albury       17.5     32.3      1           4.8      8.5
##  6 2008-12-06 Albury       14.6     29.7      0.2         4.8      8.5
##  7 2008-12-07 Albury       14.3     25        0           4.8      8.5
##  8 2008-12-08 Albury        7.7     26.7      0           4.8      8.5
##  9 2008-12-09 Albury        9.7     31.9      0           4.8      8.5
## 10 2008-12-10 Albury       13.1     30.1      1.4         4.8      8.5
## # … with 176,737 more rows, and 17 more variables: wind_gust_dir <ord>,
## #   wind_gust_speed <dbl>, wind_dir_9am <ord>, wind_dir_3pm <ord>,
## #   wind_speed_9am <dbl>, wind_speed_3pm <dbl>, humidity_9am <dbl>,
## #   humidity_3pm <dbl>, pressure_9am <dbl>, pressure_3pm <dbl>,
## #   cloud_9am <dbl>, cloud_3pm <dbl>, temp_9am <dbl>, temp_3pm <dbl>,
## #   rain_today <fct>, risk_mm <dbl>, rain_tomorrow <fct>
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

10.17 Subset of Rows Within Groups

20200814 The dplyr::slice() command understands groups and so the indicies supplied will apply to each group. To select, for example, the second row of each group, simply supply `2` as the argument. Notice the rows selected are all on the second of the month.

```
ds %>%
  group_by(location) %>%
  slice(2)
```

```
## # A tibble: 49 x 24
## # Groups:   location [49]
##    date       location      min_temp max_temp rainfall evaporation sunshine
##    <date>     <chr>            <dbl>    <dbl>    <dbl>       <dbl>    <dbl>
##  1 2008-07-02 Adelaide          12.7     15.8      0.8         1.4      7.8
##  2 2008-12-02 Albany            14.7     21        0           5.4      9.1
##  3 2008-12-02 Albury             7.4     25.1      0           4.8      8.5
##  4 2008-12-02 AliceSprings      21       39.1      1.2         9       12.2
##  5 2009-01-02 BadgerysCreek     14.7     26.1      0           4.8      8.5
##  6 2008-12-02 Ballarat           4.3     19.5      0.4         4.8      8.5
##  7 2008-12-02 Bendigo            6.2     23        0           8.8      8.5
##  8 2008-07-02 Brisbane           8.1     22.9      0           2        9.8
##  9 2008-12-02 Cairns            24.2     32.3      8.4         6.6      5.3
## 10 2007-11-02 Canberra          14       26.9      3.6         4.4      9.7
## # … with 39 more rows, and 17 more variables: wind_gust_dir <ord>,
## #   wind_gust_speed <dbl>, wind_dir_9am <ord>, wind_dir_3pm <ord>,
## #   wind_speed_9am <dbl>, wind_speed_3pm <dbl>, humidity_9am <dbl>,
## #   humidity_3pm <dbl>, pressure_9am <dbl>, pressure_3pm <dbl>,
## #   cloud_9am <dbl>, cloud_3pm <dbl>, temp_9am <dbl>, temp_3pm <dbl>,
## #   rain_today <fct>, risk_mm <dbl>, rain_tomorrow <fct>
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

10.18 Data Source

20180908 We begin with identifying a data source and choose the simplest of sources—a text-based **csv** (comma separated value) file as a typical data source format.

The rattle::weatherAUS dataset from rattle (G. Williams 2020) will again be used. A binary formatted R dataset is provided by the package but the CSV file for the same dataset is available at https://rattle.togaware.com/weatherAUS.csv.

Identify and record the location of the CSV file to analyse. R can ingesting data directly from the Internet and so we will illustrate that here. The location of the file (the so-called URL or universal resource location) will be saved as a string in a variable called `dspath` —the path to the dataset. The following assignment command does this for us. Simply type this into your R script file within RStudio. The command is then executed in RStudio by clicking the `Run` button whilst the cursor is situated on the line within the script file.

```
# Note the source location of a dataset to ingest into R.

dspath <- "http://rattle.togaware.com/weatherAUS.csv"
```

The assignment operator `<-` will store the value on the right hand side (which is a string enclosed within quotation marks) into the computer's memory and we can later refer to it as the R variable `dspath` —we retrieve the string simply by reference to the variable `dspath`.

By typing the name of the variable (`dspath`) in the R Console at the `>` prompt R will respond with the value stored in the variable:

```
dspath
```

```
## [1] "http://rattle.togaware.com/weatherAUS.csv"
```

If not connected to the Internet we can read the data directly from a local copy of the **csv** file. The rattle (G. Williams 2020) package (once the package has been installed) provides a smaller sample . The location of the CSV file within rattle (G. Williams 2020) is determined using base::system.file(). Knowing that **csv** files are located within the `csv` sub-directory of the rattle (G. Williams 2020) package we generate the string that identifies the file system path to .

This is the path to the CSV file on my file system. Your path may well be different depending on where your system installed the rattle (G. Williams 2020) package.

Note that this is a considerably smaller subset of the full weatherAUS dataset and ingesting this rather than the full dataset will lead to different results to those presented here.

If you have separately downloaded then you can identify its location. Here we identify that the downloaded file is located in the current working directory.

```
dspath <- "./weatherAUS.csv"
```

References

Williams, Graham. 2020. *Rattle: Graphical User Interface for Data Science in r*. https://rattle.togaware.com/.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

10.19 Data Ingestion

20180721 Having identified the source of the dataset we can ingest the dataset into the memory of the computer using readr::read_csv() which returns an enhanced .

We set up a reference to the data frame's location in the computer's memory by assigning the result of the call to the function readr::read_csv() to the R variable `weather` .

```
# Ingest the dataset.

weatherAUS <- read_csv(file=dspath)
```

As a side effect of calling the function readr::read_csv() helpful messages are displayed that identify the data types for each of the variables found in the ingested dataset. We should review these to ensure they match our expectations. If they don't, there are optional arguments to readr::read_csv() to inform it otherwise.

Note that the rattle (G. Williams 2020) also provides a smaller rattle::weather dataset as an R dataset, also named `weather` . Simply by attaching the rattle (G. Williams 2020) package from the library a variable called `weather` becomes available. Running the above command will replace the dataset provided by rattle (G. Williams 2020). Having done so we can still access the `weather` dataset provided by rattle (G. Williams 2020) using the package prefix as in rattle::weather.

References

Williams, Graham. 2020. *Rattle: Graphical User Interface for Data Science in r*. https://rattle.togaware.com/.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open*

source software since 1984.

Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

10.20 The Shape of the Dataset

20180721 Once the dataset is loaded we want to get a basic idea of what it looks like—its shape. Being an extended data frame (what we call a tibble), we can display the data as a tibble simply by printing the data referred to by the variable name.

```
# Print the dataset in a human useful way.

weather
```

```
## # A tibble: 366 x 24
##    Date       Location MinTemp MaxTemp Rainfall Evaporation Sunshine WindGustDir
##    <date>     <chr>      <dbl>   <dbl>    <dbl>       <dbl>    <dbl> <ord>
##  1 2007-11-01 Canberra     8      24.3      0           3.4      6.3 NW
##  2 2007-11-02 Canberra    14      26.9      3.6         4.4      9.7 ENE
##  3 2007-11-03 Canberra    13.7    23.4      3.6         5.8      3.3 NW
##  4 2007-11-04 Canberra    13.3    15.5     39.8         7.2      9.1 NW
##  5 2007-11-05 Canberra     7.6    16.1      2.8         5.6     10.6 SSE
##  6 2007-11-06 Canberra     6.2    16.9      0           5.8      8.2 SE
##  7 2007-11-07 Canberra     6.1    18.2      0.2         4.2      8.4 SE
##  8 2007-11-08 Canberra     8.3    17        0           5.6      4.6 E

##  9 2007-11-09 Canberra     8.8    19.5      0           4        4.1 S
## 10 2007-11-10 Canberra     8.4    22.8     16.2         5.4      7.7 E
## # … with 356 more rows, and 16 more variables: WindGustSpeed <dbl>,
## #   WindDir9am <ord>, WindDir3pm <ord>, WindSpeed9am <dbl>, WindSpeed3pm <dbl>,
## #   Humidity9am <int>, Humidity3pm <int>, Pressure9am <dbl>, Pressure3pm <dbl>,
## #   Cloud9am <int>, Cloud3pm <int>, Temp9am <dbl>, Temp3pm <dbl>,
## #   RainToday <fct>, RISK_MM <dbl>, RainTomorrow <fct>
```

We observe that dataset consists of observations of variables. The enhanced nature of the data frame that representing it as a tibble brings to us is that the printout is more informative. The first few observations are shown with a subset of the variables followed by a list of all of the other variables.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

10.21 A Glimpse of the Dataset

20180721 A useful alternative to gain some insight into the dataset is through pillar::glimpse().

```
# A quick view of the contents of the dataset.

glimpse(weather)
```

```
## Rows: 366
## Columns: 24
## $ Date          <date> 2007-11-01, 2007-11-02, 2007-11-03, 2007-11-04, 2007-11…
## $ Location      <chr> "Canberra", "Canberra", "Canberra", "Canberra", "Canberr…
## $ MinTemp       <dbl> 8.0, 14.0, 13.7, 13.3, 7.6, 6.2, 6.1, 8.3, 8.8, 8.4, 9.1…
## $ MaxTemp       <dbl> 24.3, 26.9, 23.4, 15.5, 16.1, 16.9, 18.2, 17.0, 19.5, 22…
## $ Rainfall      <dbl> 0.0, 3.6, 3.6, 39.8, 2.8, 0.0, 0.2, 0.0, 0.0, 16.2, 0.0,…
## $ Evaporation   <dbl> 3.4, 4.4, 5.8, 7.2, 5.6, 5.8, 4.2, 5.6, 4.0, 5.4, 4.2, 7…
## $ Sunshine      <dbl> 6.3, 9.7, 3.3, 9.1, 10.6, 8.2, 8.4, 4.6, 4.1, 7.7, 11.9,…
## $ WindGustDir   <ord> NW, ENE, NW, NW, SSE, SE, SE, E, S, E, N, E, WNW, NW, NW…
## $ WindGustSpeed <dbl> 30, 39, 85, 54, 50, 44, 43, 41, 48, 31, 30, 41, 30, 44, …
## $ WindDir9am    <ord> SW, E, N, WNW, SSE, SE, SE, SE, E, S, SE, E, S, WNW, S, …
## $ WindDir3pm    <ord> NW, W, NNE, W, ESE, E, ESE, E, ENE, ESE, NW, NW, NW, W, …
## $ WindSpeed9am  <dbl> 6, 4, 6, 30, 20, 20, 19, 11, 19, 7, 6, 2, 6, 7, 6, 7, 6,…
## $ WindSpeed3pm  <dbl> 20, 17, 6, 24, 28, 24, 26, 24, 17, 6, 9, 15, 7, 20, 20, …
## $ Humidity9am   <int> 68, 80, 82, 62, 68, 70, 63, 65, 70, 82, 74, 54, 62, 67, …
## $ Humidity3pm   <int> 29, 36, 69, 56, 49, 57, 47, 57, 48, 32, 34, 35, 29, 20, …
## $ Pressure9am   <dbl> 1019.7, 1012.4, 1009.5, 1005.5, 1018.3, 1023.8, 1024.6, …
## $ Pressure3pm   <dbl> 1015.0, 1008.4, 1007.2, 1007.0, 1018.5, 1021.7, 1022.2, …
## $ Cloud9am      <int> 7, 5, 8, 2, 7, 7, 4, 6, 7, 7, 1, 0, 0, 1, 0, 0, 7, 1, 1,…
## $ Cloud3pm      <int> 7, 3, 7, 7, 7, 5, 6, 7, 7, 1, 2, 3, 1, 4, 1, 3, 6, 5, 3,…
## $ Temp9am       <dbl> 14.4, 17.5, 15.4, 13.5, 11.1, 10.9, 12.4, 12.1, 14.1, 13…
## $ Temp3pm       <dbl> 23.6, 25.7, 20.2, 14.1, 15.4, 14.8, 17.3, 15.5, 18.9, 21…
## $ RainToday     <fct> No, Yes, Yes, Yes, Yes, No, No, No, No, Yes, No, No, No,…
## $ RISK_MM       <dbl> 3.6, 3.6, 39.8, 2.8, 0.0, 0.2, 0.0, 0.0, 16.2, 0.0, 0.2,…
## $ RainTomorrow  <fct> Yes, Yes, Yes, Yes, No, No, No, No, Yes, No, No, No, No,…
```

Again we receive a printed summary of the dataset, reporting on the number of observations and variables, but now the table is effectively rotated so that all variables can be listed along with their data type and a selection of their values for the first few observations.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

10.22 Introducing Template Variables

20180721 A reference to the original dataset can be created using a template (or generic) variable. The new variable will be called `ds` (short for dataset).

```
# Take a copy of the dataset into a generic variable.

ds <- weatherAUS
```

Both `ds` and `weatherAUS` will now reference the same dataset within the computer's memory. As we modify `ds` those modifications will only affect the data referenced by `ds`. Effectively, an extra copy of the dataset in the computer's memory will start to grow as we change the data from its original form. R avoids making copies of datasets unnecessarily and so a simple assignment does not create a new copy. As modifications are made to one or the other copy of a dataset then extra memory will be used to store the columns that differ between the datasets.

From here on we no longer refer to the dataset as `weather` but as `ds`. This allows the following analyses and processing to be rather generic—turning the R code into a *template* and so requiring only minor modification when used with a different dataset assigned into `ds`.

Often we will find that we can simply load a different dataset into memory, store it as `ds` and the remaining steps of our analyses and processing will essentially work unchanged.

The first few steps of our template are then captured as creating the reference to the dataset and presenting our initial view of the dataset.

```
# Prepare for a templated analysis and processing.

dsname <- "weatherAUS"
ds     <- get(dsname)
ds %<>% clean_names(numerals="right")
glimpse(ds)
```

```
## Rows: 176,747
## Columns: 24
## $ date            <date> 2008-12-01, 2008-12-02, 2008-12-03, 2008-12-04, 2008-…
## $ location        <chr> "Albury", "Albury", "Albury", "Albury", "Albury", "Alb…
## $ min_temp        <dbl> 13.4, 7.4, 12.9, 9.2, 17.5, 14.6, 14.3, 7.7, 9.7, 13.1…
## $ max_temp        <dbl> 22.9, 25.1, 25.7, 28.0, 32.3, 29.7, 25.0, 26.7, 31.9, …
## $ rainfall        <dbl> 0.6, 0.0, 0.0, 0.0, 1.0, 0.2, 0.0, 0.0, 0.0, 1.4, 0.0,…
## $ evaporation     <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA…
## $ sunshine        <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA…
## $ wind_gust_dir   <ord> W, WNW, WSW, NE, W, WNW, W, W, NNW, W, N, NNE, W, SW, …
## $ wind_gust_speed <dbl> 44, 44, 46, 24, 41, 56, 50, 35, 80, 28, 30, 31, 61, 44…
## $ wind_dir_9am    <ord> W, NNW, W, SE, ENE, W, SW, SSE, SE, S, SSE, NE, NNW, W…
## $ wind_dir_3pm    <ord> WNW, WSW, WSW, E, NW, W, W, W, NW, SSE, ESE, ENE, NNW,…
## $ wind_speed_9am  <dbl> 20, 4, 19, 11, 7, 19, 20, 6, 7, 15, 17, 15, 28, 24, 4,…
## $ wind_speed_3pm  <dbl> 24, 22, 26, 9, 20, 24, 24, 17, 28, 11, 6, 13, 28, 20, …
## $ humidity_9am    <int> 71, 44, 38, 45, 82, 55, 49, 48, 42, 58, 48, 89, 76, 65…
## $ humidity_3pm    <int> 22, 25, 30, 16, 33, 23, 19, 19, 9, 27, 22, 91, 93, 43,…
## $ pressure_9am    <dbl> 1007.7, 1010.6, 1007.6, 1017.6, 1010.8, 1009.2, 1009.6…
## $ pressure_3pm    <dbl> 1007.1, 1007.8, 1008.7, 1012.8, 1006.0, 1005.4, 1008.2…
## $ cloud_9am       <int> 8, NA, NA, NA, 7, NA, 1, NA, NA, NA, NA, 8, 8, NA, NA,…
## $ cloud_3pm       <int> NA, NA, 2, NA, 8, NA, NA, NA, NA, NA, NA, 8, 8, 7, NA,…
## $ temp_9am        <dbl> 16.9, 17.2, 21.0, 18.1, 17.8, 20.6, 18.1, 16.3, 18.3, …
## $ temp_3pm        <dbl> 21.8, 24.3, 23.2, 26.5, 29.7, 28.9, 24.6, 25.5, 30.2, …
## $ rain_today      <fct> No, No, No, No, No, No, No, No, No, Yes, No, Yes, Yes,…
## $ risk_mm         <dbl> 0.0, 0.0, 0.0, 1.0, 0.2, 0.0, 0.0, 0.0, 1.4, 0.0, 2.2,…
## $ rain_tomorrow   <fct> No, No, No, No, No, No, No, No, Yes, No, Yes, Yes, Yes…
```

We are a little tricky here in recording the dataset name in the variable `dsname` and then using the function base::get() to make a copy of the dataset reference and link it to the generic variable `ds`. We could simply assign the data to `ds` directly as we saw above. Either way the generic variable `ds` refers to the same dataset. The use of base::get() allows us to be a little more generic in our template.

The use of generic variables within a template for the tasks we perform on each new dataset will have obvious advantages but we need to be careful. A disadvantage is that we may be working with several datasets and accidentally overwrite previously processed datasets referenced using the same generic variable (`ds`). The processing of the dataset might take some time and so accidentally losing it is not an attractive proposition. Care needs to be taken to avoid this.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

10.23 Locating Datasets in Memory

20200603 We can see that `ds` and `weather` reference the same dataset in memory using lobstr::ref() which shows actual memory addresses of the data.

```
ref(weather)
ref(ds)
```

This gets rather technical, but the strings of digits beginning with `0x` are the actual memory addresses—that is, they are pointers to a direct location in our computer's memory. The `0x` at the beginning of each identifies that a hexadecimal scheme is used, thus we see digits 0 to 9 and then the letters a, b, c, d, e, and f being used (16 digits).

The addresses recorded for `weather` and `ds`, including the addresses where we find the actual variables (columns) within each dataset, are identical.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

10.24 Changing Datasets in Memory

20180721 Let's make a change to the weather dataset by simply changing a single cell, changing the value of `MinTemp` (the third variable) for the first observation to `5`.

```
weather[1,3] <- 5
```

Notice the divergence of the two datasets. They still share a lot in common, and hence only one copy of that data, but where they diverge, they now use different memory locations.

```
ref(weather)
ref(ds)
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

10.25 Reviewing Variable Names

The names of the variables within the dataset as supplied to us may not be in any particular form and may use different conventions. For example, they may use a mix of upper and lower case letters (`TempToday9AM`) or be very long (`Temperature_Recorded_Today_9am`) or use sequential numbers to identify each variable (`V004` or `V004_rainToday`) or use codes (`XVn34_rain`) or any number of other conventions. Often we prefer to simplify the variable names to ease our processing and thinking and to enforce a standard and consistent naming convention for ourselves.

We use base::names() to list the names of the variables within a dataset.

```
# Review the variables to consider normalising their names.

names(ds)
```

```
##  [1] "date"            "location"        "min_temp"        "max_temp"
##  [5] "rainfall"        "evaporation"     "sunshine"        "wind_gust_dir"
##  [9] "wind_gust_speed" "wind_dir_9am"    "wind_dir_3pm"    "wind_speed_9am"
## [13] "wind_speed_3pm"  "humidity_9am"    "humidity_3pm"    "pressure_9am"
## [17] "pressure_3pm"    "cloud_9am"       "cloud_3pm"       "temp_9am"
## [21] "temp_3pm"        "rain_today"      "risk_mm"         "rain_tomorrow"
```

Notice that the names here use a scheme whereby the initial letter is capitalised and each word within the variable name is also capitalised. That's a reasonable naming scheme and is preferred by some.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open*

source software since 1984.

Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

10.26 Effect on Data Storage

20180721 When the names of the variables within a dataset are changed R does not make a complete new copy of the dataset. Instead, the actual data in the column remains in tack whilst the variable itself (`ds`) references a new memory location where the new variable names get noted. The underlying data within the table is unaffected.

```
location(weather)
```

```
## Warning: `location()` was deprecated in dplyr 1.0.0.
## Please use `lobst::ref()` instead.
```

```
## <0x5639a22b1620>
## Variables:
##  * Date:          <0x5639a03d7520>
##  * Location:      <0x5639a03d80d0>
##  * MinTemp:       <0x5639a22b0870>
##  * MaxTemp:       <0x56399f997030>
##  * Rainfall:      <0x5639a558e610>
##  * Evaporation:   <0x5639a558f1c0>
##  * Sunshine:      <0x5639a4f24d40>
##  * WindGustDir:   <0x56399f997be0>
##  * WindGustSpeed: <0x5639a4f258f0>
##  * WindDir9am:    <0x5639a558fd70>
##  * WindDir3pm:    <0x5639a4f264a0>
##  * WindSpeed9am:  <0x5639a1de2d40>
##  * WindSpeed3pm:  <0x5639a1de38f0>
##  * Humidity9am:   <0x5639a1de44a0>
##  * Humidity3pm:   <0x5639a4a90d60>
##  * Pressure9am:   <0x56399f4befe0>
##  * Pressure3pm:   <0x56399f4bfb90>
##  * Cloud9am:      <0x56399f4c0740>
##  * Cloud3pm:      <0x5639a2fbfed0>
##  * Temp9am:       <0x5639a2fc04c0>
##  * Temp3pm:       <0x5639a2fc1070>
##  * RainToday:     <0x5639a142ec00>
##  * RISK_MM:       <0x5639a142f1f0>
##  * RainTomorrow:  <0x5639a142fda0>
## Attributes:
##  * names:         <0x5639a1cf5400>
##  * row.names:     <0x56399fad8898>
##  * class:         <0x5639a0a4b6c8>
```

```
location(ds)
```

```
## <0x5639a7f06630>
## Variables:
##  * date:            <0x56399e795b50>
##  * location:        <0x563998533d90>
##  * min_temp:        <0x5639a3b65070>
##  * max_temp:        <0x5639a3555bb0>
##  * rainfall:        <0x5639a387f3d0>
##  * evaporation:     <0x5639a39d8760>
##  * sunshine:        <0x56399e0e8a70>
##  * wind_gust_dir:   <0x56399868d120>
##  * wind_gust_speed: <0x56399e241e00>
##  * wind_dir_9am:    <0x5639a0470100>
##  * wind_dir_3pm:    <0x5639a0af3320>
##  * wind_speed_9am:  <0x5639a4000230>
##  * wind_speed_3pm:  <0x5639a41595c0>
##  * humidity_9am:    <0x5639a057c840>
##  * humidity_3pm:    <0x5639a07a6ee0>
##  * pressure_9am:    <0x56399dc0c2c0>
##  * pressure_3pm:    <0x56399dd65650>
##  * cloud_9am:       <0x5639a08a1e00>
##  * cloud_3pm:       <0x5639a0667e60>
##  * temp_9am:        <0x56399debe9e0>
##  * temp_3pm:        <0x5639a46ed660>
##  * rain_today:      <0x5639a0c0b520>
##  * risk_mm:         <0x5639a48469f0>
##  * rain_tomorrow:   <0x5639a1d20f30>
## Attributes:
##  * names:           <0x5639a0eea680>
##  * row.names:       <0x5639a0f94270>
##  * class:           <0x5639a3553ce8>
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

10.27 Special Case Variable Name Transformations

20180721 When reviewing the variables of a dataset we often notice other changes that could be made to the variable names. This might be to simplify the variables or to clarify the meaning of the variable. The string processing functions provided by stringr (Wickham 2019b) come in handy for such processing.

In the following example we remove the prefix of the variable names where we identify that the prefix consists of all characters up to the first underscore. This is useful where a dataset has prefixed each variable by a sequential number or by some other code and we have no real use of such a prefix in our processing.

```
names(ds) %<>% str_replace("^[^_]*_", "")
```

This will take a variable name like `ab123_tax_payable` and convert it to `tax_payable`.

```
str_replace("ab123_tax_payable", "^[^_]*_", "")
```

```
## [1] "tax_payable"
```

The odd looking characters in the argument to stringr::str_replace() are a . Regular expressions are a very powerful concept and can get quite complex. The reader is referred to the many resources on-line that cover regular expressions. The regular expression is a pattern used to match some part of the variable name. The pattern begins with `^` which anchors the match to the beginning of the variable name. This can be followed by zero or more characters (`*`) that do not match the underscore (`[^_]`)—the `*` specifies that the preceding pattern can be repeated zero or more times. The preceding pattern here is actually a list of characters included between square brackets. Since this list begins with `^` the listed characters are excluded from the matching. That is, the pattern preceding the `*` will match any character that is not an

underscore. The third component of the match is then an actual underscore. Combined this regular expression matches any sequence (including an empty sequence) of characters (except for an underscore) that is at the beginning of the variable name and followed by an underscore.

The next argument to stringr::str_replace() is the replacement string. In this case we are replacing the matched pattern with an empty string.

The example here is simply one example of very many possible transformations we become used to in cleaning our datasets. The aim in transforming the variable names is to make then easier to use and to understand, both for ourselves and for others.

References

———. 2019b. *Stringr: Simple, Consistent Wrappers for Common String Operations*. https://CRAN.R-project.org/package=stringr.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

10.28 Data Review

20180721 Having ingested the dataset and normalised the variable names we can now explore more. Using pillar::glimpse() gives us some insight:

```
# Review the dataset.

glimpse(ds)
```

```
## Rows: 176,747
## Columns: 24
## $ date            <date> 2008-12-01, 2008-12-02, 2008-12-03, 2008-12-04, 2008-…
## $ location        <chr> "Albury", "Albury", "Albury", "Albury", "Albury", "Alb…
## $ min_temp        <dbl> 13.4, 7.4, 12.9, 9.2, 17.5, 14.6, 14.3, 7.7, 9.7, 13.1…
## $ max_temp        <dbl> 22.9, 25.1, 25.7, 28.0, 32.3, 29.7, 25.0, 26.7, 31.9, …
## $ rainfall        <dbl> 0.6, 0.0, 0.0, 0.0, 1.0, 0.2, 0.0, 0.0, 0.0, 1.4, 0.0,…
## $ evaporation     <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA…
## $ sunshine        <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA…
## $ wind_gust_dir   <ord> W, WNW, WSW, NE, W, WNW, W, W, NNW, W, N, NNE, W, SW, …
## $ wind_gust_speed <dbl> 44, 44, 46, 24, 41, 56, 50, 35, 80, 28, 30, 31, 61, 44…
## $ wind_dir_9am    <ord> W, NNW, W, SE, ENE, W, SW, SSE, SE, S, SSE, NE, NNW, W…
## $ wind_dir_3pm    <ord> WNW, WSW, WSW, E, NW, W, W, W, NW, SSE, ESE, ENE, NNW,…
## $ wind_speed_9am  <dbl> 20, 4, 19, 11, 7, 19, 20, 6, 7, 15, 17, 15, 28, 24, 4,…
## $ wind_speed_3pm  <dbl> 24, 22, 26, 9, 20, 24, 24, 17, 28, 11, 6, 13, 28, 20, …
## $ humidity_9am    <int> 71, 44, 38, 45, 82, 55, 49, 48, 42, 58, 48, 89, 76, 65…
## $ humidity_3pm    <int> 22, 25, 30, 16, 33, 23, 19, 19, 9, 27, 22, 91, 93, 43,…
## $ pressure_9am    <dbl> 1007.7, 1010.6, 1007.6, 1017.6, 1010.8, 1009.2, 1009.6…
## $ pressure_3pm    <dbl> 1007.1, 1007.8, 1008.7, 1012.8, 1006.0, 1005.4, 1008.2…
## $ cloud_9am       <int> 8, NA, NA, NA, 7, NA, 1, NA, NA, NA, NA, 8, 8, NA, NA,…
## $ cloud_3pm       <int> NA, NA, 2, NA, 8, NA, NA, NA, NA, NA, NA, 8, 8, 7, NA,…
## $ temp_9am        <dbl> 16.9, 17.2, 21.0, 18.1, 17.8, 20.6, 18.1, 16.3, 18.3, …
## $ temp_3pm        <dbl> 21.8, 24.3, 23.2, 26.5, 29.7, 28.9, 24.6, 25.5, 30.2, …
## $ rain_today      <fct> No, No, No, No, No, No, No, No, No, Yes, No, Yes, Yes,…
## $ risk_mm         <dbl> 0.0, 0.0, 0.0, 1.0, 0.2, 0.0, 0.0, 0.0, 1.4, 0.0, 2.2,…
## $ rain_tomorrow   <fct> No, No, No, No, No, No, No, No, Yes, No, Yes, Yes, Yes…
```

Observe the variety of data types here, ranging from *Date* (`date)` , through *character* (`chr`) and *numeric* (`dbl`). The data mostly looks as expected though it is odd that `evaporation` and `sunshine` are identified as character. Probably because they seem to be all missing, at least in the first 10 or so observations. We begin question other aspects of the data too. For example, is `date` an ongoing sequence of days as it appears to be here? Does `location` have values other than `Albury` ? What is the distribution of the different variables?

These are all questions we will start asking ourselves in the context of living and breathing'' our data. Our aim should be to gleam all we can about the data that we are dealing with. Data science is very much about understanding, not blindly processing. The excitement is in the discovery of patterns in the data and the narrative the data is seeking to tell.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

10.29 Dataset Head and Tail

20180721 Datasets can be very large, with many observations (millions) and many variables (thousands). We can't be expected to browse through all of the observations and variables. Instead we might review the contents of the dataset using utils::head() and utils::tail() to consider the top six (by default) and the bottom six observations.

```
# Review the first few observations.

head(ds) %>% print.data.frame()
```

```
##         date location min_temp max_temp rainfall evaporation sunshine
## 1 2008-12-01   Albury     13.4     22.9      0.6          NA       NA
## 2 2008-12-02   Albury      7.4     25.1      0.0          NA       NA
## 3 2008-12-03   Albury     12.9     25.7      0.0          NA       NA
## 4 2008-12-04   Albury      9.2     28.0      0.0          NA       NA
## 5 2008-12-05   Albury     17.5     32.3      1.0          NA       NA
## 6 2008-12-06   Albury     14.6     29.7      0.2          NA       NA
##   wind_gust_dir wind_gust_speed wind_dir_9am wind_dir_3pm wind_speed_9am
## 1             W              44            W          WNW             20
## 2           WNW              44          NNW          WSW              4
## 3           WSW              46            W          WSW             19
## 4            NE              24           SE            E             11
## 5             W              41          ENE           NW              7
## 6           WNW              56            W            W             19
##   wind_speed_3pm humidity_9am humidity_3pm pressure_9am pressure_3pm cloud_9am
## 1             24           71           22       1007.7       1007.1         8
## 2             22           44           25       1010.6       1007.8        NA
## 3             26           38           30       1007.6       1008.7        NA
## 4              9           45           16       1017.6       1012.8        NA
## 5             20           82           33       1010.8       1006.0         7
## 6             24           55           23       1009.2       1005.4        NA
##   cloud_3pm temp_9am temp_3pm rain_today risk_mm rain_tomorrow
## 1        NA     16.9     21.8         No     0.0            No
## 2        NA     17.2     24.3         No     0.0            No
## 3         2     21.0     23.2         No     0.0            No
## 4        NA     18.1     26.5         No     1.0            No
## 5         8     17.8     29.7         No     0.2            No
## 6        NA     20.6     28.9         No     0.0            No
```

```r
# Review the last few observations.

tail(ds) %>% print.data.frame()
```

```
##         date location min_temp max_temp rainfall evaporation sunshine
## 1 2020-04-24    Uluru     23.0     36.7        0          NA       NA
## 2 2020-04-25    Uluru     18.4     37.4        0          NA       NA
## 3 2020-04-26    Uluru     21.4     32.7        0          NA       NA
## 4 2020-04-27    Uluru     19.4     32.2        0          NA       NA
## 5 2020-04-28    Uluru     16.6     32.6        0          NA       NA
## 6 2020-04-29    Uluru     16.7     25.7        0          NA       NA
##   wind_gust_dir wind_gust_speed wind_dir_9am wind_dir_3pm wind_speed_9am
## 1           ESE              31            E            E             22
## 2             W              35            E          NNW              9
## 3             S              54          ESE          ESE             20
## 4            SE              41            E           NE             28
## 5           SSW              44            S            W             19
## 6           SSW              54            S           SW             31
##   wind_speed_3pm humidity_9am humidity_3pm pressure_9am pressure_3pm cloud_9am
## 1              9           30           17       1017.3       1013.4        NA
## 2             15           27           14       1016.8       1012.3        NA
## 3             17           41           24       1017.7       1013.9         4
## 4             13           59           27       1018.7       1013.2         4
## 5             15           37           25       1015.3       1012.0        NA
## 6             26           69           28       1015.9       1014.4        NA
##   cloud_3pm temp_9am temp_3pm rain_today risk_mm rain_tomorrow
## 1         2     27.4     35.6         No       0            No
## 2        NA     26.4     36.4         No       0            No
## 3         4     24.5     32.0         No       0            No
## 4        NA     21.4     30.5         No       0            No
## 5        NA     24.0     31.3         No       0            No
## 6        NA     18.8     25.1         No       0            No
```

All the time we are building a picture of the data we are looking at. It is beginning to confirm that `location` has multiple values whilst `date` does appear to be a sequence for each location.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

10.30 Random Observations

20180721 It is also useful to review some random observations from the dataset to provide a little more insight. Here we use dplyr::sample_n() to randomly select six rows from the dataset.

```
# Review a random sample of observations.

sample_n(ds, size=6) %>% print.data.frame()
```

```
##         date      location min_temp max_temp rainfall evaporation sunshine
## 1 2011-12-30  PerthAirport     18.0     31.6      0.0         7.8     10.9
## 2 2010-05-09    Townsville     16.0     27.3      0.0         6.4      4.4
## 3 2017-06-29 NorfolkIsland     14.0     19.2      0.0         3.0       NA
## 4 2012-06-18        Cairns     16.5     26.5      0.0         6.2      9.5
## 5 2020-01-18 SydneyAirport     18.2     22.8     14.0         3.2      0.0
## 6 2015-02-14          Sale     18.5     23.0     15.6          NA       NA
##   wind_gust_dir wind_gust_speed wind_dir_9am wind_dir_3pm wind_speed_9am
## 1           WSW              35            N          WSW              9
## 2            SE              41          SSW           SE              9
## 3           SSE              30          SSE           SE             17
## 4            SE              44            S           SE             19
## 5           SSW              61          SSW          SSW             43
## 6             E              28           SE          ESE              9
##   wind_speed_3pm humidity_9am humidity_3pm pressure_9am pressure_3pm cloud_9am
## 1             20           64           53       1003.3       1003.1         5
## 2             22           47           65       1015.3       1011.9         2
## 3             13           57           54       1024.0       1022.1        NA
## 4             30           67           56       1020.1       1016.5         6
## 5             31           85           82       1009.1       1009.2         8
## 6             15           97           95       1019.5       1017.0         8
##   cloud_3pm temp_9am temp_3pm rain_today risk_mm rain_tomorrow
## 1         3     26.7     29.4         No     0.0            No
## 2         7     24.2     24.2         No     0.2            No
## 3        NA     17.1     17.5         No     0.0            No
## 4         3     22.5     25.4         No     0.0            No
## 5         8     21.5     21.7        Yes     3.6           Yes
## 6         7     18.9     19.7        Yes    35.2           Yes
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open*

source software since 1984.

Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

10.31 Characters

20180723 On ingesting the dataset into R we observe the variables identified (automatically) as having *character* base::class(). The expected values for such variables are strings of characters. We often call such variables *categoric* variables. Within R these are usually represented as a data type called *factor* and handled specially by many of the modelling algorithms.

We can observe some meta data for each of the character variables. Let's first identify the character variables.

```
# Identify the character variables by index.

ds %>%
  sapply(is.character) %>%
  which() %T>%
  print() ->
chari
```

```
## location
##        2
```

```
# Identify the chracter variables by name.

ds %>%
  names() %>%
  '['(chari) %T>%
  print() ->
charc
```

```
## [1] "location"
```

We will review each one of these in more detail so as to understand how we make use of them in our analyses. In particular we consider which of the variables might be handled as factors.

Where a *character* variable takes on a limited number of possible values we might convert the variable from *character* into *factor* (categoric}) so as to take advantage of special handling of factors in R.

In fact, we think of a *factor* as a variable that can only take on a specific number of known distinct values which we call the *levels* of the *factor*.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

10.32 Factors

20180908 For datasets that we load into R we will not always have examples of all possible levels of a factor. Consequently it is not always possible to automatically list all of the levels automatically. Be default the tidyverse ingests these variables as *character* so that we can take specific action to convert them to *factor* as required.

We first review the number of unique *levels* for each of the factors.

```
# Observe the unique levels.

ds[charc] %>% sapply(unique)
```

```
##       location
##  [1,] "Albury"
##  [2,] "BadgerysCreek"
##  [3,] "Cobar"
##  [4,] "CoffsHarbour"
##  [5,] "Moree"
##  [6,] "Newcastle"
##  [7,] "NorahHead"
##  [8,] "NorfolkIsland"
##  [9,] "Penrith"
## [10,] "Richmond"
## [11,] "Sydney"
## [12,] "SydneyAirport"
## [13,] "WaggaWagga"
## [14,] "Williamtown"
## [15,] "Wollongong"
## [16,] "Canberra"
## [17,] "Tuggeranong"
## [18,] "MountGinini"
## [19,] "Ballarat"
## [20,] "Bendigo"
## [21,] "Sale"
## [22,] "MelbourneAirport"
## [23,] "Melbourne"
## [24,] "Mildura"
## [25,] "Nhil"
## [26,] "Portland"
## [27,] "Watsonia"
## [28,] "Dartmoor"
## [29,] "Brisbane"
## [30,] "Cairns"
## [31,] "GoldCoast"
## [32,] "Townsville"
## [33,] "Adelaide"
## [34,] "MountGambier"
```

```
## [35,] "Nuriootpa"
## [36,] "Woomera"
## [37,] "Albany"
## [38,] "Witchcliffe"
## [39,] "PearceRAAF"
## [40,] "PerthAirport"
## [41,] "Perth"
## [42,] "SalmonGums"
## [43,] "Walpole"
## [44,] "Hobart"
## [45,] "Launceston"
## [46,] "AliceSprings"
## [47,] "Darwin"
## [48,] "Katherine"
## [49,] "Uluru"
```

If we decide to convert all of these variables from character into factor, then we can do so using base::factor().

```
# Convert all chracter variables to be factors.

ds[charc] %<>% map(factor)
```

We don't actually do so here instead considering each character variable in turn to decide how to handle it, especially that we might observe that evaporation and sunshine appear to be numeric.

A oneliner to do the conversion:

```
ds %<>% mutate_if(sapply(ds, is.character), as.factor)
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open*

source software since 1984.

Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

10.33 Location

20180723 From our review of the data so far we start to make some observations about the character variables. The first is `location`. We note that several locations were reported in the above exploration of the dataset. We can confirm the number of locations by counting the number of base::unique() values the variable has in the original dataset.

```
# How many locations are represented in the dataset.

ds$location %>%
  unique() %>%
  length()
```

```
## [1] 49
```

We may not know in general what other locations we will come across in related datasets and we already have quite a collection of 49 locations. We will retain this variable as a character data type.

Here is a list of locations and their frequencies in the dataset.

```
ds$location %>%
  table()
```

```
## .
##         Adelaide           Albany           Albury     AliceSprings
##             3832             3679             3680             3680
##    BadgerysCreek         Ballarat          Bendigo         Brisbane
##             3632             3680             3671             3833
##           Cairns         Canberra            Cobar     CoffsHarbour
##             3680             4076             3649             3649
##         Dartmoor           Darwin        GoldCoast           Hobart
##             3649             3833             3680             3833
##        Katherine       Launceston        Melbourne MelbourneAirport
##             2218             3680             3833             3649
##          Mildura            Moree     MountGambier      MountGinini
##             3649             3649             3679             3680
##        Newcastle             Nhil        NorahHead    NorfolkIsland
##             3680             2218             3644             3649
##        Nuriootpa       PearceRAAF          Penrith            Perth
##             3648             3648             3679             3832
##     PerthAirport         Portland         Richmond             Sale
##             3648             3649             3649             3649
##       SalmonGums           Sydney    SydneyAirport       Townsville
##             3602             3984             3649             3680
##      Tuggeranong            Uluru       WaggaWagga          Walpole
##             3679             2218             3649             3645
##         Watsonia      Williamtown      Witchcliffe       Wollongong
##             3649             3649             3648             3680
##          Woomera
##             3649
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

10.34 Evaporation and Sunshine

20180723 The next two character variables are: . It does seem odd that these would be character, expecting both to be numeric values. If we look at the top of the dataset we see they have missing values.

```
# Note the character remaining variables to be dealt with.

head(ds$evaporation)
```

```
## [1] NA NA NA NA NA NA
```

```
head(ds$sunshine)
```

```
## [1] NA NA NA NA NA NA
```

```
# Review other random values.

sample(ds$evaporation, 8)
```

```
## [1]  NA  NA 5.4  NA 2.8  NA  NA 3.4
```

```
sample(ds$sunshine, 8)
```

```
## [1]  2.1 11.7   NA  0.1 10.4   NA  3.6  0.5
```

The heuristic used to determine the data type when ingesting data only looks at a subset of all the data before it determines the data type. In this case the early observations are all missing and so default to character which is general enough to capture all potential values. We need to convert the variables to numeric.

```
# Identify the vairables to process.

cvars <- c("evaporation", "sunshine")

# Check the current class of the variables.

ds[cvars] %>% sapply(class)
```

```
## evaporation    sunshine
##   "numeric"   "numeric"
```

```
# Convert to numeric.

ds[cvars] %<>% sapply(as.numeric)

# Review some random values.

sample(ds$evaporation, 10)
```

```
##  [1] 2.6  NA 2.4  NA  NA 0.4  NA 1.0  NA 8.2
```

```
sample(ds$sunshine, 10)
```

```
##  [1] 9.7 0.6 7.9  NA  NA 7.6 9.4  NA 5.2 4.2
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

10.35 Wind Directions

20180723 The three wind direction variables () are also identified as *character*. We review the distribution of values here with dplyr::select() identifying any variable that tidyr::contains() the string `_dir` and then build a base::table() over those variables.

```
# Review the distribution of observations across levels.

ds %>%
  select(contains("_dir")) %>%
  sapply(table)
```

```
##     wind_gust_dir wind_dir_9am wind_dir_3pm
## N           10989        13978        10475
## NNE          7937         9782         8002
## NE           8715         9335        10092
## ENE          9965         9592         9605
## E           11071        11237        10123
## ESE          9055         9536        10290
## SE          11331        11398        12919
## SSE         10946        10954        11089
## S           11043        10519        11788
## SSW         10809         9272         9902
## SW          10793        10135        11166
## WSW         11136         8392        11700
## W           12122        10183        12411
## WNW         10045         9067        10846
## NW           9705        10488        10315
## NNW          7954         9468         9358
```

Observe all 16 compass directions are represented and it would make sense to convert this into a factor. Notice that the directions are in alphabetic order and conversion to factor will retain that. Instead we can construct an ordered factor to capture the compass order (from `N`, `NNE`, to `NW` and `NNW`). We note the ordering of the directions here.

```
# Levels of wind direction are ordered compas directions.

compass <- c("N", "NNE", "NE", "ENE",
             "E", "ESE", "SE", "SSE",
             "S", "SSW", "SW", "WSW",
             "W", "WNW", "NW", "NNW")
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

10.36 Ordered Factor

20180723 Given our knowledge that compass directions have an obvious order, we convert the direction variables into an ordered factor. We do so using *ordered*= `TRUE` with base::factor().

```
# Note the names of the wind direction variables.

ds %>%
  select(contains("_dir")) %>%
  names() %T>%
  print() ->
vnames
```

```
## [1] "wind_gust_dir" "wind_dir_9am"  "wind_dir_3pm"
```

```
# Convert these variables from character to factor.

ds[vnames] %<>%
  lapply(factor, levels=compass, ordered=TRUE) %>%
  data.frame() %>%
  as_tibble()
# Confirm they are now factors.

ds[vnames] %>% sapply(class)
```

```
##      wind_gust_dir wind_dir_9am wind_dir_3pm
## [1,] "ordered"     "ordered"    "ordered"
## [2,] "factor"      "factor"     "factor"
```

We can again obtain a distribution of the variables to confirm that all we have changed is the data type.

```
# Verify the distribution has not changed.

ds %>%
  select(contains("_dir")) %>%
  sapply(table)
```

```
##     wind_gust_dir wind_dir_9am wind_dir_3pm
## N           10989        13978        10475
## NNE          7937         9782         8002
## NE           8715         9335        10092
## ENE          9965         9592         9605
## E           11071        11237        10123
## ESE          9055         9536        10290
## SE          11331        11398        12919
## SSE         10946        10954        11089
## S           11043        10519        11788
## SSW         10809         9272         9902
## SW          10793        10135        11166
## WSW         11136         8392        11700
## W           12122        10183        12411
## WNW         10045         9067        10846
## NW           9705        10488        10315
## NNW          7954         9468         9358
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

10.37 Rain

20180723 The two remaining character variables are: . Their distributions are generated by dplyr::select()ing from the dataset those variables that start with `rain_` and then build a base::table() over those variables. We use base::sapply() to apply base::table() to the selected columns to count the frequency of the occurrence of each value of a variable within the dataset.

```
# Review the distribution of observations across levels.

ds %>%
  select(starts_with("rain_")) %>%
  sapply(table)
```

```
##     rain_today rain_tomorrow
## No      135371        135353
## Yes      37058         37077
```

Noting that `No` and `Yes` are the only values these two variables will take it makes sense to convert them both to factors. We will keep the ordering as alphabetic and so a simple call to base::factor() will to convert from character to factor.

```
# Note the names of the rain variables.

ds %>%
  select(starts_with("rain_")) %>%
  names() ->
vnames

# Confirm these are currently character variables.

ds[vnames] %>% sapply(class)
```

```
##    rain_today rain_tomorrow
##      "factor"      "factor"
```

```
# Convert these variables from character to factor.

ds[vnames] %<>%
  lapply(factor) %>%
  data.frame() %>%
  as_tibble()

# Confirm they are now factors.

ds[vnames] %>% sapply(class)
```

```
##    rain_today rain_tomorrow
##      "factor"      "factor"
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

10.38 Numeric

20180723 Summaries of numeric data are provided using base::summary(). In the following we identify the numeric variables and summarise each. In doing so, as a data scientist, we want to again observe any oddities and to explain them.

```
ds %>%
  sapply(is.numeric) %>%
  which() %>%
  names %T>%
  print() ->
numi
```

```
##  [1] "min_temp"        "max_temp"        "rainfall"        "evaporation"
##  [5] "sunshine"        "wind_gust_speed" "wind_speed_9am"  "wind_speed_3pm"
##  [9] "humidity_9am"    "humidity_3pm"    "pressure_9am"    "pressure_3pm"
## [13] "cloud_9am"       "cloud_3pm"       "temp_9am"        "temp_3pm"
## [17] "risk_mm"
```

```
ds[numi] %>%
  summary()
```

```
##     min_temp        max_temp        rainfall        evaporation
##  Min.   :-8.70   Min.   :-4.10   Min.   :  0.000   Min.   :  0.00
##  1st Qu.: 7.50   1st Qu.:18.10   1st Qu.:  0.000   1st Qu.:  2.80
##  Median :12.00   Median :22.80   Median :  0.000   Median :  4.80
##  Mean   :12.15   Mean   :23.36   Mean   :  2.241   Mean   :  5.53
##  3rd Qu.:16.90   3rd Qu.:28.40   3rd Qu.:  0.600   3rd Qu.:  7.40
##  Max.   :33.90   Max.   :48.90   Max.   :474.000   Max.   :133.90
##  NA's   :2349    NA's   :2105    NA's   :4318      NA's   :86289
##     sunshine     wind_gust_speed  wind_speed_9am  wind_speed_3pm
##  Min.   : 0.00   Min.   :  2.00   Min.   : 0.00   Min.   : 0.00
##  1st Qu.: 4.90   1st Qu.: 31.00   1st Qu.: 7.00   1st Qu.:13.00
##  Median : 8.50   Median : 39.00   Median :13.00   Median :19.00
##  Mean   : 7.66   Mean   : 40.19   Mean   :14.05   Mean   :18.72
##  3rd Qu.:10.60   3rd Qu.: 48.00   3rd Qu.:19.00   3rd Qu.:24.00
##  Max.   :14.50   Max.   :135.00   Max.   :87.00   Max.   :87.00
##  NA's   :93859   NA's   :13036    NA's   :2924    NA's   :5434
##   humidity_9am     humidity_3pm     pressure_9am     pressure_3pm
##  Min.   :  0.00   Min.   :  0.00   Min.   : 979.1   Min.   : 978.9
##  1st Qu.: 56.00   1st Qu.: 35.00   1st Qu.:1013.1   1st Qu.:1010.6
##  Median : 69.00   Median : 51.00   Median :1017.7   Median :1015.3
##  Mean   : 68.22   Mean   : 50.68   Mean   :1017.8   Mean   :1015.3
##  3rd Qu.: 83.00   3rd Qu.: 65.00   3rd Qu.:1022.5   3rd Qu.:1020.1
##  Max.   :100.00   Max.   :100.00   Max.   :1041.1   Max.   :1040.1
##  NA's   :3423     NA's   :6523     NA's   :19267    NA's   :19254
##    cloud_9am       cloud_3pm        temp_9am        temp_3pm
##  Min.   :0.00    Min.   :0.00    Min.   :-6.20   Min.   :-5.10
##  1st Qu.:1.00    1st Qu.:2.00    1st Qu.:12.30   1st Qu.:16.70
##  Median :5.00    Median :5.00    Median :16.80   Median :21.30
##  Mean   :4.52    Mean   :4.54    Mean   :17.02   Mean   :21.82
##  3rd Qu.:7.00    3rd Qu.:7.00    3rd Qu.:21.60   3rd Qu.:26.60
##  Max.   :9.00    Max.   :9.00    Max.   :40.20   Max.   :48.20
##  NA's   :73113   NA's   :78265   NA's   :2445    NA's   :5536
##     risk_mm
##  Min.   :  0.000
##  1st Qu.:  0.000
```

```
##  Median :  0.000
##  Mean   :  2.244
##  3rd Qu.:  0.600
##  Max.   :474.000
##  NA's   :4317
```

Reviewing this information we can make some obvious observations. Temperatures, for example, appears to be in degrees Celsius rather than Fahrenheit. Rainfall looks like millimetres. There are some quite skewed distributions with min and median small but large max values. As data scientists we will further explore the distributions as in Chapter 9.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

10.39 Logical

20180723 Above we converted `rain_today` and `rain_tomorrow` to factors. They have just two values as we confirm here, in addition to a small number of missing values (`NA`).

```
ds %>%
  select(rain_today, rain_tomorrow) %>%
  summary()
```

```
##  rain_today    rain_tomorrow
##  No  :135371   No  :135353
##  Yes : 37058   Yes : 37077
##  NA's:  4318   NA's:  4317
```

As binary valued factors, and particularly as the values suggest, they are both candidates for being considered as logical variables (sometimes called Boolean). They can be treated as FALSE/TRUE instead of No/Yes and so supported directly by R as class *logical*. Different functions will then treat them as appropriate but not all functions do anything special. If this suits our purposes then the following can be used to perform the conversion to logical.

Best to now check that the distribution itself has not changed.

Observe that the TRUE (`Yes`) values are much less frequent than the FALSE (`No`) values, and we also note the missing values.

The majority of days not having rain can be cross checked with the rainfall variable. In the previous summary of its distribution we note that rainfall has a median of zero, consistent with fewer days of actual rain. As data scientists we perform various cross checks on the hunt for oddities in the data.

As data scientists we will also want to understand why there are missing values. Is it simply some rare failures to capture the observation, or for example is there a particular location not recording rainfall? We would explore that now before moving on.

For our purposes going forward we will retain these two variables as factors. One reason for doing so is that we will illustrate missing value imputation using randomForest::na.roughfix() and this functoin does not handle logical data but keeping `rain_tomorrow` as character will allow missing value imputation. Of course we could skip this variable for the imputation.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

10.40 Variable Roles

20180723 Now that we have a basic idea of the size and shape and contents of the dataset and have performed some basic data type identification and conversion we are in a position to identify the roles played by the variables within the dataset. First we will record the list of available variables so that we might reference them below.

```
# Note the available variables.

vars <- names(ds) %T>% print()
```

```
##  [1] "date"            "location"        "min_temp"        "max_temp"
##  [5] "rainfall"        "evaporation"     "sunshine"        "wind_gust_dir"
##  [9] "wind_gust_speed" "wind_dir_9am"    "wind_dir_3pm"    "wind_speed_9am"
## [13] "wind_speed_3pm"  "humidity_9am"    "humidity_3pm"    "pressure_9am"
## [17] "pressure_3pm"    "cloud_9am"       "cloud_3pm"       "temp_9am"
## [21] "temp_3pm"        "rain_today"      "risk_mm"         "rain_tomorrow"
```

By this stage of the project we will usually have identified a business problem that is the focus of attention. In our case we will assume it is to build a predictive analytics model to predict the chance of it raining tomorrow given the observation of today's weather. In this case the variable `rain_tomorrow` is the *target variable*. Given today's observations of the weather this is what we want to predict. The dataset we have is then a *training dataset* of historic observations. The task is to identify any patterns among the other observed variables that suggest that it rains the following day.

```r
# Note the target variable.

target <- "rain_tomorrow"

# Place the target variable at the beginning of the vars.

vars <- c(target, vars) %>% unique() %T>% print()
```

```
##  [1] "rain_tomorrow"   "date"            "location"        "min_temp"
##  [5] "max_temp"        "rainfall"        "evaporation"     "sunshine"
##  [9] "wind_gust_dir"   "wind_gust_speed" "wind_dir_9am"    "wind_dir_3pm"
## [13] "wind_speed_9am"  "wind_speed_3pm"  "humidity_9am"    "humidity_3pm"
## [17] "pressure_9am"    "pressure_3pm"    "cloud_9am"       "cloud_3pm"
## [21] "temp_9am"        "temp_3pm"        "rain_today"      "risk_mm"
```

We have taken the opportunity here to move the target variable to be the first in the vector of variables recorded in `vars`. This is common practice where the first variable in a dataset is the target (dependent variable) and the remainder are the variables (the independent variables) that will be used to build a model to predict that target.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

10.41 Risk Variable

20180723 With some knowledge of the data we observe `risk_mm` captures the amount of rain recorded tomorrow. We refer to this as a *risk variable*, being a measure of the impact or risk of the target we are predicting (rain tomorrow). The risk is an output variable and should not be used as an input to the modelling—it is not an independent variable. In other circumstances it might actually be treated as the target variable.

```
# Note the risk variable - measures the severity of the outcome.

risk <- "risk_mm"
```

For this risk variable note that we expect it to have a value of 0 for all observations when the target variable has the value `No`.

```
# Review the distribution of the risk variable for non-targets.

ds %>%
  filter(rain_tomorrow == "No") %>%
  select(risk_mm) %>%
  summary()
```

```
##     risk_mm
##  Min.   :0.0000
##  1st Qu.:0.0000
##  Median :0.0000
##  Mean   :0.0726
##  3rd Qu.:0.0000
##  Max.   :1.0000
```

Interestingly, even a little rain (defined as 1mm or less) is regarded as no rain. That is useful to keep in mind and is a discovery of the data that we might not have expected. As data scientists we should be expecting to find the unexpected.

A similar analysis for the target observations is more in line with expectations.

```
# Review the distribution of the risk variable for targets.

ds %>%
  filter(rain_tomorrow == "Yes") %>%
  select(risk_mm) %>%
  summary()
```

```
##     risk_mm
##  Min.   :  1.10
##  1st Qu.:  2.40
##  Median :  5.00
##  Mean   : 10.17
##  3rd Qu.: 11.40
##  Max.   :474.00
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

10.42 ID Variables

20180723 From our observations so far we note that the variable (`date`) acts as an identifier as does the variable (`location`). Given a `date` and a `location` we have an observation of the remaining variables. Thus we note that these two variables are so-called identifiers. Identifiers would not usually be used as independent variables for building predictive analytics models.

```
# Note any identifiers.

id <- c("date", "location")
```

We might get a sense of how this works with the following which will list a random sample of locations and how long the observations for that location have been collected.

```
ds[id] %>%
  group_by(location) %>%
  count() %>%
  rename(days=n) %>%
  mutate(years=round(days/365)) %>%
  as.data.frame() %>%
  sample_n(10)
```

```
##         location days years
## 1    Williamtown 3649    10
## 2       Ballarat 3680    10
## 3   MountGambier 3679    10
## 4          Moree 3649    10
## 5      Newcastle 3680    10
## 6         Albury 3680    10
## 7           Sale 3649    10
## 8  NorfolkIsland 3649    10
## 9       Portland 3649    10
## 10         Uluru 2218     6
```

The data for each location ranges in length from 4 years up to 9 years, though most have 8 years of data.

```
ds[id] %>%
  group_by(location) %>%
  count() %>%
  rename(days=n) %>%
  mutate(years=round(days/365)) %>%
  ungroup() %>%
  select(years) %>%
  summary()
```

```
##      years
##  Min.   : 6.000
##  1st Qu.:10.000
##  Median :10.000
##  Mean   : 9.878
##  3rd Qu.:10.000
##  Max.   :11.000
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

10.43 Ignore IDs and Outputs

20180723 The identifiers and any risk variable (which is an output variable) should be ignored in any predictive modelling. Always watch out for treating output variables as inputs to modelling—this is a surprisingly common trap for beginners. We will build a vector of the names of the variables to ignore. Above we have already recorded the `id` variables and (optionally) the `risk`. Here we join them together into a new vector using base::union() which performs a set union operation—that is, it joins the two arguments together and removes any repeated variables.

```
# Initialise ignored variables: identifiers and risk.

ignore <- union(id, risk) %T>% print()
```

```
## [1] "date"     "location" "risk_mm"
```

We might also check for any variable that has a unique value for every observation. These are often identifiers and if so they are candidates for ignoring. We select the `vars` from the dataset and pipe through to base::sapply() for any variables having only unique values. In our case there are no further candidate identifiers. as indicated by the empty result, `character(0)`.

```
# Heuristic for candidate indentifiers to possibly ignore.

ds[vars] %>%
  sapply(function(x) x %>% unique() %>% length()) %>%
  equals(nrow(ds)) %>%
  which() %>%
  names() %T>%
  print() ->
ids
```

```
## character(0)
```

```r
# Add them to the variables to be ignored for modelling.

ignore <- union(ignore, ids) %T>% print()
```

```
## [1] "date"     "location" "risk_mm"
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

10.44 Ignore Missing

20180723 We next remove any variable where all of the values are missing. There are none like this in the weather dataset but in general for other datasets with thousands of variables there may be some. Here we first count the number of missing values for each variable and then list the names of those variables that have no values.

```
# Identify variables with only missing values.

ds[vars] %>%
  sapply(function(x) x %>% is.na %>% sum) %>%
  equals(nrow(ds)) %>%
  which() %>%
  names() %T>%
  print() ->
missing
```

```
## character(0)
```

```
# Add them to the variables to be ignored for modelling.

ignore <- union(ignore, missing) %T>% print()
```

```
## [1] "date"     "location" "risk_mm"
```

It is also useful to identify those variables which are very sparse—that have mostly missing values. We can decide on a threshold of the proportion missing above which to ignore the variable as not likely to add much value to our analysis. For example, we may want to ignore variables with more than 70% of the values missing:

```r
# Identify a threshold above which proportion missing is fatal.

missing.threshold <- 0.7

# Identify variables that are mostly missing.

ds[vars] %>%
  sapply(function(x) x %>% is.na() %>% sum()) %>%
  '>'(missing.threshold*nrow(ds)) %>%
  which() %>%
  names() %T>%
  print() ->
mostly
```

```
## character(0)
```

```r
# Add them to the variables to be ignored for modelling.

ignore <- union(ignore, mostly) %T>% print()
```

```
## [1] "date"     "location" "risk_mm"
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

10.45 Ignore Excessive Level Variables

20180723 Another issue we traditionally come across in our datasets are those factors with very many levels. This is more common when we read data as factors rather than as character, and so this step depends on where the data has come from. Nonetheless We might want to check for and ignore such variables.

```
# Identify a threshold above which we have too many levels.

levels.threshold <- 20

# Identify variables that have too many levels.

ds[vars] %>%
  sapply(is.factor) %>%
  which() %>%
  names() %>%
  sapply(function(x) ds %>% pull(x) %>% levels() %>% length()) %>%
  '>='(levels.threshold) %>%
  which() %>%
  names() %T>%
  print() ->
too.many
```

```
## character(0)
```

```
# Add them to the variables to be ignored for modelling.

ignore <- union(ignore, too.many) %T>% print()
```

```
## [1] "date"     "location" "risk_mm"
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

10.46 Dealing with Correlations

20180726 From the final result we can identify pairs of variables where we might want to keep one but not the other variable because they are highly correlated. We will select them manually since it is a judgement call. Normally we might limit the removals to those correlations that are 0.90 or more. In our case here the three pairs of highly correlated variables make intuitive sense.

```
# Note the correlated variables that are redundant.

correlated <- c("temp_3pm", "pressure_3pm", "temp_9am")

# Add them to the variables to be ignored for modelling.

ignore <- union(ignore, correlated) %T>% print()
```

```
## [1] "date"         "location"     "risk_mm"      "temp_3pm"     "pressure_3pm"
## [6] "temp_9am"
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

10.47 Removing Ignored Variables

20180726 Once we have identified all of the variables to ignore we remove them from our list of variables to use.

```
# Check the number of variables currently.

length(vars)
```

```
## [1] 24
```

```
# Remove the variables to ignore.

vars <- setdiff(vars, ignore)

# Confirm they are now ignored.

length(vars)
```

```
## [1] 18
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

10.48 Feature Selection

20180726 The FSelector (Romanski, Kotthoff, and Schratz 2021) package provides functions to identify subsets of variables that might be more effective for modelling. We can use this (and other packages) to assist us in reducing the variables that will be useful in our modelling. As we find useful functionality we will add them to our standard template so that for our next dataset we have the functionality readily available.

We first use FSelector::cfs() to identify a good subset of variables using correlation and entropy. We then list the variable importance using FSelector::information.gain() to advise a useful subset of variables. Note that the stringi::%s+% operator is a convenience to concatenate strings together to produce a formula that indicates we will model the target variable using all of the other variables of the dataset.

% Do we need options(scipen=10) here - Tony does?

```
# Construct the formulation of the modelling we plan to do.

form <- formula(target %s+% " ~ .") %T>% print()
```

```
## rain_tomorrow ~ .
```

```
# Use correlation search to identify key variables.

cfs(form, ds[vars])
```

```
## [1] "rainfall"     "sunshine"     "humidity_3pm" "cloud_3pm"    "rain_today"
```

```r
# Use information gain to identify variable importance.

information.gain(form, ds[vars])
```

```
##                 attr_importance
## min_temp            0.005574465
## max_temp            0.014306291
## rainfall            0.058146788
## evaporation         0.004816268
## sunshine            0.050838588
## wind_gust_dir       0.005941322
## wind_gust_speed     0.026772843
## wind_dir_9am        0.008740164
## wind_dir_3pm        0.004717540
## wind_speed_9am      0.004189375
## wind_speed_3pm      0.005426695
## humidity_9am        0.038093685
## humidity_3pm        0.108910707
## pressure_9am        0.027130527
## cloud_9am           0.032297858
## cloud_3pm           0.047524832
## rain_today          0.046474217
```

The two measures are consistent in this case in that the variables identified by FSelector::cfs() are the more important variables identified by FSelector::information.gain().

References

Romanski, Piotr, Lars Kotthoff, and Patrick Schratz. 2021. *FSelector: Selecting Attributes*. https://github.com/larskotthoff/fselector.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

10.49 Missing Targets

20180726 Sometimes there may be further operations to perform on the dataset prior to modelling. A common task is to deal with missing values. Here we remove observations with a missing target. As with any missing data we should also analyse whether there is any pattern to the missing targets. This may be indicative of a systemic data issue rather than simply randomly missing values.

```
# Check the dimensions to start with.

dim(ds)
```

```
## [1] 176747     24
```

```
# Identify observations with a missing target.

ds %>%
  pull(target) %>%
  is.na() ->
missing.target

# Check how many are found.

sum(missing.target)
```

```
## [1] 4317
```

```r
# Remove observations with a missing target.

ds %<>% filter(!missing.target)

# Confirm the filter delivered the expected dataset.

dim(ds)
```

```
## [1] 172430     24
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

10.50 Missing Values

20180726 Missing values for the variables are an issue for some but not all algorithms. For example randomForest::randomForest() omits observations with missing values by default whilst rpart::rpart() has a particularly well developed approach to dealing with missing values.

We may want to impute missing values in the data (though it is not always wise to do so). Here we do this using randomForest::na.roughfix() from randomForest (Breiman et al. 2018). This function provides, as the name implies, a rather basic algorithm for imputing missing values. Because of this we will demonstrate the process but then restore the original dataset—we will not want this imputation to be included in our actual dataset.

```
# Backup the dataset so we can restore it as required.

ods <- ds
```

```
# Count the number of missing values.

ds[vars] %>%  is.na() %>% sum()
```

```
## [1] 399165
```

```
# Impute missing values.

ds[vars] %<>% na.roughfix()

# Confirm that no missing values remain.

ds[vars] %>%  is.na() %>% sum()
```

```
## [1] 0
```

As foreshadowed we now restore the dataset with its original contents.

```
# Restore the original dataset.

ds <- ods
```

References

Breiman, Leo, Adele Cutler, Andy Liaw, and Matthew Wiener. 2018. *randomForest: Breiman and Cutler's Random Forests for Classification and Regression*. https://www.stat.berkeley.edu/~breiman/RandomForests/.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

10.51 Omitting Observations

20180726 An alternative is to remove observations that have missing values. Here data.table::na.omit() identifies the rows to omit based on the `vars` to be included for modelling. The list of rows to omit is stored as the `na.action` attribute of the returned object. We then remove these observations from the dataset.

Notice we keep a copy of the original dataset and then restore it.

```
# Backup the dataset so we can restore it as required.

ods <- ds

# Initialise the list of observations to be removed.

omit <- NULL
```

```
# Review the current dataset.

ds[vars] %>% nrow()
```

```
## [1] 172430
```

```
ds[vars] %>% is.na() %>% sum()
```

```
## [1] 399165
```

```
# Identify any observations with missing values.

mo <- attr(na.omit(ds[vars]), "na.action")

# Record the observations to omit.

omit <- union(omit, mo)

# If there are observations to omit then remove them.

if (length(omit)) ds <- ds[-omit,]

# Confirm the observations have been removed.

ds[vars] %>% nrow()
```

```
## [1] 61820
```

```
ds[vars] %>% is.na() %>% sum()
```

```
## [1] 0
```

```
# Restore the original dataset.

ds <- ods
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open*

source software since 1984.

Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

10.52 Normalise Factors

20180726 Some variables will have levels with spaces, and mixture of cases, etc. We may like to normalise the levels for each of the categoric variables. For very large datasets this can take some time and so we may want to be selective.

```
# Note which variables are categoric.

ds %>%
  sapply(is.factor) %>%
  which() ->
catc

# Normalise the levels of all categoric variables.

for (v in catc)
  levels(ds[[v]]) %<>% normVarNames()
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

10.53 Target as a Factor

20180726 We often build classification models. For such models we want to ensure the target is categoric. Often it is 0/1 and hence is loaded as numeric. We could tell our model algorithm of choice to explicitly do classification or else set the target using base::as.factor() in the formula. Nonetheless it is generally cleaner to do this here and note that this code has no effect if the target is already categoric.

```
# Ensure the target is categoric.

ds[[target]] %<>% as.factor()

# Confirm the distribution.

ds[target] %>% table()
```

```
## .
##     no    yes
## 135353  37077
```

We can visualise the distribution of the target variable using ggplot2 (Wickham et al. 2020). The dataset is piped to ggplot2::ggplot() whereby the target is associated through ggplot2::aes_string() (the aesthetics) with the x-axis of the plot. To this we add a graphics layer using ggplot2::geom_bar() to produce the bar chart, with bars having *width*= `0.2` and a *fill*= color of `"grey"`. The resulting plot can be seen in Figure @ref(fig:data:plot_target_distribution).

```
ds %>%
  ggplot(aes_string(x=target)) +
  geom_bar(width=0.2, fill="grey") +
  theme(text=element_text(size=14))
```

(#fig:data:plot_target_distribution)Target variable distribution. Plotting the distribution is useful to gain an insight into the number of observations in each category. As is the case here we often see a skewed distribution.

References

Wickham, Hadley, Winston Chang, Lionel Henry, Thomas Lin Pedersen, Kohske Takahashi, Claus Wilke, Kara Woo, Hiroaki Yutani, and Dewey Dunnington. 2020. *Ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics*. https://CRAN.R-project.org/package=ggplot2.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

10.54 Identify Variable Types

20180726 Metadata is data about the data. We now record data about our dataset that we use later in further processing and analysing our data. In one sense the metadata is simply a convenient store.

We identify the variables that will be used to build analytic models that provide different kinds of insight into our data. Above we identified the variable roles such as the target, a risk variable and the ignored variables. From an analytic modelling perspective we identify variables that are the model inputs. We record then both as a vector of characters (the variable names) and a vector of integers (the variable indicies).

```
inputs <- setdiff(vars, target) %T>% print()
```

```
##  [1] "min_temp"        "max_temp"        "rainfall"        "evaporation"
##  [5] "sunshine"        "wind_gust_dir"   "wind_gust_speed" "wind_dir_9am"
##  [9] "wind_dir_3pm"    "wind_speed_9am"  "wind_speed_3pm"  "humidity_9am"
## [13] "humidity_3pm"    "pressure_9am"    "cloud_9am"       "cloud_3pm"
## [17] "rain_today"
```

The integer indices are determined from the base::names() of the variables in the original dataset. Note the use of *USE.NAMES*= from base::sapply() to turn off the inclusion of names in the resulting vector to keep the result as a simple vector.

```
inputi <- sapply(inputs,
                 function(x) which(x == names(ds)),
                 USE.NAMES=FALSE)
inputi
```

```
##  [1]  3  4  5  6  7  8  9 10 11 12 13 14 15 16 18 19 22
```

For convenience we record the number of observations:

```
nobs <- nrow(ds) %T>% print()
```

```
## [1] 172430
```

Here we simply report on the dimensions of various data subsets primarily to confirm the dataset appear as we expect:

```
dim(ds)
```

```
## [1] 172430     24
```

```
dim(ds[vars])
```

```
## [1] 172430     18
```

```
dim(ds[inputs])
```

```
## [1] 172430     17
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

10.55 Identify Numeric and Categoric Variables

20180726 Identifying numeric and categoric variables may be useful for example for cluster analysis algorithms that only deal with numeric variables. Here we identify them by name (a character string) and by index. When using the index we have to assume the variables remain in the same order within the dataset and all variables are present, otherwise the indicies will get out of sync.

```
# Identify the numeric variables by index.

ds %>%
  sapply(is.numeric) %>%
  which() %>%
  intersect(inputi) %T>%
  print() ->
numi
```

```
##  [1]  3  4  5  6  7  9 12 13 14 15 16 18 19
```

```
# Identify the numeric variables by name.

ds %>%
  names() %>%
  extract(numi) %T>%
  print() ->
numc
```

```
##  [1] "min_temp"        "max_temp"        "rainfall"        "evaporation"
##  [5] "sunshine"        "wind_gust_speed" "wind_speed_9am"  "wind_speed_3pm"
##  [9] "humidity_9am"    "humidity_3pm"    "pressure_9am"    "cloud_9am"
## [13] "cloud_3pm"
```

```
# Identify the categoric variables by index and then name.

ds %>%
  sapply(is.factor) %>%
  which() %>%
  intersect(inputi) %T>%
  print() ->
cati
```

```
## [1]  8 10 11 22
```

```
ds %>%
  names() %>%
  extract(cati) %T>%
  print() ->
catc
```

```
## [1] "wind_gust_dir" "wind_dir_9am"  "wind_dir_3pm"  "rain_today"
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

10.56 Save the Dataset

For large datasets we may want to save it to a binary RData file once we have wrangled it into the right shape and collected the metadata. Loading a binary dataset is generally quicker than loading a CSV file—a CSV file with 2 million observations and 800 variables can take 30 minutes to utils::read.csv(), 5 minutes to base::save(), and 30 seconds to base::load().

```
# Timestamp for the dataset.

dsdate  <- "_" %s+% format(Sys.Date(), "%y%m%d") %T>% print()
```

```
## [1] "_210624"
```

```
# Filename for the saved dataset

dsrdata <- dsname %s+% dsdate %s+% ".RData" %T>% print()
```

```
## [1] "weatherAUS_210624.RData"
```

```
# Save relevant R objects to binary RData file.

save(ds, dsname, dspath, dsdate, nobs,
     vars, target, risk, id, ignore, omit,
     inputi, inputs, numi, numc, cati, catc,
     file=dsrdata)
```

Notice that in addition to the dataset (`ds`) we also store the collection of *metadata*. This begins with items such as the name of the dataset, the source file path, the date we obtained the dataset, the number of observations, the variables of interest, the target variable, the name of the

risk variable (if any), the identifiers, the variables to ignore and observations to omit. We continue with the indicies of the input variables and their names, the indicies of the numeric variables and their names, and the indicies of the categoric variables and their names.

Each time we wish to use the dataset we can now simply base::load() it into R. The value that is invisibly returned by base::load() is a vector naming the R objects loaded from the binary RData file.

```
load(dsrdata) %>% print()
```

```
##  [1] "ds"     "dsname" "dspath" "dsdate" "nobs"   "vars"   "target" "risk"
##  [9] "id"     "ignore" "omit"   "inputi" "inputs" "numi"   "numc"   "cati"
## [17] "catc"
```

We place the call to base::load() within a call to [(](https://www.rdocumentation.org/packages/base/topics/){ target="_blank" } (i.e., we have surrounded the call with round brackets) to ensure the result of the function call is printed. A call to base::load() returns its result invisibly since we are primarily interested in its side-effect. The side-effect is to read to R binary data from disk and to make it available within our current R session.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

10.57 A Template for Data Preparation

Through this chapter we have built a template for data preparation. An actual knitr template based on this chapter for data preparation is available as http://HandsOnDataScience.com/scripts/data.Rnw. An automatically derived version including just the R code is also available as http://HandsOnDataScience.com/scripts/data.R. Notice that we would not necessarily perform all of the steps, such as normalising the variable names, imputing missing values, omitting observations with missing values, and so on. Instead we pick and choose as is appropriate to our situation and specific datasets. Also, some data specific transformations are not included in the template and there may be other transforms we need to perform that we have not covered here. As we discover new tools to support the data scientist we can add them into our own templates.

```
dsname        <- "weatherAUS"
ds            <- get(dsname)
vnames        <- names(ds)
names(ds)    %<>% normVarNames()
names(vnames) <- names(ds)
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

11 Data Visualisation

20200608 One of the most important tasks for any data scientist is to visualise data. Presenting data visually will often lead to new insights and discoveries, as well as providing glimpses of any issues with the data itself. Sometimes these will stand out in a visual presentation whilst being hidden in textual views. A visual presentation is also an effective means for communicating insight to others.

The simplest of plots is a Scatter Plot which plots points onto the graph. A Line Graph joins the points. A Bar Chart plots bars up to the points. A Box Plot presents the distribution of values of an observation. A Pie Chart is not usually recommended. Faceted plots consist of multiple graphs of the same data but across different factors. Plots can be coloured or annotated with text. Indeed, we are free to innovate with visualisations, though we need to be aware of principles for effectively communicating.

R is capable of producing excellent publication ready graphics in many formats, including PostScript, PDF, PNG (for vector images and text), and JPG (for colorful images). R offers a comprehensive suite of tools to visualise data including graphics (**R-graphics?**) (traditional), grid (**R-grid?**) and lattice (Sarkar 2021) (high level graphics), and now ggplot2 (Wickham et al. 2020), the newest plotting development.

ggplot2 (Wickham et al. 2020) is based on the idea of a grammar for graphics, a grammar for writing sentences describing the graphics. The default themes for producing plots is based on the collective wisdom of many visual presentation researchers. Using this package we construct a plot beginning with the dataset along with the aesthetics (e.g., the x-axis and y-axis) and then add geometric elements, statistical operations, scales, facets, coordinates, and numerous other components to the plot.

The official GGPlot2 documentation is extensive and accessible.

Exercises:

References

Sarkar, Deepayan. 2021. *Lattice: Trellis Graphics for r*. http://lattice.r-forge.r-project.org/.

Wickham, Hadley, Winston Chang, Lionel Henry, Thomas Lin Pedersen, Kohske Takahashi, Claus Wilke, Kara Woo, Hiroaki Yutani, and Dewey Dunnington. 2020. *Ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics*. https://CRAN.R-project.org/package=ggplot2.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

11.1 Visualisation Setup

20200608 Packages used in this chapter include GGally (Schloerke et al. 2021), RColorBrewer (Neuwirth 2014), colorRamps (Keitt 2012), dplyr (Wickham et al. 2021), epitools (Aragon 2020), ggplot2 (Wickham et al. 2020), lubridate (Spinu, Grolemund, and Wickham 2021), randomForest (Breiman et al. 2018), scales (Wickham and Seidel 2020), stringr (Wickham 2019b), tidyr (Wickham 2021), magrittr (Bache and Wickham 2020), and rattle (G. Williams 2020).

Packages are loaded into the currently running R session from your local library directories on disk. Missing packages can be installed using utils::install.packages() within R. On Ubuntu, for example, R packages can also be installed using `$ wajig install r-cran-<pkgname>`.

```
# Load required packages from local library into the R session.

library(GGally)       # Pairs plots.
library(RColorBrewer) # Brew various colour ranges.
library(colorRamps)   # Generate colour ranges: blue2green2red().
library(dplyr)        # glimpse().
library(epitools)     # Colour selection: colors.plot().
library(ggplot2)      # Visualise data.
library(lubridate)    # Tidy dates and time.
library(scales)       # commas(), percent().
library(stringr)      # str_replace_all().
library(tidyr)        # Tidy the dataset: pivot_longer().
library(rattle)       # Dataset weatherAUS.
```

References

Aragon, Tomas J. 2020. *Epitools: Epidemiology Tools*. https://CRAN.R-project.org/package=epitools.

Bache, Stefan Milton, and Hadley Wickham. 2020. *Magrittr: A Forward-Pipe Operator for r*. https://CRAN.R-project.org/package=magrittr.

Breiman, Leo, Adele Cutler, Andy Liaw, and Matthew Wiener. 2018. *randomForest: Breiman and Cutler's Random Forests for Classification and Regression*. https://www.stat.berkeley.edu/~breiman/RandomForests/.

Keitt, Tim. 2012. *colorRamps: Builds Color Tables*. https://CRAN.R-project.org/package=colorRamps.

Neuwirth, Erich. 2014. *RColorBrewer: ColorBrewer Palettes*. https://CRAN.R-project.org/package=RColorBrewer.

Schloerke, Barret, Di Cook, Joseph Larmarange, Francois Briatte, Moritz Marbach, Edwin Thoen, Amos Elberg, and Jason Crowley. 2021. *GGally: Extension to Ggplot2*. https://CRAN.R-project.org/package=GGally.

Spinu, Vitalie, Garrett Grolemund, and Hadley Wickham. 2021. *Lubridate: Make Dealing with Dates a Little Easier*. https://CRAN.R-project.org/package=lubridate.

———. 2019b. *Stringr: Simple, Consistent Wrappers for Common String Operations*. https://CRAN.R-project.org/package=stringr.

———. 2021. *Tidyr: Tidy Messy Data*. https://CRAN.R-project.org/package=tidyr.

Wickham, Hadley, Winston Chang, Lionel Henry, Thomas Lin Pedersen, Kohske Takahashi, Claus Wilke, Kara Woo, Hiroaki Yutani, and Dewey Dunnington. 2020. *Ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics*. https://CRAN.R-project.org/package=ggplot2.

Wickham, Hadley, Romain François, Lionel Henry, and Kirill Müller. 2021. *Dplyr: A Grammar of Data Manipulation*. https://CRAN.R-project.org/package=dplyr.

Wickham, Hadley, and Dana Seidel. 2020. *Scales: Scale Functions for Visualization*. https://CRAN.R-project.org/package=scales.

Williams, Graham. 2020. *Rattle: Graphical User Interface for Data Science in r*. https://rattle.togaware.com/.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*

Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

11.2 Visualisation Data

The rattle::weatherAUS dataset is loaded into the template variable `ds` and further template variables are setup as introduced by Graham J. Williams (2017). See Chapter 8 for details.

```
dsname <- "weatherAUS"
ds     <- get(dsname)

nobs   <- nrow(ds)

vnames <- names(ds)
ds    %<>% clean_names(numerals="right")
names(vnames) <- names(ds)

vars   <- names(ds)
target <- "rain_tomorrow"
vars   <- c(target, vars) %>% unique() %>% rev()
```

We also do a little more to set the data up for demonstrating various approaches to visualisation. As with the model template, a number of template variables are identified here. We also a little data wrangling to remove all missing values by performing a missing value imputation with randomForest::na.roughfix().

```
risk   <- "risk_mm"
id     <- c("date", "location")
ignore <- c(risk, id)
vars   <- setdiff(vars, ignore)
inputs <- setdiff(vars, target)
ds[vars] %<>% na.roughfix()
```

References

Williams, Graham J. 2017. *The Essentials of Data Science: Knowledge Discovery Using r*. The r Series. CRC Press.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

11.3 Visualisation Data Review

20200823 It is always good to review the dataset.

```
glimpse(ds)
```

```
## Rows: 176,747
## Columns: 24
## $ date            <date> 2008-12-01, 2008-12-02, 2008-12-03, 2008-12-04, 2008-…
## $ location        <chr> "Albury", "Albury", "Albury", "Albury", "Albury", "Alb…
## $ min_temp        <dbl> 13.4, 7.4, 12.9, 9.2, 17.5, 14.6, 14.3, 7.7, 9.7, 13.1…
## $ max_temp        <dbl> 22.9, 25.1, 25.7, 28.0, 32.3, 29.7, 25.0, 26.7, 31.9, …
## $ rainfall        <dbl> 0.6, 0.0, 0.0, 0.0, 1.0, 0.2, 0.0, 0.0, 0.0, 1.4, 0.0,…
## $ evaporation     <dbl> 4.8, 4.8, 4.8, 4.8, 4.8, 4.8, 4.8, 4.8, 4.8, 4.8, 4.8,…
## $ sunshine        <dbl> 8.5, 8.5, 8.5, 8.5, 8.5, 8.5, 8.5, 8.5, 8.5, 8.5, 8.5,…
## $ wind_gust_dir   <ord> W, WNW, WSW, NE, W, WNW, W, W, NNW, W, N, NNE, W, SW, …
## $ wind_gust_speed <dbl> 44, 44, 46, 24, 41, 56, 50, 35, 80, 28, 30, 31, 61, 44…
## $ wind_dir_9am    <ord> W, NNW, W, SE, ENE, W, SW, SSE, SE, S, SSE, NE, NNW, W…
## $ wind_dir_3pm    <ord> WNW, WSW, WSW, E, NW, W, W, W, NW, SSE, ESE, ENE, NNW,…
## $ wind_speed_9am  <dbl> 20, 4, 19, 11, 7, 19, 20, 6, 7, 15, 17, 15, 28, 24, 4,…
## $ wind_speed_3pm  <dbl> 24, 22, 26, 9, 20, 24, 24, 17, 28, 11, 6, 13, 28, 20, …
## $ humidity_9am    <dbl> 71, 44, 38, 45, 82, 55, 49, 48, 42, 58, 48, 89, 76, 65…
## $ humidity_3pm    <dbl> 22, 25, 30, 16, 33, 23, 19, 19, 9, 27, 22, 91, 93, 43,…
## $ pressure_9am    <dbl> 1007.7, 1010.6, 1007.6, 1017.6, 1010.8, 1009.2, 1009.6…
## $ pressure_3pm    <dbl> 1007.1, 1007.8, 1008.7, 1012.8, 1006.0, 1005.4, 1008.2…
## $ cloud_9am       <dbl> 8, 5, 5, 5, 7, 5, 1, 5, 5, 5, 5, 8, 8, 5, 5, 0, 8, 8, …
## $ cloud_3pm       <dbl> 5, 5, 2, 5, 8, 5, 5, 5, 5, 5, 5, 8, 8, 7, 5, 5, 1, 1, …
## $ temp_9am        <dbl> 16.9, 17.2, 21.0, 18.1, 17.8, 20.6, 18.1, 16.3, 18.3, …
## $ temp_3pm        <dbl> 21.8, 24.3, 23.2, 26.5, 29.7, 28.9, 24.6, 25.5, 30.2, …
## $ rain_today      <fct> No, No, No, No, No, No, No, No, No, Yes, No, Yes, Yes,…
## $ risk_mm         <dbl> 0.0, 0.0, 0.0, 1.0, 0.2, 0.0, 0.0, 0.0, 1.4, 0.0, 2.2,…
## $ rain_tomorrow   <fct> No, No, No, No, No, No, No, No, Yes, No, Yes, Yes, Yes…
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

11.4 Bar Chart Basic

20200427

```
ds %>%
  ggplot(aes(x=wind_dir_3pm)) +
  geom_bar() +
  scale_y_continuous(labels=comma) +
  labs(x=vnames["wind_dir_3pm"], y="Count")
```

A common and simple plot is the bar chart which displays bars with a height that corresponds to the number of observations having that value of the variable displayed on the x-axis. We use ggplot2::geom_bar() to add a bar chart layer to a plot. Only an x-axis aesthetic is required using ggplot2::aes() with the `x=` option. The y-axis is automatically computed.

In our example we pipe the dataset on to ggplot2::ggplot(), specifying the x-axis as the categoric variable `wind_dir_3pm` (the x-axis). The count is automatically determined from the dataset by ggplot2::geom_bar() which then adds the layer of bars to the plot.

The y-ticks use commas with the ggplot2::scale_continuous() function using the `labels=``comma` option. It is crucial that for large numbers commas separate the thousands, so that the reader is able to easily read the number. There can be catastrophic outcomes from a

misreading of numbers.

The x-axis and y-axis labels are set using ggplot2::labs() with the `x=` and `y=` options. The x-label uses the original dataset's variable name as recorded in the template variable `vname` (i.e., WindDir3pm). The y-label is set to be `Count` .

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

11.5 Bar Chart Colour No Legend

20200428

```
ds %>%
  ggplot(aes(x=wind_dir_3pm, fill=wind_dir_3pm)) +
  geom_bar() +
  scale_y_continuous(labels=comma) +
  labs(x=vnames["wind_dir_3pm"], y="Count") +
  theme(legend.position="none")
```

Colour can add interest to a plot, though (unlike this example) it is usually good to use the colour to communicate further information about the data. Nonetheless, it adds to the attractiveness of the visualisation.

A spread of colour is added through the aesthetic *fill*= `wind_dir_3pm`. This will fill the bars with colour controlled by the different values of the variable `wind_dir_3pm`. As this is also the variable being plotted on the x-axis a different colour is chosen for each bar from a good default set of colours.

We add a ggplot2::theme() element here to remove the legend that would be displayed by default, by indicating that the *legend.position*= `"none"`. An alternative to turn the legend off is to provide *show.legend*= `FALSE` to the layer (i.e., ggplot2::geom_bar()). This then allows layer specific control of the legends.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

11.6 Bar Chart Dodge

20210413

```
ds %>%
  ggplot(aes(x=wind_dir_3pm, fill=rain_tomorrow)) +
  geom_bar(position="dodge") +
  scale_y_continuous(labels=comma) +
  labs(x=vnames["wind_dir_3pm"], y="Count", fill="Rain") +
  scale_fill_brewer(palette="Set1")
```

A stacked bar chart is produced when a `fill` is provide through the aesthetics, as in Section 11.17. However the top bars can be difficult to compare amongst the columns. Instead of stacking the bars we can dodge the bars with the `position="dodge"` option to ggplot2::geom_bar().

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

11.7 Bar Chart Dodge Labelled Colour Brewer

20210413

```
ds %>%
  ggplot(aes(x=wind_dir_3pm, fill=rain_tomorrow)) +
  geom_bar(position="dodge") +
  geom_text(stat="count", aes(label=comma(..count..)),
            position=position_dodge(width=0.9), vjust=-0.2, size=2) +
  scale_y_continuous(labels=comma, limits=c(0, 18500)) +
  labs(x=vnames["wind_dir_3pm"], y="Count", fill="Rains Tomorrow?") +
  theme(legend.position="top") +
  scale_fill_brewer(palette="Set2")
```

A dodged bar chart is produced when a `fill=` option is provided for the ggplot2::aes() and `position="dodge"` option for the ggplot2::geom_bar(). We can add numeric labels using ggplot2::geom_label() (as in Section 11.14) or using ggplot2::geom_text() (as in Section 11.19). Here we demonstrate with ggplot2::geom_text().

Notice the use of scales::comma() for the labels as should be standard practice.

To tune the plot we've moved the legend to the top with the `legend.position="top"` option of ggplot2::theme(), increased the y-axis to fit the label using the `limits=` option of the ggplot2::scale_continuous() family of functions.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

11.8 Bar Chart Faceted Background

20200823

```
ds %>%
  sample_frac(0.1) %T>%
  {assign("norain", . %>% select(-rain_tomorrow), 1)} %>%
  mutate(rain_tomorrow=factor(rain_tomorrow,
                              levels=(rain_tomorrow %>% unique() %>%
                                        sort() %>% rev()))) %>%
  ggplot(aes(x=wind_dir_3pm, fill=wind_dir_3pm)) +
  geom_bar(data=norain, fill="gray", alpha=0.5) +
  geom_bar() +
  facet_wrap(~ rain_tomorrow) +
  labs(x=vnames["wind_dir_3pm"], y="Count") +
  scale_y_continuous(labels=comma) +
  theme(legend.position="none")
```

Using a faceted bar chart we can compare the individual charts with the global chart shown as a gray background.

Take a 10% sample. Remove the target column and save. Swap the order of No/Yes in the target column. Plot the wind direction. Add a grey layer with the non-target variable dataset. Add normal bar layer. Facet. Labels will have commas. Turn the legend off.

See https://drsimonj.svbtle.com/plotting-background-data-for-groups-with-ggplot2.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

11.9 Bar Chart Flipped Colour Mean no Legend

20200608

```
ds %>%
  filter(location %in% (ds$location %>% unique %>% sample(20))) %>%
  ggplot(aes(location, temp_3pm, fill=location)) +
  stat_summary(fun="mean", geom="bar") +
  theme(legend.position="none") +
  labs(x=vnames["temp_3pm"], y=vnames["location"]) +
  coord_flip()
```

Use ggplot2::coord_flip() to flip the coordinates and produce a horizontal histogram. Rotating the plot is generally useful when there are a large number of factor levels to fit on the x-axis. It is also easier to read the labels left to right rather than bottom up as we would need to do for the original plot.

For clarity of presentation here we reduce by using a dplyr::filter() the number of locations, selecting a random base::sample() of the base::unique() locations.

Colour is added through introducing a `fill=` option to the ggplot2::aes(). The bars in the chart generated using ggplot2::stat_summary() report on the base::mean() of the temperatures as recorded at 3pm daily for each location. The legend is removed through the ggplot2::theme().

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

11.10 Bar Chart Flipped Colour Mean Confidence Intervals

20200822

```
ds %>%
  filter(location %in% (ds$location %>% unique %>% sample(20))) %>%
  mutate(location=factor(location,
                         levels=location %>% unique() %>%
                                   sort() %>% rev())) %>%
  ggplot(aes(location, temp_3pm, fill=location)) +
  stat_summary(fun="mean", geom="bar") +
  stat_summary(fun.data="mean_cl_normal", geom="errorbar", width=0.35) +
  theme(legend.position="none") +
  labs(x=vnames["temp_3pm"], y=vnames["location"]) +
  coord_flip()
```

Various annotations can be added to plots. In this example we include a confidence interval around the average values.

Exercise: review the confidence intervals—do they make sense?

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

11.11 Bar Chart Flipped Sorted Axes

20200820

```
ds %>%
  filter(location %in% (ds$location %>% unique %>% sample(20))) %>%
  mutate(location=factor(location,
                         levels=(location %>% unique() %>%
                                   sort() %>% rev()))) %>%
  ggplot(aes(location, temp_3pm, fill=location)) +
  stat_summary(fun="mean", geom="bar") +
  theme(legend.position="none") +
  labs(x=vnames["temp_3pm"], y=vnames["location"]) +
  coord_flip()
```

Labels will sometimes appear in the reverse order to that required, particularly in this flipped bar chart. We can explicitly reorder the levels to ensure the plot labels are in a more natural (alphabetic) order for the human reader. We use dplyr::mutate() within a pipeline to create a factor

with the levels in the desired order passing that on to ggplot2::ggplot().

The embedded pipeline within the dplyr::mutate() places the base::unique() locations in base::rev()erse base::sort()ed alphabetic order.

The functions ggplot2::scale_y_reverse() ggplot2::scale_x_reverse() may also be useful.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

11.12 Bar Chart Flipped Text Annotations

20200823

```
ds %>%
  filter(location %in% (ds$location %>% unique %>% sample(20))) %>%
  mutate(location=factor(location,
                         levels=(location %>% unique() %>%
                                   sort() %>% rev()))) %>%
  ggplot(aes(location, fill=location)) +
  geom_bar(width=1, colour="white") +
  theme(legend.position="none") +
  labs(x=vnames["temp_3pm"], y=vnames["location"]) +
  coord_flip() +
  geom_text(stat="count", color="white", hjust=1.0, size=3,
            aes(y=..count.., label=..count..))
```

It can be informative to show actual numeric values on a plot. This plot shows the counts.

Exercise: Instead of plotting the counts, plot the mean `temp_3pm`, and include the textual value.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

11.13 Bar Chart Flipped Text Annotations Commas

20200823

```
ds %>%
  filter(location %in% (ds$location %>% unique %>% sample(20))) %>%
  mutate(location=factor(location,
                         levels=(location %>% unique() %>%
                                   sort() %>% rev()))) %>%
  ggplot(aes(location, fill=location)) +
  geom_bar(width=1, colour="white") +
  theme(legend.position="none") +
  coord_flip() +
  geom_text(stat="count", color="white", hjust=1.0, size=3,
            aes(y=as.integer(..count..),
                label=scales::comma(..count.., accuracy=1)))
```

Using scales::comma() for large numbers is essential in any plot to avoid misreading the magnitude.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

11.14 Bar Chart Labels

20210413

```
ds %>%
  ggplot(aes(x=wind_dir_3pm)) +
  geom_bar() +
  geom_label(stat="count", aes(label=comma(..count..)), vjust=-0.2, size=3) +
  scale_y_continuous(labels=comma, limits=c(0, 23000)) +
  labs(x=vnames["wind_dir_3pm"], y="Count")
```

A ggplot2::geom_label() can be used to add highlighted labels to a chart. Here for example they are added to a bar chart. Also see Section 11.19 to add just text and and **??** to add labels to a bar chart with dodged bars.

Notice the use of scales::comma() for the labels as should be standard practice.

To tune the plot we've increased the y-axis to fit the label using the `limits=` option of the ggplot2::scale_continuous() function.

Exercise: *further tuning to improve the presentation.*

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

11.15 Bar Chart Narrow Bars Economist Theme

20200428

```
ds %>%
  ggplot(aes(wind_dir_9am)) +
  geom_bar(width=0.5) +
  scale_y_continuous(labels=comma) +
  labs(x="", title="Wind Direction Recorded 9am Daily", y="Count") +
  ggthemes::theme_economist()
```

The width of the bars can be specified, relative to the space allowed for each bar. Here we make the bars narrower using `width=0.5` so as to be half width. This may help to make the plot look less dark.

Notice the use of the ggthemes::theme_economist() for a bit of variety. To suit this theme the usual x label is moved to be the title.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

11.16 Bar Chart Ordered X Axis

20210318

```
ds %>%
  count(wind_dir_9am) %>%
  ggplot(aes(x=reorder(wind_dir_9am, n), y=n)) +
  geom_col() +
  scale_y_continuous(labels=comma) +
  labs(x=vnames["wind_dir_9am"], y="Count")
```

In Section 11.18 a bar chart is shown with the x axis ordered by the levels of a factor. Often we want to order the bars increasing (or decreasing) according the bar heights. In this example we simply add stats::reorder() within the aesthetics to order by the count (`n` in this case).

To order the bars by decreasing heights replace `n` with `-n`.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

11.17 Bar Chart Stacked

20210413

```
ds %>%
  ggplot(aes(x=wind_dir_3pm, fill=rain_tomorrow)) +
  geom_bar() +
  scale_y_continuous(labels=comma) +
  labs(x=vnames["wind_dir_3pm"], y="Count") +
  scale_fill_brewer(palette="Set2")
```

A stacked bar chart shows a breakdown of the values in the bars based on another variable. It is achieved by adding `fill=rain_tomorrow` to the basic bar chart of Section 11.4.

An issue with stacked bar charts is that whilst the bottom bars are easy to compare bar by bar, the tops are difficult to compare since they all start from a different baseline. A good alternative is to have both bars in a column but side-by-side, as in Section 11.6.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of*

Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

11.18 Bar Chart Supplied Values

20200427

```
ds %>%
  count(wind_dir_9am) %>%
  ggplot(aes(x=wind_dir_9am, y=n)) +
  geom_col() +
  scale_y_continuous(labels=comma) +
  labs(x=vnames["wind_dir_9am"], y="Count")
```

If the values of the bars are supplied in the source dataset (the y-axis) ggplot2::geom_col() is used. In this example we manually count the frequency of each wind direction in the dataset prior to invoking ggplot2::ggplot(). This modified dataset is then passed on to ggplot2::ggplot() to display. Of course, it would be better in this specific example to let ggplot2::geom_bar() do the work itself as in Section 11.20.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

11.19 Bar Chart Texts

20210413

```
ds %>%
  ggplot(aes(x=wind_dir_3pm)) +
  geom_bar() +
  geom_text(stat="count", aes(label=comma(..count..)), vjust=-0.2, size=3) +
  scale_y_continuous(labels=comma) +
  labs(x=vnames["wind_dir_3pm"], y="Count")
```

Also see Section 11.14 and **??**.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

11.20 Bar Chart Wide Bars

20200428

```
ds %>%
  ggplot(aes(wind_dir_9am)) +
  geom_bar(width=1) +
  scale_y_continuous(labels=comma) +
  labs(x=vnames["wind_dir_9am"], y="Count")
```

The bars of a bar chart can be made to touch by specifying a full width with `width=1`. Also compare this plot to that in Section 11.18 where the counts are manually generated and ggplot2::geom_col() is used to plot the bars.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open*

source software since 1984.

Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

11.21 Bar Chart Wide and Borders

20200428

```
ds %>%
  ggplot(aes(wind_dir_9am)) +
  geom_bar(width=1, colour="blue", fill="grey") +
  scale_y_continuous(labels=comma) +
  labs(x=vnames["wind_dir_9am"], y="Count")
```

We can improve the appearance of a fat bar chart by adding, in this case, a blue border to the bars, using `colour=``"blue"` . By itself that would look a bit ugly, so we also `fill=``"grey"` the bars with grey rather than the default black. We can play with different colours to achieve a pleasing and personalised result.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open*

source software since 1984.

Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

11.22 Bar Chart Showcase Solar

20210413

```
solar %>%
  ggplot(aes(x=date, y=amount, fill=item)) +
  geom_col(position="dodge") +
  geom_label(aes(label=amount), vjust=-0.2, show.legend=FALSE,
             position=position_dodge(width=0.9), size=2) +
  theme(axis.text.x=element_text(angle=45, hjust=1)) +
  scale_x_discrete(breaks = unique(solar$date)) +
  ylim(0,600) +
  theme(legend.position="top") +
  labs(x="Quarter", y="Amount", fill="") +
  scale_fill_brewer(palette="Accent")
```

The `solar` dataset records 8 quarterly observations of the amount of electricity generated by solar photovoltaic panels, the amount consumed by the household, and a fixed overhead:

```
glimpse(solar)
```

```
## Rows: 24
## Columns: 3
## $ date   <ord> Jun 2019, Jun 2019, Jun 2019, Sep 2019, Sep 2019, Sep 2019, Dec…
## $ item   <ord> Fixed, Consumed, Generated, Fixed, Consumed, Generated, Fixed, …
## $ amount <dbl> 99, 170, 313, 97, 251, 329, 98, 115, 521, 106, 128, 514, 93, 34…
```

Exercise: *plot Fixed and Consumed as a single stacked bar placed beside the Generated bar.*

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

11.23 Box Plot Distributions

REVIEW

```
ds                                                                  %>%
  mutate(year=factor(format(ds$date, "%Y")))                        %>%
  ggplot(aes(x=year, y=max_temp, fill=year))                         +
  geom_boxplot(notch=TRUE)                                           +
  theme(legend.position="none")
```

A box plot, also known as a box and whiskers plot, shows the median (the second quartile) within a box which extends to the first and third quartiles. We note that each quartile delimits one quarter of the dataset and hence the box itself contains half the dataset.

Colour is added simply to improve the visual appeal of the plot rather than to convey new information. Since we include *fill*= we also turn off the otherwise included legend.

Here we observe the overall change in the maximum temperature over the years. Notice the first and last plots which probably reflect truncated data, providing motivation to confirm this in the data, before making significant statements regarding these observations.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

11.24 Colour Names

20200608

R provides a collection of predefined named colours as returned by grDevices::colours():

```
length(colours())
```

```
## [1] 657
```

```
colours() %>% sample(12)
```

```
##  [1] "orangered1"    "gray49"        "gainsboro"     "peachpuff4"
##  [5] "grey70"        "indianred4"    "cadetblue1"    "goldenrod2"
##  [9] "lightcoral"    "gray58"        "lemonchiffon1" "bisque1"
```

Colours can be selected using epitools::colors.plot() with `locator=``TRUE` to display a plot containing all of the named colours. Multiple clicks with the left mouse button on colours (we hear a beep for each click) will return the colour names.

```
as.character(colors.plot(locator=TRUE)$color.names)
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

11.25 Colour Ranges

20200608

Automatic colour ranges are supported using grDevices::palette(), as the default palette. A contiguous colour palette is available as grDevices::cm.colors(), with a rainbow available as grDevices::rainbow(). Others palletes includegrDevices::gray(), which provides a gray scale, colorRamps::blue2green2red(), and RColorBrewer::brewer.pal().

```
blue2green2red(15)
```

```
##  [1] "#0000CC" "#0000FF" "#0055FF" "#00AAFF" "#00FFFF" "#2BFFD5" "#55FFAA"
##  [8] "#80FF80" "#AAFF55" "#D4FF2B" "#FFFF00" "#FFAA00" "#FF5500" "#FF0000"
## [15] "#CC0000"
```

Use graphics::image() to display the colour plots as above.

```
image(matrix(1:80,  10), col=palette())
image(matrix(1:100, 10), col=cm.colors(10))
image(matrix(1:100, 10), col=rainbow(10))
image(matrix(1:60,  10), col=gray(seq(0.1, 0.9, len=6)))
image(matrix(1:150, 10), col=blue2green2red(15))
image(matrix(1:110, 10), col=brewer.pal(11, "PRGn"))
image(matrix(1:110, 10), col=brewer.pal(11, "RdBu"))
image(matrix(1:80,  10), col=brewer.pal(8,  "Accent"))
image(matrix(1:80,  10), col=brewer.pal(8,  "Dark2"))
image(matrix(1:90,  10), col=brewer.pal(9,  "Pastel1"))
image(matrix(1:120, 10), col=brewer.pal(12, "Paired"))
image(matrix(1:120, 10), col=brewer.pal(12, "Set3"))
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

11.26 Faceted Location Scatter Plot

REVIEW

```
ds %>%
  ggplot(aes(x=date, y=max_temp)) +
  geom_point(alpha=0.05, shape=".") +
  geom_smooth(method="gam", formula=y~s(x, bs="cs")) +
  facet_wrap(~location) +
  theme(axis.text.x=element_text(angle=45, hjust=1)) +
  labs(x=vnames["date"], y=vnames["max_temp"])
```

Partitioning the dataset by a categoric variable reduces the blob effect for big data. The plot uses `location` as the faceted variable to separately plot each location's maximum temperature over time. Notice the seasonal effect across all plots, some with quite different patterns.

The plot uses ggplot2::facet_wrap() to separately plot each location. Using ggplot2::geom_point() with *alpha*= reduces the effect of overlaid points as does using smaller dots on the plots by way of *shape*=. Together this works to de-clutter the plot and improves the presentation with an emphasis on the patterns. The x-axis tick labels are rotated 45° using *angle*= `45` within ggplot2::element_text() to avoid the labels overlapping. The *hjust*= `1` forces the labels to be right justified.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

11.27 Faceted Location Thin Lines

20200608

```
ds %>%
  ggplot(aes(x=date, y=max_temp)) +
  geom_line(alpha=0.1, size=0.05) +
  geom_smooth(method="gam", formula=y~s(x, bs="cs"), size=0.05) +
  facet_wrap(~location) +
  theme(axis.text.x=element_text(angle=45, hjust=1)) +
  labs(x=vnames["date"], y=vnames["max_temp"])
```

An alternative is to present the plot as a line chart rather than a scatter plot. It does make more sense for a time series plot such as this, though the effect is little changed due to the amount of data being displayed.

Changing to lines simply uses ggplot2::geom_line() instead of ggplot2::geom_point(). Very thin lines are used as specified through the *size*= option. Nonetheless, the data remains quite dense.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

11.28 Faceted Wind Direction

20200608

```
lblr <- function(x)
{
  x %>%
    str_replace_all("n", "North ") %>%
    str_replace_all("s", "South ") %>%
    str_replace_all("e", "East ") %>%
    str_replace_all("w", "West ") %>%
    str_replace(" $", "")
}

ds %>%
  sample_n(10000) %>%
  ggplot(aes(x=min_temp, y=max_temp, colour=rain_tomorrow)) +
  geom_point(shape=".") +
  geom_smooth(method="gam", formula=y~s(x, bs="cs")) +
  facet_wrap(~wind_dir_3pm, labeller=labeller(wind_dir_3pm=lblr)) +
  labs(x      = vnames["min_temp"],
       y      = vnames["max_temp"],
       colour = vnames["rain_tomorrow"])
```

Labels of a faceted plot can be modified as here expanding n to North, s to South, etc. Observe that the linear relationship for rainy days is below that for dry days. The maximum temperature is generally closer to the minimum temperature on days where it rains the following day.

The function to remap the directions uses stringr::str_replace_all() to do the work. It is then transformed into a ggplot2::labeller() for *wind_dir_3pm*=.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

11.29 Labels

20200422 The x-axis and y-axis labels can be specified using ggplot2::xlab() and ggplot2::ylab():

```
...
xlab("Minimum Temperature") +
ylab("Maximum Temperature") +
...
```

The labels can alternatively be specified using ggplot2::labs() which also accepts many other label options too:

```
...
labs(x="Minimum Temperature",
     y="Maximum Temperature",
     colour="Rain")
...
```

A quick way to remove a label is to specify `NULL`:

```
...
xlab(NULL) +
...
```

See Section @ref(ggplot2:remove_axis_labels) to remove the label, ticks, and text.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

11.30 Labels with Comma

20200428

```
ds %>%
  ggplot(aes(wind_dir_9am, fill=wind_dir_9am)) +
  geom_bar() +
  scale_y_continuous(labels=comma) +
  theme(legend.position="none") +
  labs(x=vnames["wind_dir_9am"], y="Count")
```

With ggplot2 (Wickham et al. 2020) Version 0.9.0 the package introduced a mechanism to handle many of the scale operations, in such a way as to support base and lattice graphics, as well as graphics. Scale operations include position guides, as in the axes, and aesthetic guides, as in the legend.

The function scales::comma() is useful for presenting numbers using commas to separate the thousands. This is *always* a good idea as it assists the reader in quickly determining the magnitude of the numbers we are looking at. As a matter of course, I recommend commas in

plots (and tables) at all times. To do so use the ggplot2::scale_continuous() function with *labels=* `comma`.

References

Wickham, Hadley, Winston Chang, Lionel Henry, Thomas Lin Pedersen, Kohske Takahashi, Claus Wilke, Kara Woo, Hiroaki Yutani, and Dewey Dunnington. 2020. *Ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics*. https://CRAN.R-project.org/package=ggplot2.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

11.31 Labels with Dollars

20200428

```
ds %>%
  ggplot(aes(wind_dir_3pm, fill=wind_dir_3pm)) +
  geom_bar() +
  scale_y_continuous(labels=dollar) +
  labs(y="") +
  labs(x=vnames["wind_dir_3pm"], y="") +
  theme(legend.position="none")
```

The package provides numerous formatting options that can be readily utilised for plot labels. The function scales::dollar() is useful for presenting numbers that represent currency, specifying, for example, the ggplot2::scale_continuous() function with *labels*= `dollar`. The numbers being plotted here are not currency, but are used for demonstrating the functionality. Notice that comma is also used in the displayed values.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

11.32 Labels Removed

{#ggplot2:remove_axis_labels}

20200317

```
ds %>%
  subset(location %in% c("Canberra", "Adelaide", "Darwin")) %>%
  sample_frac(0.2) %>%
  ggplot(aes(min_temp, max_temp, colour=location)) +
  geom_point() +
  theme(axis.ticks.x = element_blank(),
        axis.text.x  = element_blank(),
        axis.title.x = element_blank(),
        axis.ticks.y = element_blank(),
        axis.text.y  = element_blank(),
        axis.title.y = element_blank())
```

The top plot has default x-axis and y-axis ticks, text, and title. To remove axis labels altogether the bottom plot adds a layer to set the axis ticks, text, and title to ggplot2::element_blank() which is used to draw nothing and to take no space within a non-data component of the plot, using the ggplot2::theme() system:

Notice the use of dplyr::sample_frac() to plot a 20% random sample of the dataset and thus the actual points plotted in the two plots above are randomly different.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

11.33 Labels Rotated

20200428

```
ds %>%
  ggplot(aes(x=location, y=temp_3pm, fill=location)) +
  stat_summary(fun="mean", geom="bar") +
  labs(x=vnames["location"],
       y=glue('Mean {vnames["temp_3pm"]}')) +
  theme(legend.position="none",
        axis.text.x=element_text(angle=45, hjust=1, size=6))
```

When a bar chart has many bars and non-trivial labels then it is useful to rotate the labels. Here we plot the mean temperature at 3pm for each location in the dataset. Without rotating the x-axis labels (the locations) the x-axis will be blighted with much overlapping. Labels can be rotated using a ggplot2::theme() option of *axis.text.x*= with *angle*=45°. Use *hjust*=1 to align the labels appropriately.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

11.34 Line Chart Basic

20180603

```
set.seed(26439)
ds %>%
  sample_n(100) %>%
  ggplot(aes(x=min_temp, y=max_temp)) +
  geom_line()
```

As an alternative to plotting the points we could join the dots and plot a line. One reason for doing so might be to attempt to observe more clearly any relationship between the two dimensions (i.e., the two variables we are plotting). Add a ggplot2::geom_line() to the canvas we can draw a line from left to right the follows the points exactly. The result is a rather jugged line and it is debatable whether it has added any further insight into our understanding of the data.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

11.35 Line Chart Density Distribution

20200428

```
ds %>%
  ggplot(aes(x=max_temp)) +
  geom_density() +
  labs(x=vnames["max_temp"], y="Density")
```

A density plot using ggplot2::geom_density() will convey the distribution of the values of a variable. Here, for example, we can see that `max_temp` has a roughly normal distribution with a peak around the low 20's (degrees Celsius) and fairly short tails.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

11.36 Line Chart Skewed Distributions

20200823

```
ds %>%
  filter(rainfall != 0) %>%
  ggplot(aes(x=rainfall)) +
  geom_density() +
  scale_y_continuous(labels=comma) +
  theme(legend.position="none")
```

The plot informs us about the skewed nature of the amount of rainfall recorded for any one day, but we lose a lot of resolution at the low end.

Note that we use a subset of the dataset to include only those observations of rainfall (i.e., where the rainfall is non-zero). Otherwise a warning will note many rows contain non-finite values in calculating the density statistic.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of*

Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

11.37 Line Chart Log X Axis

20200823

```
ds                                                          %>%
  filter(rainfall != 0)                                     %>%
  ggplot(aes(x=rainfall))                                    +
  geom_density()                                             +
  scale_x_log10()                                            +
  theme(legend.position="none")
```

A common approach to dealing with the skewed distribution is to transform the scale to be a logarithmic scale, base 10. The following code produces Figure @ref(fig:ggplot2:frequency_rainfall_log_x).

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

11.38 Line Chart Log Breaks

20200823

```
ds                                                                  %>%
  filter(rainfall != 0)                                             %>%
  ggplot(aes(x=rainfall))                                            +
  geom_density() +
  scale_x_log10(breaks=c(1, 10, 100), labels=comma)                  +
  theme(legend.position="none")
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

11.39 Line Chart Log Ticks

20200823

```
ds                                                                  %>%
  filter(rainfall != 0)                                             %>%
  ggplot(aes(x=rainfall))                                            +
  geom_density() +
  scale_x_log10(breaks=c(1, 10, 100), labels=comma)                  +
  annotation_logticks(sides="bt")                                    +
  theme(legend.position="none")
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

11.40 Line Chart Log Custom Labels

REVIEW

```
mm <- function(x) { sprintf("%smm", x) }

ds                                                                    %>%
  filter(rainfall != 0)                                               %>%
  ggplot(aes(x=rainfall))                                              +
  geom_density()                                                       +
  scale_x_log10(breaks=c(1, 10, 100), labels=mm)                       +
  annotation_logticks(sides="bt")                                      +
  theme(legend.position="none")
```

% %———————————————————————————————— % ## Pairs Plot: Using ggpairs() % % ————————————————————————————————

% <<ggplot2:ggpairs, echo=FALSE, fig.width=12, fig.height=12, out.width='\textwidth,' message=FALSE} % weather[c(3,4,6,7,10,19,22,24)] %>% % na.omit() %>% % ggpairs(cardinality_threshold=16) % @

% <<ggplot2:ggpairs_echo, eval=FALSE} % @

% A sophisticated pairs plot, as delivered by GGally::ggpairs() from % , brings together many of the kinds of plots we have % already seen to compare pairs of variables. Here we have used it to % plot the dataset from . We see % scatter plotes, histograms, and box plots in the pairs plot.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

11.41 Pie Chart

REVIEW

```
ds %>%
  group_by(rain_tomorrow) %>%
  count() %>%
  ungroup()  %>%
  mutate(per=round(`n`/sum(`n`), 2)) %>%
  mutate(label=paste(rain_tomorrow, percent(per))) %>%
  arrange(per) %>%
  ggplot(aes(x=1, y=per, fill=rain_tomorrow)) +
  geom_bar(stat="identity") +
  coord_polar(theta='y') +
  theme_void() +
  theme(legend.position="none") +
  geom_text(aes(x=1, y=cumsum(per)-per/2, label=label), size=8)
```

A pie chart is a popular circular plot showing the relative proportions through angular slices. Generally, pie charts are not recommended, particularly for multiple wedges, because humans generally have difficulty perceiving the relative angular differences between slices. For two or three slices it may be argued that the pie chart is just fine, particularly if further information is provided, such as labelling the slices with their sizes.

See also the and for explorations of the utility of pie charts.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open*

source software since 1984.

Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

11.42 Plotting Regions

REVIEW The basic concept of R's graphics model is that a plot is built up layer by layer. Each latter component of the plot overlays earlier components. A plot also has two components. The plotting area is identified by through the parameter, as 4 numbers x_1, x_2, y_1, and y_2. You can retrieve the current plotting region (which is defined by the first component of a plot) with:

```
> plot(rnorm(10))
> par("usr")
[1]  0.640000 10.360000 -1.390595  1.153828
```

The whole figure itself will encompass the plotting region and the region around the plot used to add axis information and labels. Outside of the figure region is the device region. Normally, adding components to a plot, outside of the plotting region, will have no effect—they will be cropped. To ensure they do not get cropped, set the graphic parameter to TRUE:

```
> par(xpd=TRUE)
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

11.43 Rose Chart

20210318

```r
ds %>%
  filter(location=="Canberra",
         date >= as.Date("2019-01-01"), date <= as.Date("2020-12-31")) %>%
  mutate(month = lubridate::month(date, label=TRUE, abbr=TRUE),
         year  = lubridate::year(date)) %>%
  group_by(month, year) %>%
  summarise(hot   = sum(max_temp > 30),
            wet   = sum(rainfall > 1),
            windy = sum(wind_gust_speed > 60)) %>%
  pivot_longer(cols=hot:windy, names_to="measure", values_to="days") %>%
  ggplot(aes(x=month, y=days)) +
  geom_col(aes(fill=measure), width=1, position="identity") +
  coord_polar() +
  facet_wrap(~year)  +
  scale_fill_manual(labels=c("Hot", "Wet", "Windy"),
                    values=c("firebrick1", "skyblue3", "grey80")) +
  scale_y_sqrt() +
  theme_void() +
  theme(axis.text.x = element_text(size = 9),
        strip.text = element_text(size = 11),
        legend.position = "bottom",
        plot.background = element_rect(fill = alpha("cornsilk", 0.5)),
        plot.margin = unit(c(10, 10, 10, 10), "pt"),
        plot.title = element_text(vjust = 5)) +
  labs(fill="Number of Days ...") +
  ggtitle("Weather Extremes in Canberra Measuring Number of Days")
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

11.44 Rose Chart Discussion

20210318

According to Neil Saunders Florence Nightingale developed the concept of a rose chart to visualise the cause of mortality in the army. Neil implemented the chart in R.

The basic concept of a rose chart is that it is a bar chart projected onto polar coordinates.

The plot here replicates the rose chart for the weather dataset. It nicely illustrates the changed weather pattern from 2019 to 2020. The weather patterns used are a count of the number of days each month that are *hot*, *wet*, and *windy*.

The charts tell quite an interesting story. Comparing 2019 to 2020 we can see some stark weather pattern differences. The year 2019, in Canberra, was generally hotter than 2020, but 2020 was a lot wetter. We had major bush fires at the end of a drought in Canberra from the end of 2019 to

the beginning of 2020. Then 2020 was particularly wet.

The data processing begins with dplyr::filter()'ing the daily weather observations from Canberra airport for the 2019 and 2020 years. The date is dplyr::mutate()'d into *month* and *year* which is how the data will be dplyr::group_by() for display. It is then dplyr::summarise()'d to include measures of *hot*, *wet*, and *windy* as the base::sum() of the number of days in the month above a threshold. With a tidyr::pivot_longer() the table is in the format required for plotting.

The ggplot2::ggplot() itself is a ggplot2::geom_col() using the supplied data (day counts) as the column values. It is plot on to ggplot2::coord_polar() and ggplot2::facet_wrap()'d on the *year*. The remainder deals with aspects of the presentation.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

11.45 Save Plot to File

20200820 Generating a plot is one thing but we will want to make use of the plot possibly in multiple ways. Once we have a plot displayed we can save the plot to file quite simply using ggplot2::ggsave(). The format is determined automatically by the name of the file to which we save the plot. Here, for example, we save the plot as a PDF that we might include in other documents or share with a colleague for discussion. Using `11` and `7`, the plot will be wider that it is tall.

```
ggsave("myplot.pdf", width=11, height=7)
```

The default values for *width*= and *height*= are those of the current plotting window. Specifying them explicitly overrides the current window dimensions. By trial and error or by experience we find the dimensions that suit our requirements.

There is some art required in choosing a good width and height as discussed in Chapter 26. By increasing the height or width any text that is displayed on the plot essentially stays the same size. Thus by increasing the plot size the text will appear smaller. By decreasing the plot size the text becomes larger. Experimentation can help get the right size for any particular purpose.

The function ggplot2::ggsave() is a wrapper around various other R functions that save plots to files, and adds some value to these default plotting functions.

A more traditional approach to saving a plot is to initiate a graphic device such as grDevices::pdf(), print the plot as below where we assume the plot is saved in the template variable `p`, and then close the device with grDevices::dev.off().

```
pdf("myplot.pdf", width=11, height=7)
p
dev.off()
```

For a grDevices::pdf() device the default *width*= and *height*= are 7 (inches). Thus, for both of the above examples we are widening the plot whilst retaining the height.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

11.46 Scatter Plot

20200822

```
set.seed(26439)
ds %>%
  sample_n(100) %>%
  ggplot(aes(x=min_temp, y=max_temp)) +
  geom_point() +
  labs(x = vnames["min_temp"],
       y = vnames["max_temp"])
```

The simplest of plots is a scatter plot which displays a two dimensional plot of points. The dimensions are the and axes. The observations from the dataset, which are the rows in the dataset, are plotted, scattering the points over the plot.

Using the template variable `ds` the dataset is piped (`%>%`) into ggplot2::ggplot(). The aesthetics ggplot2::aes() are set up with `min_temp` as the axis and `max_temp` as the axis. To add the points (observations) to the plot we employ ggplot2::geom_point().

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

11.47 Scatter Plot Colour

20180603

```
set.seed(26439)
ds %>%
  sample_n(100) %>%
  ggplot(aes(x=min_temp, y=max_temp, colour=rain_tomorrow)) +
  geom_point() +
  labs(x      = vnames["min_temp"],
       y      = vnames["max_temp"],
       colour = vnames["rain_tomorrow"])
```

A key variable in the rattle::weatherAUS dataset is `rain_tomorrow`. By colouring the dots according to the `rain_tomorrow` we may see a relationship in the data.

The colour is added simply by specifying a further aesthetic, *colour*= `rain_tomorrow`. This results in the different values of the variable `rain_tomorrow` being coloured differently.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

11.48 Scatter Plot Colour Alternative

20200608

```
ds %>%
  sample_n(1000) %>%
  ggplot(aes(x=min_temp, y=max_temp, colour=rain_tomorrow)) +
  geom_point() +
  scale_colour_brewer(palette="Set2") +
  labs(x      = vnames["min_temp"],
       y      = vnames["max_temp"],
       colour = vnames["rain_tomorrow"])
```

The simplest plot is a scatter plot which displays points scattered over a plot. If the dataset is large the resulting plot will be rather dense. For illustrative purposes a random subset of just 1,000 observations is used. A linear relationship between the two variables can be seen.

The random sample of 1,000 rows is generated using dplyr::sample_n() and is then piped through to ggplot2::ggplot(). The function argument identifies the aesthetics of the plot so that *x*= associates the variable `min_temp` with the x-axis and *y*= associates the variable `max_temp`

with the y-axis.

In addition the *colour*= option provides a mechanism to distinguish between days where the observation `rain_tomorrow` is Yex and where it is No. A colour palette can be chosen using ggplot2::scale_colour_brewer().

A graphical layer is added to the plot consisting of (x, y) points coloured appropriately. The function ggplot2::geom_point() achieves this.

The original variable names stored as `vnames` are used to label the plot using ggplot2::labs(). The original names will make more sens to the reader than our chosen normalised names.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

11.49 Scatter Plot Smooth Gam

REVIEW

```
ds %>%
  filter(location=="Canberra") %>%
  ggplot(aes(x=date, y=max_temp)) +
  geom_point(shape=".") +
  geom_smooth(method="gam", formula=y~s(x, bs="cs")) +
  labs(x=vnames["date"], y=vnames["max_temp"])
```

This scatter plot of *x*= `date` against *y*= `max_temp` shows a pattern of seasonality over the dataset and a trend line over the period of the dataset.

The scatter plot is again created using ggplot2::geom_point(). Typical of scatter plots of big data there will be many overlaid points. To reduce the impact the points are reduced to a small dot using *shape*= `"."` .

An additional layer of a smooth fitted curve, using ggplot2::geom_smooth(), is added. The dataset has many points and so a smoothing method recommended is *method*= `"gam"` which will automatically be chosen if not specified but with a message to that effect. The formula specified using *formula*= is also the default for *method*= `"gam"`.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

11.50 Scatter Plot Smooth Loess

20180603

```
set.seed(26439)
ds %>%
  sample_n(100) %>%
  ggplot(aes(x=min_temp, y=max_temp)) +
  geom_point() +
  stat_smooth(method="loess", formula=y~x)
```

A smooth curve here fits the statistical loess function for a single smoother line to the points.

The resulting plot uses ggplot2::geom_smooth() with a so called locally weighted scatterplot smoothing method or LOESS to produce the smoothed line. As a result we can observe statistically something of a relationship between the two variables although it is not a simple relationship.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

11.51 Transparent Plots

% When I do this manually, the order in the legend is alphabetic, as I % want. In knitr it is reveresed.

REVIEW

```
cities <- c("Canberra", "Darwin", "Melbourne", "Sydney")

ds                                                                  %>%
  filter(location %in% cities)                                      %>%
  ggplot(aes(temp_3pm, colour = location, fill = location))          +
  geom_density(alpha = 0.55)
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

11.52 Violin Plot

REVIEW

```
ds %>%
  mutate(year=factor(format(ds$date, "%Y"))) %>%
  ggplot(aes(x=year, y=max_temp, fill=year)) +
  geom_violin() +
  theme(legend.position="none")
```

A violin plot is another interesting way to present a distribution, using a shape that resembles a violin.

We again use colour to improve the visual appeal of the plot.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of*

Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

11.53 Violin Plot Embedded Box Plot

REVIEW

```
ds                                                                    %>%
  mutate(year=factor(format(ds$date, "%Y")))                          %>%
  ggplot(aes(x=year, y=max_temp, fill=year))                           +
  geom_violin()                                                        +
  geom_boxplot(width=.5, position=position_dodge(width=0))             +
  theme(legend.position="none")
```

We can overlay the violin plot with a box plot to show the quartiles.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open*

source software since 1984.

Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

11.54 Violin Plot Faceted Location

REVIEW

```
ds %>%
  mutate(year=factor(format(ds$date, "%Y"))) %>%
  filter(location %in% (ds$location %>% unique %>% sample(12))) %>%
  sample_frac(0.1) %>%
  ggplot(aes(x=year, y=max_temp, fill=year)) +
  geom_violin() +
  geom_boxplot(width=.5, position=position_dodge(width=0)) +
  theme(legend.position="none") +
  theme(axis.text.x=element_text(angle=45, hjust=1)) +
  labs(x="Year", y=vnames["max_temp"]) +
  facet_wrap(~location)
```

We can readily split the plot across the locations. Things get a little crowded, but we get an overall view across all of the different weather stations. Notice we also rotated the x-axis labels so that they don't overlap.

We can immediately see one of the issues with this dataset, noting that three weather stations have fewer observations that then others.

Various other observations are also interesting. Some locations have little variation in their maximum temperatures over the years.

%## Waffle Chart

% This is a square pie chart :-)

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

11.55 XFig Support

REVIEW Another format supported is XFIG. Such output is editable with the *xfig* graphics editor, allowing further annotations and modifications to be made to the automatically generated plot. The XFIG graphics can then be converted to an even larger collection of graphics formats, including PDF. For the graphics actually presented here in the book R has been used, in fact, to generate XFIG output which is then converted to PDF. Thus the code examples here, generating PDF directly, may give slightly different layouts to the figures that actually appear here.

A highly interoperable approach is to generate graphs in FIG format which can then be loaded into the **xfig** application, for example, for further editing. This allows, for example, minor changes to be made to fine tune the graphics, but at the cost of losing the ability to automatically regenerate the plot from the original R code. For LaTeX processing the **rubber** package (under Debian GNU/Linux) will automatically convert them to the appropriate EPS or PDF format. Of course, xfig can also generate PNG and JPG and many other formats.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

12 Statistics

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

12.1 Analysis of Variance ANOVA

```
ds %>%
  group_by(x) %>%
  summarise(
    n    = n(),
    sm   = mean(y),
    rssq = sum((y - sm)^2)
  ) %>%
  summarise(
    om   = sum(n * sm) / sum(n),
    ssq  = sum(n * (sm - om)^2),
    rssq = sum(rssq),
    df1  = n_distinct(x) - 1,
    mse  = ssq / df1,
    df2  = sum(n) - n_distinct(x),
    rmse = rssq / df2,
    F    = mse / rmse,
    P    = pf(F, df1, df2, lower.tail=FALSE)
  )
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

13 Spatial Data and Maps

Focus on rosm throughout the chapter.

Spatial data is central to many of our tasks as Data Scientists. Identifying spatial patterns and spatial correlations and relationships between those patterns offers opportunities for identifying, for example, locations of new services for populations. Imagine predicting common routes for travellers this morning, and dynamically routing public transport to meet those needs. Fundamental to the Data Scientist is the ability to process, visualize and then model spatial data. Done right maps can be a very effective communications tool. Numerous R packages work together to bring us a sophisticated mapping and spatial analysis capability.

The required packages for this chapter include:

Spatial data is central to many of our tasks as Data Scientists. Identifying patterns, correlations and relationships between those patterns delivers opportunities for delivering new services. Imagine predicting common routes for travellers this morning, and dynamically routing public transport to meet those needs. Fundamental to the Data Scientist is the ability to process, visualize and then model spatial data. Done right maps can be a very effective communications tool. Numerous R packages work together to bring us a sophisticated mapping and spatial analysis capability.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

13.1 Geocodes

One of the fundamental things about spatial data and mapping is the geographic coordinate system used to uniquely identify locations. We use longitude (x axis, abbreviated lon) and latitude (y axis, abbreviated lat) for locations on our planet. The longitude is the angle from the meridian through Greenwich and the latitude is the angle from the equator. We can use street addresses and locations. Here are a few examples.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

13.2 Understanding Spatial Data

Spatial data involves a combination of information about locations and geometric features related that best describe the location. The most commonly used formats to store spatial information are Raster and Vector. A Raster format is a type of digital image represented by grids which can be reduced and enlarged. They are made up of a matrix of pixels with each pixel representing an area within the spatial object. A Vector format consists of geometric shapes organized in the form of shapefiles. A shapefile consist of three files under the extensions .shp, .shx and .dbf which contain the features, indices of shapes and attributes respectively.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

13.3 Plotting Shapefiles

Let us visualize the streets and places of Singapore. We can obtain the data from data.gov.sg . We shall project it on top of a base map obtained from OpenStreetMap.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

13.4 Google Maps: Geocoding

One of the fundamental things about spatial data and mapping is the geographic coordinate system used to uniquely identify locations. We use longitude (x axis, abbreviated lon) and latitude (y axis, abbreviated lat) for locations on our planet. The longitude is the angle from the meridian through Greenwich and the latitude is the angle from the equator. We can use to street addresses and locations. Here are a few examples.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

14 Model Template

20200607 We started with a template for data wrangling in Chapter 8, explored the data in Chapter 9, learnt how to transform the data in Chapter 1.1, and gained initial insights through visualisations in Chapter 11. We are now ready to use machine learning to build analytic models and begin to understand the knowledge the models are capturing. In this chapter we develop a template for building and evaluating models. As with the data template in Chapter 8 the intention of the template is to provide a starting point for building and evaluating models.

R offers an full suite of model builders. We will use one such model builder in this chapter. A model builder is access via a function call and the arguments to the function are generally similar across different model builders. Non-the-less, each algorithm is generally implemented by different developers and so idiosyncratic differences occur.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

14.1 ML Setup

20210102 Packages used in this chapter include ROCR (Sing et al. 2020), ggplot2 (Wickham et al. 2020), rpart (Therneau and Atkinson 2019), and rattle (G. Williams 2020).

Packages are loaded into the currently running R session from your local library directories on disk. Missing packages can be installed using utils::install.packages() within R. On Ubuntu, for example, R packages can also be installed using `$ wajig install r-cran-<pkgname>` .

```
# Load required packages from local library into the R session.

library(ROCR)         # Use prediction() for evaluation.
library(ggplot2)      # Display evaluations.
library(rattle)       # Dataset: weather.
library(rpart)        # ML: decision tree rpart().
library(scales)       # Support: commas(), percent().
```

References

Sing, Tobias, Oliver Sander, Niko Beerenwinkel, and Thomas Lengauer. 2020. *ROCR: Visualizing the Performance of Scoring Classifiers*. http://ipa-tys.github.io/ROCR/.

Therneau, Terry, and Beth Atkinson. 2019. *Rpart: Recursive Partitioning and Regression Trees*. https://CRAN.R-project.org/package=rpart.

Wickham, Hadley, Winston Chang, Lionel Henry, Thomas Lin Pedersen, Kohske Takahashi, Claus Wilke, Kara Woo, Hiroaki Yutani, and Dewey Dunnington. 2020. *Ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics*. https://CRAN.R-project.org/package=ggplot2.

Williams, Graham. 2020. *Rattle: Graphical User Interface for Data Science in r*. https://rattle.togaware.com/.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of*

Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

14.2 ML Data and Variables

20210104

The rattle::weatherAUS dataset is loaded into the template variable `ds` and further template variables are setup as introduced by Graham J. Williams (2017). See Chapter 8 for details.

```
dsname <- "weatherAUS"
ds     <- get(dsname)

nobs   <- nrow(ds)

vnames <- names(ds)
ds    %<>% clean_names(numerals="right")
names(vnames) <- names(ds)

vars   <- names(ds)
target <- "rain_tomorrow"
vars   <- c(target, vars) %>% unique() %>% rev()
```

It is always useful to remind ourselves of the dataset with a random sample:

```
ds  %>% sample_frac() %>% select(date, location, sample(3:length(vars), 5))
```

```
## # A tibble: 176,747 x 7
##    date       location    wind_dir_9am rainfall min_temp wind_speed_3pm temp_9am
##    <date>     <chr>       <ord>           <dbl>    <dbl>          <dbl>    <dbl>
##  1 2018-01-29 NorahHead   NNE               0.2     21.7             17     24.5
##  2 2017-11-22 Dartmoor    NNE               0       14.9             11     26.4
##  3 2020-02-16 MelbourneA… SW                3.4     15.2             11     17.1
##  4 2015-11-03 Penrith     SW                0.2     17.7             11     18.2
##  5 2019-07-26 Canberra    NNE               0       -1.8              9      2
##  6 2010-09-26 Watsonia    NE                0        6.9             19     11.8
##  7 2010-09-21 BadgerysCr… NNW               0.2      6.3             24     16.6
##  8 2010-08-24 Brisbane    W                10.2     14               20     17.9
##  9 2019-07-10 Penrith     NE                0        3.1             11      9.7
## 10 2012-11-07 PerthAirpo… SW                0.2      9.5             26     18.2
## # … with 176,737 more rows
```

References

Williams, Graham J. 2017. *The Essentials of Data Science: Knowledge Discovery Using r*. The r Series. CRC Press.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

14.3 ML Modelling Setup

20200607

For the rattle::weatherAUS dataset we similarly define the following template variables (Graham J. Williams 2017) used for predictive modelling. See Chapter 8 for details.

```
risk   <- "risk_mm"
id     <- c("date", "location")
ignore <- c(risk, id)
vars   <- setdiff(vars, ignore)
inputs <- setdiff(vars, target)

form   <- formula(target %s+% " ~ .")

ds[vars] %<>% na.roughfix()

SPLIT <- c(0.70, 0.15, 0.15)

nobs %>% sample(SPLIT[1]*nobs)                                -> tr
nobs %>% seq_len() %>% setdiff(tr) %>% sample(SPLIT[2]*nobs) -> tu
nobs %>% seq_len() %>% setdiff(tr) %>% setdiff(tu)           -> te

ds %>% slice(tr) %>% pull(target) -> actual_tr
ds %>% slice(tu) %>% pull(target) -> actual_tu
ds %>% slice(te) %>% pull(target) -> actual_te

ds %>% slice(tr) %>% pull(risk) -> risk_tr
ds %>% slice(tu) %>% pull(risk) -> risk_tu
ds %>% slice(te) %>% pull(risk) -> risk_te
```

The 176,747 observations from the dataset have been randomly partitioned into a *training* dataset with 123,722 observations, a *tuning* dataset with 26,512 observations, and a *testing* dataset with 26,513 observations. The target variable (rain_tomorrow) has the classes: No (139670), Yes (37077).

References

Williams, Graham J. 2017. *The Essentials of Data Science: Knowledge Discovery Using r*. The r Series. CRC Press.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

14.4 ML Data Glimpse

20210104

A dplyr::glimpse() over all the variables of the dataset provides a fuller picture of the data.

```
glimpse(ds)
```

```
## Rows: 176,747
## Columns: 24
## $ date            <date> 2008-12-01, 2008-12-02, 2008-12-03, 2008-12-04, 2008-…
## $ location        <chr> "Albury", "Albury", "Albury", "Albury", "Albury", "Alb…
## $ min_temp        <dbl> 13.4, 7.4, 12.9, 9.2, 17.5, 14.6, 14.3, 7.7, 9.7, 13.1…
## $ max_temp        <dbl> 22.9, 25.1, 25.7, 28.0, 32.3, 29.7, 25.0, 26.7, 31.9, …
## $ rainfall        <dbl> 0.6, 0.0, 0.0, 0.0, 1.0, 0.2, 0.0, 0.0, 0.0, 1.4, 0.0,…
## $ evaporation     <dbl> 4.8, 4.8, 4.8, 4.8, 4.8, 4.8, 4.8, 4.8, 4.8, 4.8, 4.8,…
## $ sunshine        <dbl> 8.5, 8.5, 8.5, 8.5, 8.5, 8.5, 8.5, 8.5, 8.5, 8.5, 8.5,…
## $ wind_gust_dir   <ord> W, WNW, WSW, NE, W, WNW, W, W, NNW, W, N, NNE, W, SW, …
## $ wind_gust_speed <dbl> 44, 44, 46, 24, 41, 56, 50, 35, 80, 28, 30, 31, 61, 44…
## $ wind_dir_9am    <ord> W, NNW, W, SE, ENE, W, SW, SSE, SE, S, SSE, NE, NNW, W…
## $ wind_dir_3pm    <ord> WNW, WSW, WSW, E, NW, W, W, W, NW, SSE, ESE, ENE, NNW,…
## $ wind_speed_9am  <dbl> 20, 4, 19, 11, 7, 19, 20, 6, 7, 15, 17, 15, 28, 24, 4,…
## $ wind_speed_3pm  <dbl> 24, 22, 26, 9, 20, 24, 24, 17, 28, 11, 6, 13, 28, 20, …
## $ humidity_9am    <dbl> 71, 44, 38, 45, 82, 55, 49, 48, 42, 58, 48, 89, 76, 65…
## $ humidity_3pm    <dbl> 22, 25, 30, 16, 33, 23, 19, 19, 9, 27, 22, 91, 93, 43,…
## $ pressure_9am    <dbl> 1007.7, 1010.6, 1007.6, 1017.6, 1010.8, 1009.2, 1009.6…
## $ pressure_3pm    <dbl> 1007.1, 1007.8, 1008.7, 1012.8, 1006.0, 1005.4, 1008.2…
## $ cloud_9am       <dbl> 8, 5, 5, 5, 7, 5, 1, 5, 5, 5, 5, 8, 8, 5, 5, 0, 8, 8, …
## $ cloud_3pm       <dbl> 5, 5, 2, 5, 8, 5, 5, 5, 5, 5, 5, 8, 8, 7, 5, 5, 1, 1, …
## $ temp_9am        <dbl> 16.9, 17.2, 21.0, 18.1, 17.8, 20.6, 18.1, 16.3, 18.3, …
## $ temp_3pm        <dbl> 21.8, 24.3, 23.2, 26.5, 29.7, 28.9, 24.6, 25.5, 30.2, …
## $ rain_today      <fct> No, No, No, No, No, No, No, No, No, Yes, No, Yes, Yes,…
## $ risk_mm         <dbl> 0.0, 0.0, 0.0, 1.0, 0.2, 0.0, 0.0, 0.0, 1.4, 0.0, 2.2,…
## $ rain_tomorrow   <fct> No, No, No, No, No, No, No, No, Yes, No, Yes, Yes, Yes…
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

14.5 Model Building

20200607 We now build, fit, or train a model. R has most machine learning algorithms available. We will begin with a simple favourite—the decision tree algorithm— using rpart::rpart(). We record this information using the generic variables `mdesc` (human readable description of the model type) and `mtype` (type of the model).

```
mtype <- "rpart"
mdesc <- "decision tree"
```

The model will be built using tidyselect::all_of() the dplyr::select()'ed variables from the training dplyr::slice() of the dataset. The training slice is identified as the row numbers stored as `tr` and the column names stored as `vars`. This training dataset is piped on to rpart::rpart() together with a specification of the model to be built as stored in `form`. Using generic variables allows us to change the formula, the dataset, the observations and the variables used in building the model yet retain the same programming code. The resulting model is saved into the variable `model`.

```
ds %>%
  select(all_of(vars)) %>%
  slice(tr) %>%
  rpart(form, .) ->
model
```

To view the model simply reference the generic variable `model` on the command line. This asks R to base::print() the model.

```
model
```

```
## n= 123722
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##  1) root 123722 25921 No (0.7904900 0.2095100)
##    2) humidity_3pm< 71.5 104352 14394 No (0.8620630 0.1379370) *
##    3) humidity_3pm>=71.5 19370  7843 Yes (0.4049045 0.5950955)
##      6) humidity_3pm< 83.5 11433  5331 No (0.5337182 0.4662818)
##       12) wind_gust_speed< 42 6885  2533 No (0.6320988 0.3679012) *
##       13) wind_gust_speed>=42 4548  1750 Yes (0.3847845 0.6152155) *
##      7) humidity_3pm>=83.5 7937  1741 Yes (0.2193524 0.7806476) *
```

This textual version of the model provides the basic structure of the tree. We present the details in Chapter 20. Different model builders will base::print() different information.

This is our first predictive model. Be sure to spend some time to understand and reflect on the knowledge that the model is exposing.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

14.6 Predict Class

20200607 R provides stats::predict() to obtain predictions from the model. We can obtain the predictions, for example, on the `te` st dataset and thereby determine the apparent accuracy of the model.

For different types of models stats::predict() will behave in a similar way. There are however variations that we need to be aware of for each. For an `\Sexpr{mtype}` model to predict the class (i.e., `Yes` or `No`) use *type*= `"class"`:

```
ds[te, vars] %>% predict(model, newdata=., type="class") -> predict_te

head(predict_te)
```

```
##  1  2  3  4  5  6
## No No No No No No
## Levels: No Yes
```

We can then compare this to the actual class for these observations as is recorded in the original `te` dataset. The actual classes have already been stored as the variable `target_te`:

```
head(actual_te)
```

```
## [1] No No No No No No
## Levels: No Yes
```

We can observe from the above that the model correctly predicts of the first 6 observations from the `te` st dataset, suggesting a % accuracy. Over the full observations contained in the `te` st dataset are correctly predicted, which is % accurate.

For different evaluations of the model we will collect the class predictions from the training and tuning datasets as well:

```
ds[tr, vars] %>% predict(model, newdata=., type="class") -> predict_tr
ds[tu, vars] %>% predict(model, newdata=., type="class") -> predict_tu
```

We can also calculate the accuracy for each of these datasets:

```
glue("{round(100*sum(predict_tr==actual_tr)/length(predict_tr))}%")
```

```
## 83%
```

```
glue("{round(100*sum(predict_tu==actual_tu)/length(predict_tu))}%")
```

```
## 83%
```

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

14.7 Predict Probability

20200607 To extract the probability of it raining tomorrow from the prediction made by the model. For a rpart model we can obtain the probability of it raining tomorrow using stats::predict() with *type=* `"prob"` .

```
ds[tr, vars] %>% predict(model, newdata=., type="prob") %>% '['(,2) -> pr_tr
ds[tu, vars] %>% predict(model, newdata=., type="prob") %>% '['(,2) -> pr_tu
ds[te, vars] %>% predict(model, newdata=., type="prob") %>% '['(,2) -> pr_te
```

```
head(pr_tr)
```

```
##        1        2        3        4        5        6
## 0.137937 0.137937 0.137937 0.137937 0.137937 0.137937
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

14.8 Evaluating the Model

REVIEW Before deploying a model we will want to have some measure of confidence in the predictions. This is the role of evaluation—we evaluate the performance of a model to gain an expectation of how well the model will perform on new observations.

We evaluate a model by making predictions on observations that were not used in building the model. These observations will need to have a known outcome so that we can compare the model prediction against the known outcome. This is the purpose of the `te`st dataset as explained in Section 8.12.

```
head(predict_te) == head(actual_te)
```

```
## [1] TRUE TRUE TRUE TRUE TRUE TRUE
```

```
sum(head(predict_te) == head(actual_te))
```

```
## [1] 6
```

```
sum(predict_te == actual_te)
```

```
## [1] 22177
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

14.9 Accuracy and Error Rate

From the two vectors `cl_te` and `target_te` we can calculate the overall accuracy of the predictions over the `te` dataset. This will simply be the sum of the number of times the prediction agrees with the actual class, divided by the size of the `te`st dataset (which is the same as the length of `target_te`).

```
acc_te <- sum(predict_te == actual_te, na.rm=TRUE)/length(actual_te)
round(100*acc_te, 2)
```

```
## [1] 83.65
```

Here we can see that the model has an overall accuracy of %. That is a relatively high accuracy for a typical model build.

We can also calculate the overall error rate in a similar fashion. Some Data Scientists prefer to talk in terms of the error rate rather than the accuracy:

```
err_te <- sum(predict_te != actual_te, na.rm=TRUE)/length(actual_te)
round(100*err_te, 2)
```

```
## [1] 16.35
```

Thus our decision tree model has an overall error rate of %.

Notice also that we have now twice converted a proportion (generally a number between 0 and 1) into a percentage (generally a number between 0 and 100) by multiplying the proportion by 100 and then base::round()ing it to 2 decimal places. We will no doubt want to do this regularly (if we find percentages to be more quickly accessible than proportions). This is thus a candidate for packaging up as a function. To do so we use base::function() and provide it with a single argument—the number we wish to convert to a percentage:

```
per <- function(n) { p <- round(100*n, 2); return(p) }
```

We can now use this as a convenience:

```
per(acc_te)
```

```
## [1] 83.65
```

```
per(err_te)
```

```
## [1] 16.35
```

To illustrate the more optimistic measure that we obtain when we apply our model to the `tr` aining dataset we can repeat the above calculations:

```
acc_tr <- sum(predict_tr == actual_tr, na.rm=TRUE)/length(actual_tr)
per(acc_tr)
```

```
## [1] 83.5
```

```
err_tr <- sum(predict_tr != actual_tr, na.rm=TRUE)/length(actual_tr)
per(err_tr)
```

```
## [1] 16.5
```

The overall accuracy over the `tr` aining dataset is % compared to the % accuracy calculated over the `te` dataset. The difference for this small dataset is small but we do see that the accuracy is higher on the `tr` aining dataset. Similarly the overall error rate is % on the `tr` aining dataset compared to the `te` error rate of %.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

14.10 Confusion Matrix

The accuracy and error rate are rather blunt measures of performance. They are a good starting point to get a sense of how good the model is but more is required. A confusion matrix allows us to review how well the model performs against the actual classes.

```
round(100*table(actual_te, predict_te, dnn=c("Actual", "Predicted"))
      /length(actual_te))
```

```
##       Predicted
## Actual No Yes
##    No  76   3
##    Yes 14   7
```

This is useful since the consequences of a wrong decision will be different for the different decisions. For example, if it is incorrectly predicted that it will not rain tomorrow and I decide not to carry an umbrella with me then the consequence is that I will get wet. We might experience this as a more severe consequence than the situation where it is incorrectly predicted that it will rain and so we unnecessarily carry an umbrella with us all day.

We again compare this to the performance on the `tr`aining dataset to note that model performs better, at least when predicting that it will not rain tomorrow.

```
round(100*table(actual_tr, predict_tr, dnn=c("Actual", "Predicted"))
      /length(actual_tr))
```

Notice that the false negative rate (the errors that have a higher consequence—getting uncomfortably wet if it does actually rain) is reduced from % to %. The performance as measured over the `te` dataset is likely to be more indicative of the actual model performance and the false negative rate is one that we would rather minimize.

We will be generating confusion matrices quite regularly based on the predicted classes (counting just those that are not missing using base::is.na()) and the target classes. This is another candidate for wrapping up into a function to save us having to remember the command in detail and also to save us having to type out the command each time.

```
con <- function(predicted, actual)
{
  tbl <- table(actual, predicted, dnn=c("Actual", "Predicted"))
  tbl <- round(100*tbl/sum(!is.na(predicted)))
  return(tbl)
}
```

We can simply call this function with the appropriate arguments to have the confusion matrix printed.

```
con(predict_tr, actual_tr)
```

```
##       Predicted
## Actual No Yes
##    No  76   3
##    Yes 14   7
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

14.11 ROC Chart

Another measure of the quality of a model is the ROC curve and in particular the area under the ROC curve. This area can be calculated using ROCR::prediction() and ROCR::performance() from ROCR (Sing et al. 2020). These functions use the probability of a prediction rather than the prediction of a class.

In the following code block we obtain the predicted probabilities from the model, predicting over the `te` dataset. The result from calling stats::predict() is a matrix with columns corresponding to the possible class values recording the probability of each class for each observation. The second column is the one of interest (the probability that it will rain tomorrow or `rain_tomorrow==yes`). These probabilities are passed on to ROCR::prediction() to compare them with the actual target values. The result is then passed on to ROCR::performance() from which we obtain the base::attr()ibute `y.values` and then magrittr::extract2() the first value as the area under the curve.

References

Sing, Tobias, Oliver Sander, Niko Beerenwinkel, and Thomas Lengauer. 2020. *ROCR: Visualizing the Performance of Scoring Classifiers*. http://ipa-tys.github.io/ROCR/.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

14.12 Risk Chart

A risk chart is also known as an accumulative performance plot.
Figure @ref(fig:memplate:rp_evaluate_test_riskchart) shows a rattle::riskchart() based on the `te` dataset.

```
riskchart(pr_te, actual_te, risk_te) +
  labs(title="Risk Chart - Test Dataset") +
  theme(plot.title=element_text(size=14))
```

(#fig:memplate:rp_evaluate_test_riskchart)A Risk Chart for the test dataset. The risk chart plots the population against the performance captured as the true positive rate. The area under the risk chart is another single measure for comparing performance.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

14.13 Biased Estimate from the Training Dataset

We noted above that evaluating a model on the `tr`aining dataset on which the model was built will result in overly optimistic performance outcomes. We can compare the performance of the randomForest::randomForest() model on the `te` dataset presented above with that on the `tr`aining dataset. As expected the performance on the `tr`aining dataset is wildly optimistic. In fact it is common for the randomForest (Breiman et al. 2018) model to predict perfectly over the training dataset as we see from the confusion matrix.

```
predict_tr <- predict(model, newdata=ds[tr, vars], type="class")
con(predict_tr, actual_tr)
```

```
##       Predicted
## Actual No Yes
##    No  76   3
##    Yes 14   7
```

Similarly Figure @ref(fig:memplate:rf_riskchart_tr) illustrates the problem of evaluating a model based on the `tr`aining dataset. Again we see perfect performance when we evaluate the model on the `tr`aining dataset. The performance line (the Recall which is plotted as the green line) follows the best achievable which is the grey line.

```
pr_tr <- predict(model, newdata=ds[tr, vars], type="prob")[,2]
riskchart(pr_tr, actual_tr, risk_tr, title.size=14) +
  labs(title="Risk Chart - " %s+% mtype %s+% " - Training Dataset")
```

(#fig:memplate:rf_riskchart_tr)Performance chart for randomForst over the training dataset.

References

Breiman, Leo, Adele Cutler, Andy Liaw, and Matthew Wiener. 2018. *randomForest: Breiman and Cutler's Random Forests for Classification and Regression*. https://www.stat.berkeley.edu/~breiman/RandomForests/.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

14.14 Step 8: Save the Model to File

We save the model, together with the dataset and other variables, into a binary R file.

```
mddate  <- "_" %s+% format(Sys.time(), "%Y%m%d_%H%M%S")
mdrdata <- dsname %s+% "_" %s+% mtype %s+% mddate %s+% ".RData"
save(ds, dsname,
     tr, te, actual_tr, risk_tr, actual_te, risk_te,
     per, con, model, mtype,
     file=mdrdata)
```

The dataset, its meta-data and the model are now stored in *. We can then load this later and replicate the process.

```
(load(mdrdata))
```

```
##  [1] "ds"        "dsname"    "tr"        "te"        "actual_tr" "risk_tr"
##  [7] "actual_te" "risk_te"   "per"       "con"       "model"     "mtype"
```

Note that by using generic variable names we can load different model files and perform common operations on them without changing the names within a script. However, do note that each time we load such a saved model file we overwrite any other variables of the same name.

We have also defined some additional meta-data variables relating to modelling and these will be useful to include in the data file with the dataset, separate to the saved model. We augment the original data file with the modelling meta data and save that to a new file.

```
mddate  <- "_" %s+% format(Sys.time(), "%Y%m%d_%H%M%S")
mdrdata <- dsname %s+% "_meta" %s+% mddate %s+% ".RData"
save(ds, dsname,
     tr, te, actual_tr, risk_tr, actual_te, risk_te,
     per, con,
     file=mdrdata)
```

The dataset and its meta-data are now stored on our computer in the current folder in a file with the name *.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

14.15 Boilerplate

Here in one block is the code to perform model building.

```
INSERT model.R
```

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

14.16 Command Summary

This chapter has referenced the following R packages, functions, commands, operators, and datasets: \[2mm]

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

14.17 Model Template Further Reading

Other resources include:

- The process we have resented here for modelling from data has been tuned over many years of delivering analytics and data mining projects. An early influence was CRISP-DM the CRoss Industry Standard Process for Data Mining.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

14.18 Model Template Example

TODO Template commands for model build and evaluate process end-to-end.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

15 ML Scenarios

There are three basic machine learning scenarios: reinforcement learning, supervised learning, unsupervised learning.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

15.1 Machine Learning Setup

20200514 Packages used in this chapter include magrittr (Bache and Wickham 2020), and rattle (G. Williams 2020).

Packages are loaded into the currently running R session from your local library directories on disk. Missing packages can be installed using utils::install.packages() within R. On Ubuntu, for example, R packages can also be installed using `$ wajig install r-cran-<pkgname>`.

```
# Load required packages from local library into the R session.

library(magrittr)     # Data pipelines: %>% %<>% %T>% equals().
library(rattle)       # Dataset: weather.
```

The rattle::weatherAUS dataset is loaded into the template variable `ds` and further template variables are setup as introduced by Graham J. Williams (2017). See Chapter 8 for details.

```
dsname <- "weatherAUS"
ds     <- get(dsname)

nobs   <- nrow(ds)

vnames <- names(ds)
ds    %<>% clean_names(numerals="right")
names(vnames) <- names(ds)

vars   <- names(ds)
target <- "rain_tomorrow"
vars   <- c(target, vars) %>% unique() %>% rev()
```

The variable `form` is used in this chapter as the formula describing the model to be built.

```
form
```

```
## rain_tomorrow ~ .
```

```
ds  %>% sample_frac()
```

```
## # A tibble: 176,747 x 24
##    date       location     min_temp max_temp rainfall evaporation sunshine
##    <date>     <chr>           <dbl>    <dbl>    <dbl>       <dbl>    <dbl>
##  1 2015-09-14 Nuriootpa         8.7     18.6      0           4.9      8.8
##  2 2014-02-22 Katherine        19.6     34.5      0           7.4     NA
##  3 2010-07-10 PearceRAAF        7.7     17.3     NA          NA        4.4
##  4 2019-12-10 Albury           15.3     36.3      0          NA       NA
##  5 2012-05-14 Albany           10.1     19.2      7.8         3.8      6.7
##  6 2012-03-11 MountGambier      4.5     22.3      0           2.2      6.1
##  7 2008-09-06 Melbourne         7.3     18.9      0           2.6      7.3
##  8 2017-10-20 NorahHead        17.4     19.8      4          NA       NA
##  9 2009-08-03 Melbourne         8.8     15.4      3.2         3.6      7.1
## 10 2014-04-17 MountGinini       3.2     12.4      0          NA       NA
## # … with 176,737 more rows, and 17 more variables: wind_gust_dir <ord>,
## #   wind_gust_speed <dbl>, wind_dir_9am <ord>, wind_dir_3pm <ord>,
## #   wind_speed_9am <dbl>, wind_speed_3pm <dbl>, humidity_9am <int>,
## #   humidity_3pm <int>, pressure_9am <dbl>, pressure_3pm <dbl>,
## #   cloud_9am <int>, cloud_3pm <int>, temp_9am <dbl>, temp_3pm <dbl>,
## #   rain_today <fct>, risk_mm <dbl>, rain_tomorrow <fct>
```

References

Bache, Stefan Milton, and Hadley Wickham. 2020. *Magrittr: A Forward-Pipe Operator for r*. https://CRAN.R-project.org/package=magrittr.

Williams, Graham. 2020. *Rattle: Graphical User Interface for Data Science in r*. https://rattle.togaware.com/.

Williams, Graham J. 2017. *The Essentials of Data Science: Knowledge Discovery Using r*. The r Series. CRC Press.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

15.2 Reinforcement Learning

Reinforcement learning algorithms use a goal-oriented approach to learn with a focus on maximising a complex goal or cumulative reward.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

15.3 Supervised Learning

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

15.4 Unsupervised Learning

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

16 ML Activities

Machine learning can be applied for many different generic activities. we describe many of those activities here.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

16.1 Classification

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

16.2 Cluster Analysis

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

16.3 Outlier Detection

A useful algorithm for outlier detection is the one-class support vector machine (Section @ref(sec:one_class_cupport_vector_machine)).

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

16.4 Prediction

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

16.5 Text Mining

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

17 ML Applications

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

17.1 Airport Gate Assignment

The task here is to plan and the to update the assignment of gates at an airport to aircraft using real-time data and a collection of often quite complex constraints and preferences.

Algorithms used for this task include reinforcement learning and neural networks.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

17.2 Article Summarisation

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

17.3 Facial Recognition

20210405 Recognising faces in a photo or a video stream is a common task for computer vision. The identification of the people within the photos/video is also regularly utilised.

This technology can be and is variously used for the benefit of society, to improve processing of long queues at immigration at airports, for securing data centres, for search and rescue. However, it can also be used for the oppression of our freedoms. It is commonly reported that it has been used by governments to monitor, oppress, and assimilate, whole populations of religious and ethnic groups. Indeed, collaborative research has been terminated by more open governments where that research was in collaboration with more oppressive governments.

Deep neural network facial recognition technology is also particularly sensitive to the training data, meaning that it does not generalise well. The pre-built models have variously been found to be poor identifying emotion or age or agenda for specific racial populations that might be under-represented in the training data. This can result in unexpected and widespread automated racial discrimination.

Resources

- Explore facial recognition yourself:
  - MLHub Azure Face
- Popular Press
  - Gender Shades
  - Don’t look now: why you should be worried about machines reading your emotions
  - This is how we lost control of our faces
- Research Papers
  - About Face: A survey of facial recognition evaluation

We need to be very careful in the deployment of this and other AI technology.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open*

source software since 1984.

Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

17.4 Gene Detection

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

17.5 Group Recommendations

Group recommendation provides recommendations for a group or community rather than an individual. Examples include group study, friends watchig videos together and families travelling.

Recommendations can be provided to small user groups or to much larger online sharing communities.

A typical approach for large group recommendation identifies group interests and connections between users within groups. Different interest subgroups are identified through closely connected users with similar interests. Collaborative filtering can then be used for the subgroups. These recommendations are then aggregated in some way to the provide the group recommendations.

See Qin et al. (2020).

References

Qin, D., X. Zhou, L. Chen, G. Huang, and Y. Zhang. 2020. “Dynamic Connection-Based Social Group Recommendation.” *IEEE Transactions on Knowledge and Data Engineering* 32 (3): 453–67.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

17.6 Program Comprehension

Within software engineering the task of comprehending programs can be a challenge.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

17.7 Pull Request Generation

20200525 AI can be applied to software engineering activities and practices. For example, the chances of a pull request being accepted are increased with good descriptions.

Algorithms used for this task include nearest neighbours.

See (Liu et al. (2019)).

References

Liu, Zhongxin, Xin Xia, Christoph Treude, David Lo, and Shanping Li. 2019. “Automatic Generation of Pull Request Descriptions.” *2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, November. https://doi.org/10.1109/ase.2019.00026.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

17.8 Reservoir Inflow

Reservoirs collect and store water, most obviously for human consumption, but also for irrigation, generation of electricity, and also for industrial use. Reservoirs need to be carefully managed in order to ensure the supply of water under many scenarios.

By predicting water inflow, for example, the balance between maintaining water in the reservoir and releasing water to avoid overflow can be monitored. At times of peak flow, for example, the reservoir needs to be emptied before the flow arrives as outflow is often slower than inflow. On the other hand the reservoir needs to be maintained as full as possible for hydro electricity generation. Forecasting inflow is thus a crucial element to the management of a reservoir.

Algorithms used for this task include neural networks.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

17.9 Sentiment Analysis

Sentiment analysis is the process of analysing text to determine the sentiment being expressed.

Algorithms used for this task includes neural networks.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

17.10 Topic Modelling

Algorithms include bag-of-words, TF-IDF, phrase-mining, Latent Dirichelet Allocation.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

18 ML Algorithms

20210103 We can think of algorithms for machine learning along three basic dimensions:

- knowledge representation (the language used to represent models);
- method or search heuristic (how to search different models);
- measure of goodness (how do we know we have a good model).

This basic concept was introduced in (Graham J. Williams 2011) characterising different artificial intelligence and machine learning algorithms in terms of the target language for representing knowledge, how the search space defined by the language is navigated to express sentences in the language, and how the sentences are measured to determine whether we have a good sentence.

In this chapter we present a range of machine learning algorithms, relating them to these dimensions, and demonstrating the algorithms in action.

References

Williams, Graham J. 2011. *Data Mining with Rattle and R: The Art of Excavating Data for Knowledge Discovery.* Use R! New York: Springer.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

18.1 Algorithms Setup

20210103 Packages used in this chapter include nnet (Ripley 2021), rpart (Therneau and Atkinson 2019), and rattle (G. Williams 2020).

Packages are loaded into the currently running R session from your local library directories on disk. Missing packages can be installed using utils::install.packages() within R. On Ubuntu, for example, R packages can also be installed using `$ wajig install r-cran-<pkgname>` .

```
# Load required packages from local library into the R session.

library(nnet)
library(rpart)        # ML: decision tree rpart().
```

References

Ripley, Brian. 2021. *Nnet: Feed-Forward Neural Networks and Multinomial Log-Linear Models*. http://www.stats.ox.ac.uk/pub/MASS4/.

Therneau, Terry, and Beth Atkinson. 2019. *Rpart: Recursive Partitioning and Regression Trees*. https://CRAN.R-project.org/package=rpart.

Williams, Graham. 2020. *Rattle: Graphical User Interface for Data Science in r*. https://rattle.togaware.com/.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*

Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

18.2 Algorithms Data and Variables

20210103

The rattle::weatherAUS dataset is loaded into the template variable `ds` and further template variables are setup as introduced by Graham J. Williams (2017). See Chapter 8 for details.

```
dsname <- "weatherAUS"
ds     <- get(dsname)

nobs   <- nrow(ds)

vnames <- names(ds)
ds    %<>% clean_names(numerals="right")
names(vnames) <- names(ds)

vars   <- names(ds)
target <- "rain_tomorrow"
vars   <- c(target, vars) %>% unique() %>% rev()
```

For the rattle::weatherAUS dataset we similarly define the following template variables (Graham J. Williams 2017) used for predictive modelling. See Chapter 8 for details.

```
risk   <- "risk_mm"
id     <- c("date", "location")
ignore <- c(risk, id)
vars   <- setdiff(vars, ignore)
inputs <- setdiff(vars, target)

form   <- formula(target %s+% " ~ .")

ds[vars] %<>% na.roughfix()

SPLIT <- c(0.70, 0.15, 0.15)

nobs %>% sample(SPLIT[1]*nobs)                                -> tr
nobs %>% seq_len() %>% setdiff(tr) %>% sample(SPLIT[2]*nobs) -> tu
nobs %>% seq_len() %>% setdiff(tr) %>% setdiff(tu)           -> te

ds %>% slice(tr) %>% pull(target) -> actual_tr
ds %>% slice(tu) %>% pull(target) -> actual_tu
ds %>% slice(te) %>% pull(target) -> actual_te

ds %>% slice(tr) %>% pull(risk) -> risk_tr
ds %>% slice(tu) %>% pull(risk) -> risk_tu
ds %>% slice(te) %>% pull(risk) -> risk_te
```

References

Williams, Graham J. 2017. *The Essentials of Data Science: Knowledge Discovery Using r*. The r Series. CRC Press.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

18.3 Algorithms Data Review

20210103 We can review a random sample of the dataset.

```
ds  %>% sample_frac()
```

```
## # A tibble: 176,747 x 24
##    date       location      min_temp max_temp rainfall evaporation sunshine
##    <date>     <chr>            <dbl>    <dbl>    <dbl>       <dbl>    <dbl>
##  1 2014-12-04 Walpole           12.8     20.3      1.4         4.8      8.5
##  2 2018-10-31 CoffsHarbour      16.3     28.3      0           4.8      8.5
##  3 2018-04-25 NorfolkIsland     17.8     22.2     38.6         5.2      8.5
##  4 2018-06-13 PerthAirport       6.5     17.7      6.2         1.6      1.6
##  5 2019-10-07 Townsville        16.5     29        0           8        8.5
##  6 2009-04-08 Nuriootpa          9.9     21.8      0           4        9.7
##  7 2020-04-08 Perth             17.1     32.6      0           7       10.5
##  8 2019-06-11 Albany             8.1     14.2      4.6         2.6      8.5
##  9 2014-05-13 Ballarat           3.3     15.6      0           4.8      8.5
## 10 2019-06-26 CoffsHarbour      13.2     17.2     45.6         4.8      8.5
## # … with 176,737 more rows, and 17 more variables: wind_gust_dir <ord>,
## #   wind_gust_speed <dbl>, wind_dir_9am <ord>, wind_dir_3pm <ord>,
## #   wind_speed_9am <dbl>, wind_speed_3pm <dbl>, humidity_9am <dbl>,
## #   humidity_3pm <dbl>, pressure_9am <dbl>, pressure_3pm <dbl>,
## #   cloud_9am <dbl>, cloud_3pm <dbl>, temp_9am <dbl>, temp_3pm <dbl>,
## #   rain_today <fct>, risk_mm <dbl>, rain_tomorrow <fct>
```

And glimpse all of the variables.

```
glimpse(ds)
```

```
## Rows: 176,747
## Columns: 24
## $ date            <date> 2008-12-01, 2008-12-02, 2008-12-03, 2008-12-04, 2008-…
## $ location        <chr> "Albury", "Albury", "Albury", "Albury", "Albury", "Alb…
## $ min_temp        <dbl> 13.4, 7.4, 12.9, 9.2, 17.5, 14.6, 14.3, 7.7, 9.7, 13.1…
## $ max_temp        <dbl> 22.9, 25.1, 25.7, 28.0, 32.3, 29.7, 25.0, 26.7, 31.9, …
## $ rainfall        <dbl> 0.6, 0.0, 0.0, 0.0, 1.0, 0.2, 0.0, 0.0, 0.0, 1.4, 0.0,…
## $ evaporation     <dbl> 4.8, 4.8, 4.8, 4.8, 4.8, 4.8, 4.8, 4.8, 4.8, 4.8, 4.8,…
## $ sunshine        <dbl> 8.5, 8.5, 8.5, 8.5, 8.5, 8.5, 8.5, 8.5, 8.5, 8.5, 8.5,…
## $ wind_gust_dir   <ord> W, WNW, WSW, NE, W, WNW, W, W, NNW, W, N, NNE, W, SW, …
## $ wind_gust_speed <dbl> 44, 44, 46, 24, 41, 56, 50, 35, 80, 28, 30, 31, 61, 44…
## $ wind_dir_9am    <ord> W, NNW, W, SE, ENE, W, SW, SSE, SE, S, SSE, NE, NNW, W…
## $ wind_dir_3pm    <ord> WNW, WSW, WSW, E, NW, W, W, W, NW, SSE, ESE, ENE, NNW,…
## $ wind_speed_9am  <dbl> 20, 4, 19, 11, 7, 19, 20, 6, 7, 15, 17, 15, 28, 24, 4,…
## $ wind_speed_3pm  <dbl> 24, 22, 26, 9, 20, 24, 24, 17, 28, 11, 6, 13, 28, 20, …
## $ humidity_9am    <dbl> 71, 44, 38, 45, 82, 55, 49, 48, 42, 58, 48, 89, 76, 65…
## $ humidity_3pm    <dbl> 22, 25, 30, 16, 33, 23, 19, 19, 9, 27, 22, 91, 93, 43,…
## $ pressure_9am    <dbl> 1007.7, 1010.6, 1007.6, 1017.6, 1010.8, 1009.2, 1009.6…
## $ pressure_3pm    <dbl> 1007.1, 1007.8, 1008.7, 1012.8, 1006.0, 1005.4, 1008.2…
## $ cloud_9am       <dbl> 8, 5, 5, 5, 7, 5, 1, 5, 5, 5, 5, 8, 8, 5, 5, 0, 8, 8, …
## $ cloud_3pm       <dbl> 5, 5, 2, 5, 8, 5, 5, 5, 5, 5, 5, 8, 8, 7, 5, 5, 1, 1, …
## $ temp_9am        <dbl> 16.9, 17.2, 21.0, 18.1, 17.8, 20.6, 18.1, 16.3, 18.3, …
## $ temp_3pm        <dbl> 21.8, 24.3, 23.2, 26.5, 29.7, 28.9, 24.6, 25.5, 30.2, …
## $ rain_today      <fct> No, No, No, No, No, No, No, No, No, Yes, No, Yes, Yes,…
## $ risk_mm         <dbl> 0.0, 0.0, 0.0, 1.0, 0.2, 0.0, 0.0, 0.0, 1.4, 0.0, 2.2,…
## $ rain_tomorrow   <fct> No, No, No, No, No, No, No, No, Yes, No, Yes, Yes, Yes…
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

18.4 Collaborative Filtering

Representation:

Method:

Measure:

Collaborative filtering is used in recommendation systems. It generates recommendations based on ratings of items by users.

The typical algorithm represents users and items in a matrix and records the ratings in the cells of the matrix. The aim is to match a new user's preferences for items with other existing users' preferences. The similarity measures used for matching include cosine similarity, Pearson correlation, and probability-based similarity. For the most similar users any items that the new user has not utilised but have been utilised by matching users, will be recommended.

Model-based collaborative filtering trains a predictive model from user-item ratings matrix. Modellers include Bayesian, latent semantic model, and SVD. Typically the training data consists of user-item pairs and the rating.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

18.5 Convolutional Neural Network CNN

Representation	Method	Measure

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

18.6 Decision Trees

20210103

Representation	Method	Measure
Tree	Recursive Partitioning	Information Gain

To build a decision tree we typically use rpart::rpart().

```
mtype <- "rpart"
mdesc <- "decision tree"

ds %>%
  select(all_of(vars)) %>%
  slice(tr) %>%
  rpart(form, ., method="class", control=rpart.control(maxdepth=3)) %T>%
  print() ->
model
```

```
## n= 123722
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##  1) root 123722 25822 No (0.7912902 0.2087098)
##    2) humidity_3pm< 72.5 105610 14774 No (0.8601079 0.1398921) *
##    3) humidity_3pm>=72.5 18112  7064 Yes (0.3900177 0.6099823)
##      6) humidity_3pm< 82.5 9469  4412 No (0.5340585 0.4659415)
##       12) rainfall< 0.85 5291  1909 No (0.6391986 0.3608014) *
##       13) rainfall>=0.85 4178  1675 Yes (0.4009095 0.5990905) *
##      7) humidity_3pm>=82.5 8643  2007 Yes (0.2322110 0.7677890) *
```

Chapter 20 covers decision trees in detail whilst Chapter 14 uses decision trees as the model builder to demonstrate the model template. Examples of decision tree induction are available through the rain, iris, and pyiris packages from MLHub.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

18.7 Deep Convolutional Generative Adversarial Network

REVIEW A generative adversarial network architecture which

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

18.8 Generative Adversarial Network

REVIEW Train two models simultaneously. An adversarial approach is taken. The **generator** process (e.g., *the artist*) will learn to create images that look real, and over time gets better at creating real looking images. A **discriminator** (e.g., *the art critic*) learns to identify real images from the fakes. The learning stops when the discriminator can no longer distinguish between the real and fake images.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

18.9 K Means Clustering

Representation	Method	Measure
Cluster indicies	Recursive cluster centroid assignment	Within and between cluster distances

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

18.10 K Nearest Neighbours

Representation	Method	Measure

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

18.11 Linear Regression

Representation	Method	Measure

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

18.12 Logistic Regression

Representation	Method	Measure

```
glm <- glm(v ~ g, data=ds, family="binomial")
summary(glm)
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

18.13 Long Short-Term Memory Neural Networks LSTM

Representation	Method	Measure

A long short-term memory neural network is a recurrent neural network which uses additional storage in the architecture to keep temporary information (short-term memory). Stored data will be forgotten through iteration. The efficiency of feature extraction is improved and computational time is reduced.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

18.14 Multi Layer Perceptron

Representation	Method	Measure

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

18.15 Neural Networks

Resources include:

- Ripley: http://books.google.com.au/books?id=m12UR8QmLqoC&printsec=frontcover&dq=ripley+neural+networks&ei=pXd_SvDGEoj6lQTrt-2TCg#v=onepage&q=&f=false
- http://www-ccs.ucsd.edu/matlab/pdf_doc/nnet/nnet.pdf
- http://en.wikipedia.org/wiki/Backpropagation
- http://www.idiap.ch/~bengio/lectures/
- http://books.google.com.au/books?id=Czp1N5UWpyEC&pg=PA91&lpg=PA91&dq=skip-layer+nnet&source=bl&ots=qDQNIWZMiI&sig=Mhoxr7f-dQFeNl2u7F8RN1xVn94&hl=en&ei=VLt7SvzCB4TY7AOntYBR&sa=X&oi=book_result&ct=result&resnum=8#v=onepage&q=skip-layer nnet&f=false

Representation	Method	Measure
Multi-Layered Connected Network	Back Propagation	

Artificial neural networks are inspired by biological neural networks as found in the human brain with supposedly over 60 billion neurons connected by synapses. They are often called artificial neural networks to distinguish them from the natural kind. In a natural neural network electrical impulses are transmitted from neuron to neuron. In an artificial neural network numbers are transmitted from one node to the next node.

The neurons (nodes) are organised as interconnected layers. A feed forward neural network, for example, has three layers: the input, a hidden, and an output layer. The neurons act as mathematical nodes that perform some maths on the input and if the combined *signal* is strong enough a number is sent out from the node into the network. The particular maths needs to be learned to ensure the final output of the neural network matches the expected output. The algorithm for learning (training) a neural network modifies the calculations to achieve the desired output from the output layer, using back propagation.

Numeric input data is expected, often assumed to be scaled to be between 0 and 1. The input data (as numbers) are propagated through the nodes of the network, being multiplied by a weight associated with each link between the nodes. At each node all of the input links are summed together with another number added to the mix for each node. This number is then fed into a sigmoid function (an S shaped curve) as the final output from that node.

The final node is the output node which calculates the decision.

As a classification model the number that is output through the output nodes can be interpreted as the probability of a particular class. Neural network algorithms can be used for regression or classification tasks.

The nnet (Ripley 2021) package provides algorithms for feed-forward neural networks with a single hidden layer, and for multinomial log-linear models.

```
library(nnet)
?nnet
```

Options include `linout=` to predict a linear rather than a logistic value, `rang=` is the range for the initial random weights from -rang to rang with 0.5 suggested, `decay=` used to change the default decay parameter of 0, `skip=` to turn off the use of a *skip-layer* where input variables are connected directly to the output variable. The skip-layer results in the model being a perturbation of a linear hyperplane where we can think of the direct links (i.e., the skip-layer) representing a plane and the non-skip links representing a perturbation to the plane Velten (2009).

References

Ripley, Brian. 2021. *Nnet: Feed-Forward Neural Networks and Multinomial Log-Linear Models*. http://www.stats.ox.ac.uk/pub/MASS4/.
Velten, Kai. 2009. *Mathematical Modeling and Simulation*. Wiley.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

18.16 Naive Bayes

Representation	Method	Measure
Probabilities		

Consider classifying input data by determining the probability of an event happening given the probability of another event that has already occured. Class and conditional probabilities are calculated to determine the likelihood of an event.

Na"ive Bayes is a simple and effective machine learning classifier based on Bayes' Theorem:

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)}$$

The class probabilities are $P(A)$ (the probability of event A) and $P(B)$. The conditional probabilities are $P(A|B)$ (the probability of A happening given that B occurs) and $P(B|A)$.

A Na"ive Bayes classifier uses these probabilities to classify independent features that have the same weight or importance from the input dataset to make predictions.

The idea is simple and can be applied to small datasets. It suffers from the zero frequency problem where a class with probability of 0 for a selected feature, and hence its conditional probability is then also 0 and so excluded from further consideration. Laplace smoothing assigns the class a small probability.

The concept also assumes independence of the features and when not met will perform poorly.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

18.17 One-Class Support Vector Machine

Representation	Method	Measure
Bounding Support Vectors		

A support vector machine (Section 18.20) identifies a linear separator between observations that belong to different classes. If the observations are not tagged with a class then the algorithm can be utilised to identify a bounding region for the training dataset. The model is then represented as the boundary support vectors.

No all observations from the training dataset will sit within the boundary. That is, not all observations will fit the model. Such observations we might consider as outliers.

% EXAMPLE USING CANBERRA WEATHER AND FIND THOSE THAT ARE VERY % DIFFERENT TO CANBERRA...

Explore one-class support vector machines through the ocsvm package from MLHub.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

18.18 Recurrent Neural Networks

Representation	Method	Measure

A recurrent neural network is a neural network which records information recursively between layers. An RNN is useful for predicting time series. Problems with the approach include gradient vanish or gradient explosion. Long short-term memory neural networks address these problems.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

18.19 Residual Neural Network

Representation	Method	Measure

Explore ResNet through the cars package from MLHub.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

18.20 Support Vector Machine

Representation	Method	Measure
Boundary Support Vectors		

A support vector machine identifies a hyperplane (or a line in 2D space) which linearly separates the observations that belong to different classes. Those observations that are closest to the hyperplane are called the support vectors. The concept of a margin is used to maximise the distance between the support vectors.

Data is not usually linearly separable. Thus some kind of mapping of the data to higher dimensions is performed to find a space where it is possible to separate the classes with a hyperplane. A so-called kernel function is used to do this. Common kernel functions include radial basis function, linear, polynomial, and sigmoid.

See Section 18.17 for the one-class support vector machine useful for outlier detection.

% Free and Open Source Data Science Software Survival Guide % % Time-stamp: <Saturday 2021-03-13 12:45:41 AEDT Graham Williams> % % Author: Graham.Williams@togaware.com % License: Creative Commons Attribution License 4.0 International % License: https://creativecommons.org/licenses/by/4.0/ % Copyright: (c) 1995-2021 Graham.Williams@togaware.com % % You are free to Share (copy and redistribute the material in any % medium or format) and Adapt (remix, transform, and build upon the % material for any purpose, even commercially), under specific terms % of Attribution (you must give appropriate credit, provide a link to % the license, and indicate if changes were made and you may do so in % any reasonable manner, but not in any way that suggests the licensor % endorses you or your use) and with No Additional Restrictions (you % may not apply legal terms or technological measures that legally % restrict others from doing anything the license permits).

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open*

source software since 1984.

Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

19 Cluster Analysis

{#cha:clustering}

20200902 Cluster analysis(or clustering) is widely used in data mining to identify groups of similar data. It is well supported in R (R Core Team 2021) with many packages available for preparing for cluster analysis, identifying a good number of clusters, performing a clustering, and evaluating the clustering. A variety of cluster analysis algorithms are available, each generating a cluster index for each data item, as the representation of the clustering. The measure of performance often involves measuring the distances of points withion a cluster and between clusters.

We have briefly introduced the KMeans clustering algorithm in decision trees as an algorithm in Section **??**.

References

R Core Team. 2021. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

19.1 Clustering Setup

20200902 Packages used in this chapter include biclust (Kaiser et al. 2021).

```
# Load required packages from local library into the R session.

library(biclust)      # Bicluster analysis.
library(dplyr)        # Wrangling: glimpse() group_by() print() select() mutate().
library(rattle)       # Weather dataset.
```

The rattle::weatherAUS dataset is loaded into the template variable `ds` and further template variables are setup as introduced by Graham J. Williams (2017). See Chapter 8 for details.

```
dsname <- "weatherAUS"
ds     <- get(dsname)

nobs   <- nrow(ds)

vnames <- names(ds)
ds    %<>% clean_names(numerals="right")
names(vnames) <- names(ds)

vars   <- names(ds)
target <- "rain_tomorrow"
vars   <- c(target, vars) %>% unique() %>% rev()
```

It is always useful to remind ourselves of the dataset with a random sample:

```
ds  %>% sample_frac() %>% select(date, location, sample(3:length(vars), 5))
```

```
## # A tibble: 176,747 x 7
##    date       location  risk_mm pressure_3pm pressure_9am rainfall wind_gust_dir
##    <date>     <chr>       <dbl>        <dbl>        <dbl>    <dbl> <ord>
##  1 2013-06-02 Penrith      13.4          NA           NA       8.4 SSW
##  2 2016-01-10 Badgerys…     0          1016.        1020.      0   ESE
##  3 2012-04-09 Nuriootpa     0          1027.        1027.      1   SW
##  4 2019-11-16 Cobar         0          1014         1017       0   SW
##  5 2008-10-02 Melbourne     0          1011.        1014.      0   N
##  6 2013-05-20 Wollongo…     0          1014.        1018.      0   WSW
##  7 2014-07-07 SalmonGu…    14            NA           NA       1.4 NW
##  8 2011-11-11 Bendigo       0          1020.        1024.      0   WNW
##  9 2019-01-11 Woomera       0          1008.        1011.      0   N
## 10 2020-02-01 Tuggeran…     0          1008.        1011.      0   W
## # … with 176,737 more rows
```

References

Kaiser, Sebastian, Rodrigo Santamaria, Tatsiana Khamiakova, Martin Sill, Roberto Theron, Luis Quintales, Friedrich Leisch, Ewoud De Troyer, and Sami Leon. 2021. *Biclust: BiCluster Algorithms*. https://CRAN.R-project.org/package=biclust.

Williams, Graham J. 2017. *The Essentials of Data Science: Knowledge Discovery Using r*. The r Series. CRC Press.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*

Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

19.2 Biclustering

```r
library(biclust)
tds <- matrix(rbinom(400, 50, 0.4), 20, 20)
res <- biclust(tds, method=BCCC(), delta=1.5, alpha=1, number=10)
res
```

```
##
## An object of class Biclust
##
## call:
##  biclust(x = tds, method = BCCC(), delta = 1.5, alpha = 1, number = 10)
##
## Number of Clusters found:  4
##
## First  4  Cluster sizes:
##                     BC 1 BC 2 BC 3 BC 4
## Number of Rows:       6    4    6    3
## Number of Columns:    5    6    4    6
```

```r
bicluster(tds, res)
```

```
## $Bicluster1
##      [,1] [,2] [,3] [,4] [,5]
## [1,]   20   22   19   21   24
## [2,]   16   20   17   20   22
## [3,]   16   20   17   17   17
## [4,]   19   22   19   22   20
## [5,]   19   23   21   20   20
## [6,]   20   23   24   22   24
##
## $Bicluster2
##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,]   19   19   18   23   19   19
## [2,]   22   22   18   24   24   19
## [3,]   21   19   19   20   22   21
## [4,]   18   21   17   19   19   18
##
## $Bicluster3
##      [,1] [,2] [,3] [,4]
## [1,]   19   19   19   21
## [2,]   24   23   25   26
## [3,]   18   16   19   24
## [4,]   23   21   26   26
## [5,]   18   18   21   21
## [6,]   18   17   22   22
##
## $Bicluster4
##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,]   21   18   15   24   15   19
## [2,]   23   21   14   24   15   19
## [3,]   24   19   18   22   18   19
```

```
parallelCoordinates(tds, res, number=4)
```

```
data(BicatYeast)
tds <- discretize(BicatYeast)
res <- biclust(tds, method=BCXmotifs(), alpha=0.05, number=50)
res
```

```
##
## An object of class Biclust
##
## call:
##  biclust(x = tds, method = BCXmotifs(), alpha = 0.05, number = 50)
##
## Number of Clusters found:  17
##
## First  5  Cluster sizes:
##                     BC 1 BC 2 BC 3 BC 4 BC 5
## Number of Rows:      172   99   45   34    9
## Number of Columns:     6    6    6    8   11
```

```
parallelCoordinates(BicatYeast, res, number=4)
```

```r
plotclust(res, tds)
```

```r
tds <- tribble(~x, ~y,
               1, 1,
               2, 1,
               1, 0,
               4, 7,
               3, 5,
               3, 6)

res <- biclust(as.matrix(tds), method=BCCC(), delta=50, alpha=0, number=5)
res
```

```
##
## An object of class Biclust
##
## call:
##  biclust(x = as.matrix(tds), method = BCCC(), delta = 50, alpha = 0,
##      number = 5)
##
## There was one cluster found with
##   6 Rows and  2 columns
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

19.3 References

- https://en.wikibooks.org/wiki/Data_Mining_Algorithms_In_R/Clustering/Biclust

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20 Decision Trees

20200815 Decision trees are widely used in data mining and well supported in R (R Core Team 2021). Decision tree learning deploys a divide and conquer approach, known as recursive partitioning. It is usually implemented as a greedy search using information gain or the Gini index to select the best input variable on which to partition the dataset at each step.

Here we introduce (G. Williams 2020) and (Therneau and Atkinson 2019) for building decision trees. We begin with a step-by-step example of building a decision tree using Rattle, and then illustrate the process using R beginning with Section **??**. We cover both classification trees and regression trees.

We have briefly introduced decision trees as an algorithm in Section @ref(sec:algorithms_dtrees) and Chapter @ref(cha:model_template) uses decision trees as the model builder to demonstrate the model template. Examples of decision tree induction are available through the rain, iris, and pyiris packages from MLHub.

References

R Core Team. 2021. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Therneau, Terry, and Beth Atkinson. 2019. *Rpart: Recursive Partitioning and Regression Trees*. https://CRAN.R-project.org/package=rpart.

Williams, Graham. 2020. *Rattle: Graphical User Interface for Data Science in r*. https://rattle.togaware.com/.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*

Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.1 Decision Trees Setup

20180603 Packages used in this chapter include C50 (Kuhn and Quinlan 2021), RWeka (Hornik 2020), party (Hothorn et al. 2021), partykit (Hothorn and Zeileis 2021), rpart (Therneau and Atkinson 2019), rpart.plot (Milborrow 2020), and rattle (G. Williams 2020).

Packages are loaded into the currently running R session from your local library directories on disk. Missing packages can be installed using utils::install.packages() within R. On Ubuntu, for example, R packages can also be installed using `$ wajig install r-cran-<pkgname>`.

```
# Load required packages from local library into the R session.

library(C50)          # Original C5.0 implementation.
library(RWeka)        # Weka decision tree J48.
library(party)        # Conditional decision trees ctree().
library(partykit)     # Convert rpart object to BinaryTree
library(rattle)       # GUI for building trees and fancy tree plot.
library(rpart)        # Popular decision tree algorithm.
library(rpart.plot)   # Enhanced tree plots.
```

The rattle::weatherAUS dataset is loaded into the template variable `ds` and further template variables are setup as introduced by Graham J. Williams (2017). See Chapter 8 for details.

```r
dsname <- "weatherAUS"
ds     <- get(dsname)

nobs   <- nrow(ds)

vnames <- names(ds)
ds    %<>% clean_names(numerals="right")
names(vnames) <- names(ds)

vars   <- names(ds)
target <- "rain_tomorrow"
vars   <- c(target, vars) %>% unique() %>% rev()
```

It is always useful to remind ourselves of the dataset with a random sample:

```r
ds  %>% sample_frac() %>% select(date, location, sample(3:length(vars), 5))
```

```
## # A tibble: 176,747 x 7
##    date       location      max_temp rain_today humidity_9am wind_dir_9am
##    <date>     <chr>            <dbl> <fct>             <int> <ord>
##  1 2010-09-03 Adelaide          18.3 No                   67 ENE
##  2 2010-10-30 WaggaWagga        19.3 No                   81 NE
##  3 2017-04-04 GoldCoast         25.5 No                   65 S
##  4 2018-10-28 Witchcliffe       24.7 No                   79 SE
##  5 2018-02-12 Witchcliffe       26.1 No                   59 SE
##  6 2009-03-10 Cairns            31.3 No                   88 W
##  7 2010-04-28 Portland          16.4 No                   89 NW
##  8 2009-10-22 Mildura           29.3 No                   49 SE
##  9 2015-04-26 BadgerysCreek     20   Yes                  76 WSW
## 10 2014-10-19 Sale              28.4 No                   66 NW
## # … with 176,737 more rows, and 1 more variable: wind_gust_dir <ord>
```

References

Hornik, Kurt. 2020. *RWeka: R/Weka Interface*. https://CRAN.R-project.org/package=RWeka.

Hothorn, Torsten, Kurt Hornik, Carolin Strobl, and Achim Zeileis. 2021. *Party: A Laboratory for Recursive Partytioning*. http://party.R-forge.R-project.org.

Hothorn, Torsten, and Achim Zeileis. 2021. *Partykit: A Toolkit for Recursive Partytioning*. http://partykit.r-forge.r-project.org/partykit/.

Kuhn, Max, and Ross Quinlan. 2021. *C50: C5.0 Decision Trees and Rule-Based Models*. https://topepo.github.io/C5.0/.

Milborrow, Stephen. 2020. *Rpart.plot: Plot Rpart Models: An Enhanced Version of Plot.rpart*. http://www.milbo.org/rpart-plot/index.html.

Therneau, Terry, and Beth Atkinson. 2019. *Rpart: Recursive Partitioning and Regression Trees*. https://CRAN.R-project.org/package=rpart.

Williams, Graham. 2020. *Rattle: Graphical User Interface for Data Science in r*. https://rattle.togaware.com/.

Williams, Graham J. 2017. *The Essentials of Data Science: Knowledge Discovery Using r*. The r Series. CRC Press.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*

Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.2 Decision Trees Modelling Setup

20200815

For the rattle::weatherAUS dataset we similarly define the following template variables (Graham J. Williams 2017) used for predictive modelling. See Chapter 8 for details.

```
risk   <- "risk_mm"
id     <- c("date", "location")
ignore <- c(risk, id)
vars   <- setdiff(vars, ignore)
inputs <- setdiff(vars, target)

form   <- formula(target %s+% " ~ .")

ds[vars] %<>% na.roughfix()

SPLIT <- c(0.70, 0.15, 0.15)

nobs %>% sample(SPLIT[1]*nobs)                                -> tr
nobs %>% seq_len() %>% setdiff(tr) %>% sample(SPLIT[2]*nobs) -> tu
nobs %>% seq_len() %>% setdiff(tr) %>% setdiff(tu)           -> te

ds %>% slice(tr) %>% pull(target) -> actual_tr
ds %>% slice(tu) %>% pull(target) -> actual_tu
ds %>% slice(te) %>% pull(target) -> actual_te

ds %>% slice(tr) %>% pull(risk) -> risk_tr
ds %>% slice(tu) %>% pull(risk) -> risk_tu
ds %>% slice(te) %>% pull(risk) -> risk_te
```

The 176,747 observations from the dataset have been randomly partitioned into a *training* dataset with 123,722 observations, a *tuning* dataset with 26,512 observations, and a *testing* dataset with 26,513 observations. The target variable (rain_tomorrow) has the classes: No (139670), Yes (37077).

References

Williams, Graham J. 2017. *The Essentials of Data Science: Knowledge Discovery Using r*. The r Series. CRC Press.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.3 Rattle Startup

After starting R (perhaps via RStudio) we can start up (G. Williams 2020), a graphical user interface for the data scientist, from the R Console prompt. Simply load the package then invoke the rattle::rattle() command, as in:

```
library(rattle)
rattle()
```

The above Rattle window will display, ready to load data, explore and test the data, transform the data, and then begin building models from the data.

The interface aims to be simple and easy to use. The row of buttons support the basic operations, particularly the Execute button which we click so that the particular tab setting take effect or its associated operation is performed. Below the buttons are a series of tabs that represent the major steps of a data science project. In general we work through the tabs from the left to right, loading some data, exploring and testing the data, performing some transformations on the data, doing a cluster and/or association analysis, building a predictive model, and evaluating the performance of the model.

After setting up the requirements and settings for a particular tab, be sure to click the Execute button to have the settings within tab take effect, or else perform the task associated with the tab.

References

Williams, Graham. 2020. *Rattle: Graphical User Interface for Data Science in r*. https://rattle.togaware.com/.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.4 Rattle Weather Dataset

Rattle provides a number of sample datasets. We can easily load them into Rattle. By default, Rattle will load the weather dataset.

We load the dataset in two simple steps

- Click on the Execute button and an example dataset is offered.
- Click on Yes to load the weather dataset.

We can use this dataset for predictive modelling to predict if it might rain tomorrow (aka statistical classification and supervised learning), or to predict how much rain we might get tomorrow (aka regression analysis).

The dataset itself is quite small, consisting of observations of some variables. Rattle also provides a much larger weather dataset covering all of Australia. The dataset has from %$ weather stattions, representing daily observations over many years.

We use the smaller dataset here for illustration, noting that typically, as data scientists, we often analyse significantly larger datasets.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.5 Rattle Summary of Dataset

The dataset from consists of daily observations of various weather related data over one year at one location (Canberra Airport). Each observation has a date and location. These are the *id* variables for this dataset.

The observations include the temperature during the day, humidity, the number of hours of sunshine, wind speed and direction, and the amount of evaporation. These are the *input* variables for this dataset.

Together with each day's observations we record whether it rains the following day and how much rain was received. These will be the *target* variables for this dataset.

Scroll through the list of variables to notice that default roles have been assigned to each of the variables. Rattle deploys some heuristics to decide on the roles, particularly the target and risk variables.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.6 Rattle Model Tab

We can click directly on the Model tab, skipping over the Explore, Test, Transform, Cluster and Associate tabs, to display the modelling options. The default modelling option is to build a decision tree, and it's always a good place to start.

Various options to tune the building of a decision tree are provided in the tab.

Rattle deploys (Therneau and Atkinson 2019) by default to build the decision tree, and many more options thatn are presented through the rattle GUI are available through using rpart::rpart() directly. These will be explored later in this chapter.

References

Therneau, Terry, and Beth Atkinson. 2019. *Rpart: Recursive Partitioning and Regression Trees*. https://CRAN.R-project.org/package=rpart.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.7 Rattle Build Tree

We can simply click the Execute button to build our first decision tree, using all of the default settings. Once finished, as in the figure above, the time taken to build the tree is reported in the status bar at the bottom of the window. A summary of the tree is presented in the text view panel. A classification model is built using rpart::rpart().

The number of observations from which the model was built is reported. This is 70% of the observations available in the dataset, representing the training dataset. We again note that the weather dataset is quite a tiny dataset in the context of data mining, but suitable for our purposes here.

The actual decision tree is then presented in its textual form.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.8 Interpret RPart Decision Tree

{#dtrees:sec:explain_read_tree}

```
## n= 256
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##  1) root 256 42 No (0.83593750 0.16406250)
##    2) Cloud3pm< 6.5 180 11 No (0.93888889 0.06111111) *
##    3) Cloud3pm>=6.5 76 31 No (0.59210526 0.40789474)
##      6) Pressure3pm>=1016 34  5 No (0.85294118 0.14705882) *
##      7) Pressure3pm< 1016 42 16 Yes (0.38095238 0.61904762)
##       14) WindDir3pm=ESE,SSE,W,WNW,WSW 17  5 No (0.70588235 0.29411765) *
##       15) WindDir3pm=ENE,N,NE,NNE,NNW,NW,S 25  4 Yes (0.16000000 0.84000000) *
```

The textual version of a classification decision tree is reported by rpart.

The legend, which begins with `node)` indicates that each node is identified by a number, followed by a split (which will usually be in the form of a test on the value of a variable), the number of observations n at that node, the number of observations that are incorrectly classified (the $loss$), the default classification for the node (the $yval$), and then the distribution of classes in that node (the $yprobs$) across `No` and `Yes`. The next line indicates that a *'' denotes a terminal node of the tree (i.e., a leaf node—the tree is not split any further at that node).

The actual tree starts with the root node labelled `1)`. observations and a default decision of `No`. There are 42 observations with `Yes` as the decision, so these are ` `lost'' if we make the decision` No `for all observations. The probability of` No `is reported as 0.83593750 (which is $214/256$) and of` Yes` is 0.16406250 ($42/256$).

The root node is split into two branches, nodes number 2 and 3. For node number 2, the split corresponds to those observations for which `Cloud3pm` is less than 6.5. This accounts for 180 observations and whilst 11 of them are `Yes` the majority (with a proportion of 0.93888889) are

`No`. We can read the remainder of the tree similarly. Node 3 is split into two other nodes, the second of which is split further until the terminal nodes.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.9 Rattle View Decision Tree

Click the Draw button to display the decision tree. A visual representation is often simpler to understand.

For a classification tree as we have here colour is used to differentiate the predicted (majority) class for each node. For our example the class/decision `No` is green and `Yes` is blue. The intensity of the colour indicates the strength of the prediction which is proportional to the percentage of the majority class within that node.

The root node (node number 1) has 84% `No` and 16% `Yes` and so is reported as a `No` decision or class. That is, in the absence of any other information, we predict that it will not rain tomorrow, and expect that prediction to be 84% correct.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.10 Rattle Error Matrix

Evaluating the performance of a model is crucial. Rattle supports a number of methods for evaluating performance. The simplest is the error matrix.

Click the Evaluate tab. Then click Execute to display a simple error matrix. Identify the True/False Positives/Negatives.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open*

source software since 1984.

Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.11 Rattle Risk Chart

Click the Risk type and then Execute.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.12 Rattle ROC Curve

Click the ROC type and then Execute.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.13 Rattle Hand Plots

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.14 Rattle Score Dataset

Click the Score type to score a new dataset using model.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.15 Rattle Log

Click the Log tab for a history of all your interactions. Save the log contents as a script to repeat what we did.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*

Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.16 Rattle GUI to R

The Log tab shows the call to rpart() to build the model. We can click on the Export button to save the script to file and that script can then be used to rerun this model building process, automatically within R.

The command to build the model is presented in the Log tab exactly as it is passed on to R to invoke the model building. It takes a little time to understand it, and the remainder of this module covers interacting directly with R to achieve the same results.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.17 Build a Decision Tree Model

As seen from Rattle's Log tab the decision tree model is built using rpart::rpart(). Once the different template variables have been defined as in Section @ref(dtrees:sec:model_setup) (form, ds, tr, and vars) we can use this template call to build the model:

```
library(rpart)
model <- rpart(formula=form, data=ds[tr, vars], model=TRUE)
```

This is essentially the same as the command used by Rattle except that some parameter settings are removed. These will be explored later.

In the above call to rpart::rpart() we have named each of the arguments. If we have a look at the structure of rpart::rpart() we see that the arguments are in their expected order, and hence the use of the argument names, and , is optional.

```
str(rpart)
```

```
## function (formula, data, weights, subset, na.action = na.rpart, method,
##     model = FALSE, x = FALSE, y = TRUE, parms, control, cost, ...)
```

Whilst the argument names are optional they can assist in reading the code, and so the use of argument names in function calls is encouraged.

A textual presentation of the model is concise and informative, once we learn how to read it. Note this tree is different to the previous one we have seen, since we are using a much larger (the full) weather dataset which includes multiple years of daily observations from many different weather stations across Australia.

```
model
```

```
## n= 123722
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##  1) root 123722 26101 No (0.7890351 0.2109649)
##    2) humidity_3pm< 72.5 105547 14889 No (0.8589349 0.1410651) *
##    3) humidity_3pm>=72.5 18175  6963 Yes (0.3831087 0.6168913)
##      6) humidity_3pm< 83.5 10161  4901 No (0.5176656 0.4823344)
##       12) rainfall< 2.7 6886  2757 No (0.5996224 0.4003776)
##         24) wind_gust_speed< 47 5118  1729 No (0.6621727 0.3378273) *
##         25) wind_gust_speed>=47 1768   740 Yes (0.4185520 0.5814480) *
##       13) rainfall>=2.7 3275  1131 Yes (0.3453435 0.6546565) *
##      7) humidity_3pm>=83.5 8014  1703 Yes (0.2125031 0.7874969) *
```

Refer to Section @ref(dtrees:sec:explain_read_tree) for an explanation of the format of the textual presentation of the decision tree.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.18 Summary of the Model

```
summary(model)
```

```
## Call:
## rpart(formula = form, data = ds[tr, vars], model = TRUE)
##   n= 123722
##
##           CP nsplit rel error    xerror        xstd
## 1 0.16279070      0 1.0000000 1.0000000 0.005498188
## 2 0.02628252      1 0.8372093 0.8437225 0.005154758
## 3 0.01103406      3 0.7846443 0.7888587 0.005019313
## 4 0.01000000      4 0.7736102 0.7880541 0.005017263
##
## Variable importance
##    humidity_3pm        sunshine        temp_3pm        rainfall        max_temp
##              78               4               4               4               2
##      rain_today wind_gust_speed    humidity_9am       cloud_3pm
##               2               2               1               1
##
## Node number 1: 123722 observations,    complexity param=0.1627907
##   predicted class=No   expected loss=0.2109649  P(node) =1
##     class counts: 97621 26101
##    probabilities: 0.789 0.211
##   left son=2 (105547 obs) right son=3 (18175 obs)
##   Primary splits:
##       humidity_3pm < 72.5    to the left,  improve=7021.018, (0 missing)
##       rainfall     < 0.45    to the left,  improve=4078.692, (0 missing)
##       rain_today   splits as  LR,          improve=4027.134, (0 missing)
##       cloud_3pm    < 6.5     to the left,  improve=2868.950, (0 missing)
##       sunshine     < 6.05    to the right, improve=2648.879, (0 missing)
##   Surrogate splits:
##       sunshine < 0.45    to the right, agree=0.861, adj=0.054, (0 split)
##       temp_3pm < 9.95    to the right, agree=0.860, adj=0.044, (0 split)
##       max_temp < 10.55   to the right, agree=0.857, adj=0.026, (0 split)
##       rainfall < 30.5    to the left,  agree=0.855, adj=0.014, (0 split)
##       temp_9am < -0.55   to the right, agree=0.854, adj=0.007, (0 split)
##
## Node number 2: 105547 observations
```

```
##   predicted class=No   expected loss=0.1410651  P(node) =0.8530981
##     class counts: 90658 14889
##    probabilities: 0.859 0.141
##
## Node number 3: 18175 observations,    complexity param=0.02628252
##   predicted class=Yes  expected loss=0.3831087  P(node) =0.1469019
##     class counts:  6963 11212
##    probabilities: 0.383 0.617
##   left son=6 (10161 obs) right son=7 (8014 obs)
##   Primary splits:
##       humidity_3pm    < 83.5    to the left,  improve=834.4563, (0 missing)
##       rainfall        < 2.25    to the left,  improve=547.7205, (0 missing)
##       rain_today      splits as  LR,          improve=537.8750, (0 missing)
##       wind_gust_speed < 42      to the left,  improve=370.3472, (0 missing)
##       pressure_3pm    < 1012.15 to the right, improve=333.3856, (0 missing)
##   Surrogate splits:
##       humidity_9am < 93.5    to the left,  agree=0.617, adj=0.132, (0 split)
##       cloud_3pm    < 7.5     to the left,  agree=0.605, adj=0.104, (0 split)
##       temp_3pm     < 10.55   to the right, agree=0.601, adj=0.095, (0 split)
##       sunshine     < 0.95    to the right, agree=0.593, adj=0.076, (0 split)
##       max_temp     < 11.25   to the right, agree=0.592, adj=0.075, (0 split)
##
## Node number 6: 10161 observations,    complexity param=0.02628252
##   predicted class=No   expected loss=0.4823344  P(node) =0.08212767
##     class counts:  5260  4901
##    probabilities: 0.518 0.482
##   left son=12 (6886 obs) right son=13 (3275 obs)
##   Primary splits:
##       rainfall        < 2.7     to the left,  improve=287.0070, (0 missing)
##       wind_gust_speed < 42      to the left,  improve=282.3802, (0 missing)
##       rain_today      splits as  LR,          improve=278.2783, (0 missing)
##       pressure_9am    < 1013.35 to the right, improve=221.9931, (0 missing)
##       pressure_3pm    < 1012.65 to the right, improve=212.4673, (0 missing)
##   Surrogate splits:
##       rain_today      splits as  LR,          agree=0.896, adj=0.677, (0 split)
##       pressure_9am    < 1006.65 to the right, agree=0.689, adj=0.034, (0 split)
```

```
##       pressure_3pm    < 1005.15 to the right, agree=0.685, adj=0.023, (0 split)
##       temp_3pm        < 28.05   to the left,  agree=0.680, adj=0.008, (0 split)
##       wind_gust_speed < 77      to the left,  agree=0.680, adj=0.008, (0 split)
##
## Node number 7: 8014 observations
##   predicted class=Yes  expected loss=0.2125031  P(node) =0.06477425
##     class counts:  1703  6311
##    probabilities: 0.213 0.787
##
## Node number 12: 6886 observations,    complexity param=0.01103406
##   predicted class=No   expected loss=0.4003776  P(node) =0.05565704
##     class counts:  4129  2757
##    probabilities: 0.600 0.400
##   left son=24 (5118 obs) right son=25 (1768 obs)
##   Primary splits:
##       wind_gust_speed < 47      to the left,  improve=155.98180, (0 missing)
##       pressure_3pm    < 1012.65 to the right, improve=128.70310, (0 missing)
##       pressure_9am    < 1015.45 to the right, improve=119.38520, (0 missing)
##       cloud_3pm       < 6.5     to the left,  improve=118.68430, (0 missing)
##       sunshine        < 7.95    to the right, improve= 71.97008, (0 missing)
##   Surrogate splits:
##       wind_speed_3pm < 29      to the left,  agree=0.820, adj=0.299, (0 split)
##       wind_speed_9am < 23      to the left,  agree=0.804, adj=0.236, (0 split)
##       pressure_9am   < 1008.05 to the right, agree=0.754, adj=0.044, (0 split)
##       pressure_3pm   < 1003.45 to the right, agree=0.752, adj=0.035, (0 split)
##       humidity_9am   < 56.5    to the right, agree=0.748, adj=0.020, (0 split)
##
## Node number 13: 3275 observations
##   predicted class=Yes  expected loss=0.3453435  P(node) =0.02647064
##     class counts:  1131  2144
##    probabilities: 0.345 0.655
##
## Node number 24: 5118 observations
##   predicted class=No   expected loss=0.3378273  P(node) =0.04136694
##     class counts:  3389  1729
##    probabilities: 0.662 0.338
```

```
##
## Node number 25: 1768 observations
##   predicted class=Yes  expected loss=0.418552  P(node) =0.0142901
##     class counts:   740  1028
##    probabilities: 0.419 0.581
```

In the following pages we dissect the various components of this summary.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.19 Complexity Parameter

We can print a table of optimal prunings based on a complexity parameter using rpart::printcp(). The data is actually stored as `model$cptable`.

```
printcp(model)
```

```
##
## Classification tree:
## rpart(formula = form, data = ds[tr, vars], model = TRUE)
##
## Variables actually used in tree construction:
## [1] humidity_3pm    rainfall        wind_gust_speed
##
## Root node error: 26101/123722 = 0.21096
##
## n= 123722
##
##         CP nsplit rel error  xerror      xstd
## 1 0.162791      0   1.00000 1.00000 0.0054982
## 2 0.026283      1   0.83721 0.84372 0.0051548
## 3 0.011034      3   0.78464 0.78886 0.0050193
## 4 0.010000      4   0.77361 0.78805 0.0050173
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.20 Complexity Parameter Plot

The rpart::plotcp() plots the cross-validation results. Here we see a set of possible cost-complexity prunings of the tree. We might choose to prune using the leftmost complexity parameter which has a mean below the horizontal line.

```
plotcp(model)
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open*

source software since 1984.

Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.21 Complexity Parameter Behaviour

We don't see much of the true behaviour of the complexity parameter with our small dataset. We can build a more interesting model based on the larger dataset from .

If we build a default model then we can plot the complexity parameter as before.

```
tmodel <- rpart(form, ds[vars])
plotcp(tmodel)
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of*

Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.22 Complexity Parameter 0

We can set the argument to be 0, so that no pruning of the tree is performed.

```
tmodel <- rpart(form, ds[vars], control=rpart.control(cp=0))
plotcp(tmodel)
```

Notice that as we continue to build the model, by recursive partitioning, the model gets more complex but the performance does not improve, and in fact over time the model performance starts to deteriorate because of overfitting.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.23 Complexity Parameter Table

We can look at the raw data to have a more precise and detailed view of the data. Here we only list specific rows from the complexity parameter table.

```
tmodel$cptable[c(1:5,22:29, 80:83),]
```

```
##              CP nsplit rel error    xerror        xstd
## 1  1.453731e-01      0 1.0000000 1.0000000 0.004616611
## 2  3.460366e-02      1 0.8546269 0.8546269 0.004349444
## 3  6.877579e-03      3 0.7854195 0.7894112 0.004214897
## 4  4.814305e-03      4 0.7785420 0.7856353 0.004206800
## 5  3.506217e-03      8 0.7592847 0.7633573 0.004158311
## 22 5.394180e-04     40 0.7199342 0.7314238 0.004086605
## 23 5.259325e-04     41 0.7193948 0.7308574 0.004085309
## 24 5.214374e-04     52 0.7130566 0.7302910 0.004084012
## 25 5.124471e-04     55 0.7114923 0.7302640 0.004083951
## 26 4.854762e-04     57 0.7104674 0.7300752 0.004083518
## 27 4.719907e-04     60 0.7090110 0.7294279 0.004082035
## 28 4.585053e-04     64 0.7071230 0.7294279 0.004082035
## 29 4.315344e-04     66 0.7062060 0.7285649 0.004080055
## 80 8.630687e-05    791 0.5906627 0.7232246 0.004067761
## 81 8.540784e-05    807 0.5888017 0.7235213 0.004068446
## 82 8.091270e-05    815 0.5880735 0.7241686 0.004069940
## 83 7.754133e-05   1012 0.5709469 0.7263533 0.004074973
```

%$

See how the relative error continues to decrease as the tree becomes more complex, but the cross validated error decreases and then starts to increase! We might choose a sensible value of from this table.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.24 Variable Importance

```
model$variable.importance
```

```
##    humidity_3pm        sunshine        temp_3pm        rainfall        max_temp
##      7855.47467       440.08556       392.08693       388.21787       242.69935
##      rain_today wind_gust_speed    humidity_9am       cloud_3pm        temp_9am
##       194.28843       158.34794       112.93956        86.42360        46.74241
##  wind_speed_3pm  wind_speed_9am    pressure_9am    pressure_3pm
##        46.67102        36.87804        16.43325        11.86677
```

%$

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.25 Node Details and Surrogates

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.26 Decision Tree Performance

Here we plot the performance of the decision tree, showing a risk chart. The areas under the recall and risk curves are also reported.

```
predict_te <- predict(model, ds[te, vars], type="prob")[,2]
riskchart(predict_te, actual_te, risk_te)
```

An error matrix shows, clockwise from the top left, the percentages of true negatives, false positives, true positives, and false negatives.

```
predict_te <- predict(model, ds[te, vars], type="class")
sum(actual_te != predict_te)/length(predict_te) # Overall error rate
```

```
## [1] 0.1616943
```

```
actual_te %>%
  table(predict_te, dnn=c("Actual", "Predicted")) %>%
  '*'(100/(length(predict_te))) %>%
  round()
```

```
##       Predicted
## Actual No Yes
##    No  77   3
##    Yes 13   7
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.27 Rules from Decision Tree

We can use rattle::asRules() to convert the decision tree to a set of rules:

```
asRules(model)
```

```
##
##  Rule number: 7 [rain_tomorrow=Yes cover=8014 (6%) prob=0.79]
##    humidity_3pm>=72.5
##    humidity_3pm>=83.5
##
##  Rule number: 13 [rain_tomorrow=Yes cover=3275 (3%) prob=0.65]
##    humidity_3pm>=72.5
##    humidity_3pm< 83.5
##    rainfall>=2.7
##
##  Rule number: 25 [rain_tomorrow=Yes cover=1768 (1%) prob=0.58]
##    humidity_3pm>=72.5
##    humidity_3pm< 83.5
##    rainfall< 2.7
##    wind_gust_speed>=47
##
##  Rule number: 24 [rain_tomorrow=No cover=5118 (4%) prob=0.34]
##    humidity_3pm>=72.5
##    humidity_3pm< 83.5
##    rainfall< 2.7
##    wind_gust_speed< 47
##
##  Rule number: 2 [rain_tomorrow=No cover=105547 (85%) prob=0.14]
##    humidity_3pm< 72.5
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.28 Rules Using Rpart Plot

We can also use rpart.plot::rpart.rules() from Stephen Milborrow's excellent rpart.plot (Milborrow 2020) package. This will convert the decision tree into rules:

```
rpart.rules(model)
```

```
##  rain_tomorrow
##           0.14 when humidity_3pm <  73
##           0.34 when humidity_3pm is 73 to 84 & rainfall <  2.7 & wind_gust_speed
##           0.58 when humidity_3pm is 73 to 84 & rainfall <  2.7 & wind_gust_speed
##           0.65 when humidity_3pm is 73 to 84 & rainfall >= 2.7
##           0.79 when humidity_3pm >=       84
```

Some variations appear below. The first adds the coverage of the rule, which is the percentage of the training dataset that ends up in this part of the decision tree. The second adds the proportions of observations within a leaf node that correspond to No/Yes.

```
rpart.rules(model, cover=TRUE)
```

```
##  rain_tomorrow
##           0.14 when humidity_3pm <  73
##           0.34 when humidity_3pm is 73 to 84 & rainfall <  2.7 & wind_gust_speed
##           0.58 when humidity_3pm is 73 to 84 & rainfall <  2.7 & wind_gust_speed
##           0.65 when humidity_3pm is 73 to 84 & rainfall >= 2.7
##           0.79 when humidity_3pm >=       84
```

```
rpart.rules(model, extra=4)
```

```
##  rain_tomorrow   No Yes
##             No [.86 .14] when humidity_3pm <  73
##             No [.66 .34] when humidity_3pm is 73 to 84 & rainfall <  2.7 & wind_g
##            Yes [.42 .58] when humidity_3pm is 73 to 84 & rainfall <  2.7 & wind_g
##            Yes [.35 .65] when humidity_3pm is 73 to 84 & rainfall >= 2.7
##            Yes [.21 .79] when humidity_3pm >=       84
```

We can even obtain an explanation of the rules used in a prediction. Here we select specific observations to obtain the prediction and the rule used to make that prediction.

```
rpart.predict(model, newdata=ds[c(10, 12, 17), vars], rules=TRUE)
```

```
##          No       Yes
## 1 0.8589349 0.1410651
## 2 0.2125031 0.7874969
## 3 0.6621727 0.3378273
##
## 1 because humidity_3pm <  73
## 2 because humidity_3pm >=       84
## 3 because humidity_3pm is 73 to 84 & rainfall <  2.7 & wind_gust_speed <  47
```

References

Milborrow, Stephen. 2020. *Rpart.plot: Plot Rpart Models: An Enhanced Version of Plot.rpart*. http://www.milbo.org/rpart-plot/index.html.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.29 Plot Decision Trees

```
plot(model)
text(model)
```

The default plot of the model is quite basic. In this plot we move to the left in the binary tree if the condition is true.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open*

source software since 1984.

Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.30 Plot Decision Tree Uniformly

```
plot(model, uniform=TRUE)
text(model)
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.31 Plot Decision Tree with Extra Information

```
plot(model, uniform=TRUE)
text(model, use.n=TRUE, all=TRUE, cex=.8)
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.32 Fancy Rpart Plot

```
fancyRpartPlot(model)
```

The rattle (G. Williams 2020) package provides a fancy plot based on the functionality provided by (Milborrow 2020) and using colours from (Neuwirth 2014), tuned for use in : rattle::fancyRpartPlot(). The same options can be passed directly to rpart.plot::prp() to achieve the same plot and colours, and Stephen Milborrow demonstrates this mapping in his guide at http://www.milbo.org/doc/prp.pdf, page 11. In the following pages we demonstrate some of the alternative options provided by rpart.plot::prp().

References

Milborrow, Stephen. 2020. *Rpart.plot: Plot Rpart Models: An Enhanced Version of Plot.rpart*. http://www.milbo.org/rpart-plot/index.html.

Neuwirth, Erich. 2014. *RColorBrewer: ColorBrewer Palettes*. https://CRAN.R-project.org/package=RColorBrewer.

Williams, Graham. 2020. *Rattle: Graphical User Interface for Data Science in r*. https://rattle.togaware.com/.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*

Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.33 RPart Plot Default Tree

```
prp(model)
```

Stephen Milborrow's provides a suite of enhancements to the basic base::plot() command. The following pages exhibit the various (and quite extensive) options provided by and specifically rpart.plot::prp().

For many examples from the author see http://www.milbo.org/doc/prp.pdf. The variety is impressive and should cater to most presentation styles and preferences.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.34 RPart Plot Favourite

```
prp(model, type=2, extra=104, nn=TRUE, fallen.leaves=TRUE,
    faclen=0, varlen=0, shadow.col="grey", branch.lty=3)
```

This is a plot that I find particularly useful, neat, and informative, particularly for classification models.

The leaf nodes are each labelled with the predicted class. They are neatly lined up at the bottom of the figure (*fallen.leaves=TRUE*), to visually reinforce the structure. We can see the straight lines from the top to the bottom which lead to decisions quickly, whilst the more complex paths need quite a bit more information in order to make a decision.

Each node includes the probability for each class, and the percentage of observations associated with the node (*extra=104*). The node numbers are included (*nn=TRUE*) so we can cross reference each node to the text decision tree, or other decision tree plots, or a rule set generated from the decision tree.

Using a dotted line type (*branch.lty=3*) removes some of the focus from the heavy lines and back to the nodes, whilst still clearly identifying the links. The grey shadow is an optional nicety.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.35 Enhanced Plot: With Colour

```
col <- c("#FD8D3C", "#FD8D3C", "#FD8D3C", "#BCBDDC",
         "#FDD0A2", "#FD8D3C", "#BCBDDC")
prp(model, type=2, extra=104, nn=TRUE, fallen.leaves=TRUE,
    faclen=0, varlen=0, shadow.col="grey", branch.lty=3, box.col=col)
```

The rattle::fancyRpartPlot() function from (G. Williams 2020) generates scaled colour for colouring the boxes depending on the decision and the strength. The hard work is in generating the scaled colours for the nodes. Here we use other palettes for the colours rather than those used by rattle::fancyRpartPlot().

References

Williams, Graham. 2020. *Rattle: Graphical User Interface for Data Science in r*. https://rattle.togaware.com/.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.36 Enhanced Plots: Label all Nodes

```
prp(model, type=1)
```

Here all nodes are labelled with the majority class.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.37 Enhanced Plots: Labels Below Nodes

```
prp(model, type=2)
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.38 Enhanced Plots: Split Labels

```
prp(model, type=3)
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.39 Enhanced Plots: Interior Labels

```
prp(model, type=4)
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.40 Enhanced Plots: Number of Observations

```
prp(model, type=2, extra=1)
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.41 Enhanced Plots: Add Percentage of Observations

```
prp(model, type=2, extra=101)
```

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.42 Enhanced Plots: Classification Rate

```
prp(model, type=2, extra=2)
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.43 Enhanced Plots: Add Percentage of Observations

```
prp(model, type=2, extra=102)
```

Notice the pattern? When we add 100 to the *extra*= option then the percentage of observations located with each node is then included in the plot.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.44 Enhanced Plots: Misclassification Rate

```
prp(model, type=2, extra=3)
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.45 Enhanced Plots: Probability per Class

```
prp(model, type=2, extra=4)
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.46 Enhanced Plots: Add Percentage Observations

```
prp(model, type=2, extra=104)
```

This is a particularly informative plot for classification models. Each node includes the probability of each class. Each node also includes the percentage of the training dataset that corresponds to that node.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.47 Enhanced Plots: Only Probability Per Class

```
prp(model, type=2, extra=5)
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.48 Enhanced Plots: Probability of Second Class

```
prp(model, type=2, extra=6)
```

This is particularly useful for binary classification, as here, where the second class is usually the positive response.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.49 Enhanced Plots: Add Percentage Observations

```
prp(model, type=2, extra=106)
```

This is a particularly informative plot for binary classification tasks. Each node includes the probability of the second class, which is usually the positive class in a binary classification dataset. Each node also includes the percentage of the training dataset that corresponds to that node.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.50 Enhanced Plots: Only Probability of Second Class

```
prp(model, type=2, extra=7)
```

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.51 Enhanced Plots: Probability of the Class

```
prp(model, type=2, extra=8)
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.52 Enhanced Plots: Overall Probability

```
prp(model, type=2, extra=9)
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.53 Enhanced Plots: Percentage of Observations

```
prp(model, type=2, extra=100)
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.54 Enhanced Plots: Show the Node Numbers

```
prp(model, type=2, extra=106, nn=TRUE)
```

We now take our favourite plot (*type=2* and *extra=106*) and explore some of the other options available for rpart.plot::prp() from .

Here we add the node numbers as they appear in the textual version of the model, and also often for rule sets generated from the decision tree.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.55 Enhanced Plots: Show the Node Indicies

```
prp(model, type=2, extra=106, nn=TRUE, ni=TRUE)
```

These are the row numbers of the nodes within the model object's `frame` component.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.56 Enhanced Plots: Line up the Leaves

```
prp(model, type=2, extra=106, nn=TRUE, fallen.leaves=TRUE)
```

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.57 Enhanced Plots: Angle Branch Lines

```
prp(model, type=2, extra=106, nn=TRUE, fallen.leaves=TRUE,
    branch=0.5)
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.58 Enhanced Plots: Do Not Abbreviate Factors

```
prp(model, type=2, extra=106, nn=TRUE, fallen.leaves=TRUE,
    faclen=0)
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.59 Enhanced Plots: Add a Shadow to the Nodes

```
prp(model, type=2, extra=106, nn=TRUE, fallen.leaves=TRUE,
    shadow.col="grey")
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.60 Enhanced Plots: Draw Branches as Dotted Lines

```
prp(model, type=2, extra=106, nn=TRUE, fallen.leaves=TRUE,
    branch.lty=3)
```

The *branch.lty=* option allows us to specify the type of line to draw for the branches. A dotted line is attractive as it reduces the dominance of the branches whilst retaining the node connections. Other options are just the standard values for line type in R:

% Following came from % http://students.washington.edu/mclarkso/documents/line%20styles%20Ver2.pdf Give credit in Further Reading section of Plots.Rnw

The line type can also be specified as an even length string of up eight characters of the hex digits (0–9, a–f). The pairs specify the length in pixels of the line and the blank. Thus *lty="44"* is the same as *lty=2*:

```
plot(c(0,1), c(0,0), type="l", axes=FALSE, xlab=NA, ylab=NA, lty=2)
plot(c(0,1), c(0,0), type="l", axes=FALSE, xlab=NA, ylab=NA, lty="dashed")
plot(c(0,1), c(0,0), type="l", axes=FALSE, xlab=NA, ylab=NA, lty="44")
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.61 Enhanced Plots: Other Options

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.62 Party Tree

The (Hothorn et al. 2021) package can be used to draw decision trees using partykit::as.party() from (Hothorn and Zeileis 2021) which can be installed from R-Forge:

```
install.packages("partykit", repos="http://R-Forge.R-project.org")
library(partykit)
```

```
class(model)
```

```
## [1] "rpart"
```

```
plot(as.party(model))
```

The textual presentation of an rpart decision tree can also be improved using party.

```
print(as.party(model))
```

```
##
## Model formula:
## rain_tomorrow ~ rain_today + temp_3pm + temp_9am + cloud_3pm +
##     cloud_9am + pressure_3pm + pressure_9am + humidity_3pm +
##     humidity_9am + wind_speed_3pm + wind_speed_9am + wind_dir_3pm +
##     wind_dir_9am + wind_gust_speed + wind_gust_dir + sunshine +
##     evaporation + rainfall + max_temp + min_temp
##
## Fitted party:
## [1] root
## |   [2] humidity_3pm < 72.5: No (n = 105547, err = 14.1%)
## |   [3] humidity_3pm >= 72.5
## |   |   [4] humidity_3pm < 83.5
## |   |   |   [5] rainfall < 2.7
## |   |   |   |   [6] wind_gust_speed < 47: No (n = 5118, err = 33.8%)
## |   |   |   |   [7] wind_gust_speed >= 47: Yes (n = 1768, err = 41.9%)
## |   |   |   [8] rainfall >= 2.7: Yes (n = 3275, err = 34.5%)
## |   |   [9] humidity_3pm >= 83.5: Yes (n = 8014, err = 21.3%)
##
## Number of inner nodes:    4
## Number of terminal nodes: 5
```

References

Hothorn, Torsten, Kurt Hornik, Carolin Strobl, and Achim Zeileis. 2021. *Party: A Laboratory for Recursive Partytioning*. http://party.R-forge.R-project.org.

Hothorn, Torsten, and Achim Zeileis. 2021. *Partykit: A Toolkit for Recursive Partytioning*. http://partykit.r-forge.r-project.org/partykit/.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*

Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.63 Conditional Decision Tree

Note that we are using the newer version of the ctree() function as is provided by (Hothorn and Zeileis 2021). One advantage of the newer version is that predict() with works just like other predict() methods (returns a matrix rather than a list).

```
library(partykit)
model <- ctree(formula=form, data=ds[tr, vars])
```

```
model
```

```
##
## Model formula:
## rain_tomorrow ~ rain_today + temp_3pm + temp_9am + cloud_3pm +
##     cloud_9am + pressure_3pm + pressure_9am + humidity_3pm +
##     humidity_9am + wind_speed_3pm + wind_speed_9am + wind_dir_3pm +
##     wind_dir_9am + wind_gust_speed + wind_gust_dir + sunshine +
##     evaporation + rainfall + max_temp + min_temp
##
## Fitted party:
## [1] root
## |   [2] humidity_3pm <= 72
## |   |   [3] humidity_3pm <= 50
## |   |   |   [4] wind_gust_speed <= 54
## |   |   |   |   [5] sunshine <= 7.5
## |   |   |   |   |   [6] pressure_3pm <= 1015
## |   |   |   |   |   |   [7] humidity_3pm <= 27: No (n = 313, err = 8.9%)
## |   |   |   |   |   |   [8] humidity_3pm > 27
## |   |   |   |   |   |   |   [9] wind_dir_9am <= NE
## |   |   |   |   |   |   |   |   [10] wind_gust_speed <= 35
## |   |   |   |   |   |   |   |   |   [11] wind_dir_9am <= N: No (n = 58, err = 12
## |   |   |   |   |   |   |   |   |   [12] wind_dir_9am > N: No (n = 71, err = 39.4
## |   |   |   |   |   |   |   |   [13] wind_gust_speed > 35: No (n = 372, err = 48
## |   |   |   |   |   |   |   [14] wind_dir_9am > NE
## |   |   |   |   |   |   |   |   [15] pressure_3pm <= 1004.7: No (n = 113, err = 4
## |   |   |   |   |   |   |   |   [16] pressure_3pm > 1004.7
## |   |   |   |   |   |   |   |   |   [17] wind_speed_9am <= 19
## |   |   |   |   |   |   |   |   |   |   [18] wind_gust_speed <= 41: No (n = 540,
## |   |   |   |   |   |   |   |   |   |   [19] wind_gust_speed > 41: No (n = 273, e
## |   |   |   |   |   |   |   |   |   [20] wind_speed_9am > 19: No (n = 249, err =
## |   |   |   |   |   [21] pressure_3pm > 1015
## |   |   |   |   |   |   [22] wind_gust_speed <= 33
## |   |   |   |   |   |   |   [23] pressure_3pm <= 1023.1
## |   |   |   |   |   |   |   |   [24] sunshine <= 3.8: No (n = 177, err = 14.7%)
## |   |   |   |   |   |   |   |   [25] sunshine > 3.8: No (n = 544, err = 5.5%)
## |   |   |   |   |   |   |   [26] pressure_3pm > 1023.1
```

```
## |   |   |   |   |   |   |   [27] rainfall <= 5.8
## |   |   |   |   |   |   |   |   [28] wind_speed_9am <= 13: No (n = 264, err =
## |   |   |   |   |   |   |   |   [29] wind_speed_9am > 13: No (n = 67, err = 4
## |   |   |   |   |   |   |   [30] rainfall > 5.8: No (n = 7, err = 14.3%)
## |   |   |   |   |   |   [31] wind_gust_speed > 33
## |   |   |   |   |   |   |   [32] wind_dir_9am <= ENE
## |   |   |   |   |   |   |   [33] pressure_3pm <= 1019.9
## |   |   |   |   |   |   |   |   [34] humidity_3pm <= 41: No (n = 155, err = 2
## |   |   |   |   |   |   |   |   [35] humidity_3pm > 41: No (n = 118, err = 39
## |   |   |   |   |   |   |   [36] pressure_3pm > 1019.9: No (n = 175, err = 8
## |   |   |   |   |   |   |   [37] wind_dir_9am > ENE: No (n = 840, err = 9.6%)
## |   |   |   |   [38] sunshine > 7.5
## |   |   |   |   |   [39] pressure_3pm <= 1009.3
## |   |   |   |   |   |   [40] humidity_3pm <= 26
## |   |   |   |   |   |   |   [41] humidity_3pm <= 13: No (n = 1003, err = 1.2%)
## |   |   |   |   |   |   |   [42] humidity_3pm > 13
## |   |   |   |   |   |   |   |   [43] sunshine <= 10.7: No (n = 1198, err = 6.9%)
## |   |   |   |   |   |   |   |   [44] sunshine > 10.7: No (n = 566, err = 2.1%)
## |   |   |   |   |   |   [45] humidity_3pm > 26
## |   |   |   |   |   |   |   [46] sunshine <= 9.9
## |   |   |   |   |   |   |   |   [47] wind_gust_speed <= 44
## |   |   |   |   |   |   |   |   |   [48] wind_speed_9am <= 9
## |   |   |   |   |   |   |   |   |   |   [49] max_temp <= 28.2: No (n = 282, err =
## |   |   |   |   |   |   |   |   |   |   [50] max_temp > 28.2
## |   |   |   |   |   |   |   |   |   |   |   [51] rainfall <= 5.2: No (n = 428, er
## |   |   |   |   |   |   |   |   |   |   |   [52] rainfall > 5.2: No (n = 28, err
## |   |   |   |   |   |   |   |   |   [53] wind_speed_9am > 9: No (n = 835, err = 1
## |   |   |   |   |   |   |   |   [54] wind_gust_speed > 44
## |   |   |   |   |   |   |   |   |   [55] wind_dir_9am <= ESE: No (n = 295, err =
## |   |   |   |   |   |   |   |   |   [56] wind_dir_9am > ESE
## |   |   |   |   |   |   |   |   |   |   [57] wind_speed_3pm <= 13: No (n = 45, er
## |   |   |   |   |   |   |   |   |   |   [58] wind_speed_3pm > 13: No (n = 405, er
## |   |   |   |   |   |   |   [59] sunshine > 9.9
## |   |   |   |   |   |   |   |   [60] cloud_3pm <= 4: No (n = 756, err = 4.4%)
## |   |   |   |   |   |   |   |   [61] cloud_3pm > 4: No (n = 348, err = 11.5%)
## |   |   |   |   |   [62] pressure_3pm > 1009.3
```

```
## |   |   |   |   |   |   [63] humidity_3pm <= 36
## |   |   |   |   |   |   |   [64] humidity_3pm <= 24
## |   |   |   |   |   |   |   |   [65] sunshine <= 9.8
## |   |   |   |   |   |   |   |   |   [66] wind_gust_speed <= 41: No (n = 3904, err
## |   |   |   |   |   |   |   |   |   [67] wind_gust_speed > 41
## |   |   |   |   |   |   |   |   |   |   [68] humidity_3pm <= 16: No (n = 925, err
## |   |   |   |   |   |   |   |   |   |   [69] humidity_3pm > 16
## |   |   |   |   |   |   |   |   |   |   |   [70] wind_dir_9am <= NE: No (n = 329,
## |   |   |   |   |   |   |   |   |   |   |   [71] wind_dir_9am > NE: No (n = 691,
## |   |   |   |   |   |   |   |   [72] sunshine > 9.8: No (n = 3647, err = 0.4%)
## |   |   |   |   |   |   |   [73] humidity_3pm > 24
## |   |   |   |   |   |   |   |   [74] pressure_3pm <= 1015.4
## |   |   |   |   |   |   |   |   |   [75] wind_gust_speed <= 37
## |   |   |   |   |   |   |   |   |   |   [76] sunshine <= 9.7: No (n = 1921, err =
## |   |   |   |   |   |   |   |   |   |   [77] sunshine > 9.7: No (n = 715, err = 0
## |   |   |   |   |   |   |   |   |   [78] wind_gust_speed > 37
## |   |   |   |   |   |   |   |   |   |   [79] wind_dir_9am <= ENE
## |   |   |   |   |   |   |   |   |   |   |   [80] wind_gust_speed <= 50: No (n = 9
## |   |   |   |   |   |   |   |   |   |   |   [81] wind_gust_speed > 50: No (n = 74
## |   |   |   |   |   |   |   |   |   |   [82] wind_dir_9am > ENE
## |   |   |   |   |   |   |   |   |   |   |   [83] wind_speed_9am <= 6: No (n = 248
## |   |   |   |   |   |   |   |   |   |   |   [84] wind_speed_9am > 6
## |   |   |   |   |   |   |   |   |   |   |   |   [85] cloud_3pm <= 7
## |   |   |   |   |   |   |   |   |   |   |   |   |   [86] wind_dir_9am <= NW
## |   |   |   |   |   |   |   |   |   |   |   |   |   |   [87] temp_3pm <= 34.5
## |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   [88] rain_today in No
## |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   [89] rain_today in Ye
## |   |   |   |   |   |   |   |   |   |   |   |   |   |   [90] temp_3pm > 34.5: No
## |   |   |   |   |   |   |   |   |   |   |   |   |   [91] wind_dir_9am > NW: No (n
## |   |   |   |   |   |   |   |   |   |   |   |   [92] cloud_3pm > 7: No (n = 32, e
## |   |   |   |   |   |   |   |   [93] pressure_3pm > 1015.4
## |   |   |   |   |   |   |   |   |   [94] pressure_3pm <= 1019.5
## |   |   |   |   |   |   |   |   |   |   [95] sunshine <= 9.6
## |   |   |   |   |   |   |   |   |   |   |   [96] pressure_9am <= 1019.2: No (n =
## |   |   |   |   |   |   |   |   |   |   |   [97] pressure_9am > 1019.2: No (n = 1
## |   |   |   |   |   |   |   |   |   |   [98] sunshine > 9.6
```

```
## |   |   |   |   |   |   |   |   |   |   [99] cloud_3pm <= 4
## |   |   |   |   |   |   |   |   |   |   |   [100] rainfall <= 8: No (n = 817,
## |   |   |   |   |   |   |   |   |   |   |   [101] rainfall > 8: No (n = 11, e
## |   |   |   |   |   |   |   |   |   |   [102] cloud_3pm > 4: No (n = 334, er
## |   |   |   |   |   |   |   |   [103] pressure_3pm > 1019.5: No (n = 3233, e
## |   |   |   |   |   |   [104] humidity_3pm > 36
## |   |   |   |   |   |   |   [105] pressure_3pm <= 1016.5
## |   |   |   |   |   |   |   |   [106] sunshine <= 9.3
## |   |   |   |   |   |   |   |   |   [107] wind_gust_speed <= 37: No (n = 3283, e
## |   |   |   |   |   |   |   |   |   [108] wind_gust_speed > 37
## |   |   |   |   |   |   |   |   |   |   [109] wind_speed_9am <= 15
## |   |   |   |   |   |   |   |   |   |   |   [110] wind_gust_speed <= 46: No (n =
## |   |   |   |   |   |   |   |   |   |   |   [111] wind_gust_speed > 46: No (n = 3
## |   |   |   |   |   |   |   |   |   |   [112] wind_speed_9am > 15
## |   |   |   |   |   |   |   |   |   |   |   [113] wind_gust_speed <= 48
## |   |   |   |   |   |   |   |   |   |   |   |   [114] pressure_3pm <= 1011.7: No
## |   |   |   |   |   |   |   |   |   |   |   |   [115] pressure_3pm > 1011.7: No
## |   |   |   |   |   |   |   |   |   |   |   [116] wind_gust_speed > 48: No (n = 3
## |   |   |   |   |   |   |   |   [117] sunshine > 9.3
## |   |   |   |   |   |   |   |   |   [118] sunshine <= 11.9: No (n = 1874, err = 4
## |   |   |   |   |   |   |   |   |   [119] sunshine > 11.9: No (n = 675, err = 1.3
## |   |   |   |   |   |   |   [120] pressure_3pm > 1016.5
## |   |   |   |   |   |   |   |   [121] wind_gust_speed <= 33
## |   |   |   |   |   |   |   |   |   [122] pressure_3pm <= 1024.1
## |   |   |   |   |   |   |   |   |   |   [123] sunshine <= 9.5: No (n = 2634, err
## |   |   |   |   |   |   |   |   |   |   [124] sunshine > 9.5: No (n = 943, err =
## |   |   |   |   |   |   |   |   |   [125] pressure_3pm > 1024.1: No (n = 1577, e
## |   |   |   |   |   |   |   |   [126] wind_gust_speed > 33
## |   |   |   |   |   |   |   |   |   [127] sunshine <= 9.8
## |   |   |   |   |   |   |   |   |   |   [128] rain_today in No
## |   |   |   |   |   |   |   |   |   |   |   [129] pressure_3pm <= 1018.7
## |   |   |   |   |   |   |   |   |   |   |   |   [130] pressure_9am <= 1022.3: No
## |   |   |   |   |   |   |   |   |   |   |   |   [131] pressure_9am > 1022.3: No
## |   |   |   |   |   |   |   |   |   |   |   [132] pressure_3pm > 1018.7: No (n =
## |   |   |   |   |   |   |   |   |   |   [133] rain_today in Yes: No (n = 380, er
## |   |   |   |   |   |   |   |   |   [134] sunshine > 9.8
```

```
## |   |   |   |   |   |   |   |   |   [135] rainfall <= 12: No (n = 1219, err =
## |   |   |   |   |   |   |   |   |   [136] rainfall > 12: No (n = 13, err = 7
## |   |   |   [137] wind_gust_speed > 54
## |   |   |   |   [138] sunshine <= 7.7
## |   |   |   |   |   [139] pressure_3pm <= 1006.1
## |   |   |   |   |   |   [140] sunshine <= 4
## |   |   |   |   |   |   |   [141] humidity_3pm <= 21: No (n = 19, err = 31.6%)
## |   |   |   |   |   |   |   [142] humidity_3pm > 21: Yes (n = 105, err = 25.7%)
## |   |   |   |   |   |   [143] sunshine > 4: No (n = 353, err = 47.0%)
## |   |   |   |   |   [144] pressure_3pm > 1006.1
## |   |   |   |   |   |   [145] rainfall <= 10.2: No (n = 771, err = 29.6%)
## |   |   |   |   |   |   [146] rainfall > 10.2: Yes (n = 37, err = 35.1%)
## |   |   |   |   [147] sunshine > 7.7
## |   |   |   |   |   [148] pressure_3pm <= 1008.8
## |   |   |   |   |   |   [149] humidity_3pm <= 14
## |   |   |   |   |   |   |   [150] wind_gust_speed <= 76: No (n = 474, err = 6.5%
## |   |   |   |   |   |   |   [151] wind_gust_speed > 76: No (n = 69, err = 21.7%)
## |   |   |   |   |   |   [152] humidity_3pm > 14
## |   |   |   |   |   |   |   [153] humidity_9am <= 55
## |   |   |   |   |   |   |   |   [154] cloud_3pm <= 5
## |   |   |   |   |   |   |   |   |   [155] wind_dir_9am <= ENE
## |   |   |   |   |   |   |   |   |   |   [156] pressure_3pm <= 1004.4
## |   |   |   |   |   |   |   |   |   |   |   [157] wind_dir_3pm <= SSW: Yes (n = 4
## |   |   |   |   |   |   |   |   |   |   |   [158] wind_dir_3pm > SSW: No (n = 84,
## |   |   |   |   |   |   |   |   |   |   [159] pressure_3pm > 1004.4: No (n = 169,
## |   |   |   |   |   |   |   |   |   [160] wind_dir_9am > ENE: No (n = 596, err =
## |   |   |   |   |   |   |   |   [161] cloud_3pm > 5
## |   |   |   |   |   |   |   |   |   [162] temp_3pm <= 27.8: No (n = 121, err = 22
## |   |   |   |   |   |   |   |   |   [163] temp_3pm > 27.8
## |   |   |   |   |   |   |   |   |   |   [164] wind_gust_speed <= 74: No (n = 109,
## |   |   |   |   |   |   |   |   |   |   [165] wind_gust_speed > 74: Yes (n = 24,
## |   |   |   |   |   |   |   [166] humidity_9am > 55
## |   |   |   |   |   |   |   |   [167] wind_gust_speed <= 63: No (n = 374, err = 2
## |   |   |   |   |   |   |   |   [168] wind_gust_speed > 63: No (n = 399, err = 45
## |   |   |   |   |   [169] pressure_3pm > 1008.8
## |   |   |   |   |   |   [170] sunshine <= 9.1
```

```
## |   |   |   |   |   |   |   [171] wind_gust_speed <= 63
## |   |   |   |   |   |   |   |   [172] humidity_9am <= 68
## |   |   |   |   |   |   |   |   |   [173] wind_dir_9am <= NE: No (n = 283, err =
## |   |   |   |   |   |   |   |   |   [174] wind_dir_9am > NE
## |   |   |   |   |   |   |   |   |   |   [175] humidity_9am <= 42: No (n = 235, er
## |   |   |   |   |   |   |   |   |   |   [176] humidity_9am > 42: No (n = 598, er
## |   |   |   |   |   |   |   |   [177] humidity_9am > 68
## |   |   |   |   |   |   |   |   |   [178] temp_3pm <= 25.5: No (n = 232, err = 1
## |   |   |   |   |   |   |   |   |   [179] temp_3pm > 25.5: No (n = 62, err = 38.
## |   |   |   |   |   |   |   [180] wind_gust_speed > 63
## |   |   |   |   |   |   |   |   [181] temp_3pm <= 28.6
## |   |   |   |   |   |   |   |   |   [182] humidity_3pm <= 33: No (n = 223, err =
## |   |   |   |   |   |   |   |   |   [183] humidity_3pm > 33
## |   |   |   |   |   |   |   |   |   |   [184] pressure_3pm <= 1016.6
## |   |   |   |   |   |   |   |   |   |   |   [185] temp_3pm <= 20.9: No (n = 225,
## |   |   |   |   |   |   |   |   |   |   |   [186] temp_3pm > 20.9: No (n = 66, er
## |   |   |   |   |   |   |   |   |   |   [187] pressure_3pm > 1016.6: No (n = 69,
## |   |   |   |   |   |   |   |   [188] temp_3pm > 28.6
## |   |   |   |   |   |   |   |   |   [189] humidity_3pm <= 24: No (n = 100, err =
## |   |   |   |   |   |   |   |   |   [190] humidity_3pm > 24: Yes (n = 65, err = 
## |   |   |   |   |   |   [191] sunshine > 9.1
## |   |   |   |   |   |   |   [192] cloud_3pm <= 3
## |   |   |   |   |   |   |   |   [193] pressure_3pm <= 1011: No (n = 169, err = 7
## |   |   |   |   |   |   |   |   [194] pressure_3pm > 1011: No (n = 602, err = 1.
## |   |   |   |   |   |   |   [195] cloud_3pm > 3
## |   |   |   |   |   |   |   |   [196] humidity_9am <= 41: No (n = 157, err = 1.3
## |   |   |   |   |   |   |   |   [197] humidity_9am > 41: No (n = 359, err = 13.6
## |   |   [198] humidity_3pm > 50
## |   |   |   [199] wind_gust_speed <= 48
## |   |   |   |   [200] rain_today in No
## |   |   |   |   |   [201] pressure_3pm <= 1013.3
## |   |   |   |   |   |   [202] sunshine <= 7.4
## |   |   |   |   |   |   |   [203] wind_gust_speed <= 37
## |   |   |   |   |   |   |   |   [204] sunshine <= 0.2: Yes (n = 45, err = 42.2%)
## |   |   |   |   |   |   |   |   [205] sunshine > 0.2: No (n = 751, err = 29.7%)
## |   |   |   |   |   |   |   [206] wind_gust_speed > 37
```

```
## |   |   |   |   |   |   |   |   [207] pressure_3pm <= 1006.9: Yes (n = 200, err =
## |   |   |   |   |   |   |   |   [208] pressure_3pm > 1006.9: No (n = 526, err = 3
## |   |   |   |   |   |   [209] sunshine > 7.4
## |   |   |   |   |   |   |   [210] cloud_3pm <= 2
## |   |   |   |   |   |   |   |   [211] pressure_3pm <= 1009.1: No (n = 466, err =
## |   |   |   |   |   |   |   |   [212] pressure_3pm > 1009.1
## |   |   |   |   |   |   |   |   |   [213] sunshine <= 10.9
## |   |   |   |   |   |   |   |   |   |   [214] wind_gust_speed <= 37: No (n = 359,
## |   |   |   |   |   |   |   |   |   |   [215] wind_gust_speed > 37: No (n = 193,
## |   |   |   |   |   |   |   |   |   [216] sunshine > 10.9: No (n = 315, err = 1.6
## |   |   |   |   |   |   |   [217] cloud_3pm > 2
## |   |   |   |   |   |   |   |   [218] humidity_9am <= 78
## |   |   |   |   |   |   |   |   |   [219] pressure_9am <= 1012.2
## |   |   |   |   |   |   |   |   |   |   [220] rainfall <= 0.6
## |   |   |   |   |   |   |   |   |   |   |   [221] wind_gust_speed <= 35: No (n =
## |   |   |   |   |   |   |   |   |   |   |   [222] wind_gust_speed > 35
## |   |   |   |   |   |   |   |   |   |   |   |   [223] cloud_3pm <= 5: No (n = 704
## |   |   |   |   |   |   |   |   |   |   |   |   [224] cloud_3pm > 5: No (n = 230,
## |   |   |   |   |   |   |   |   |   |   [225] rainfall > 0.6: No (n = 56, err = 4
## |   |   |   |   |   |   |   |   |   [226] pressure_9am > 1012.2
## |   |   |   |   |   |   |   |   |   |   [227] max_temp <= 22.6
## |   |   |   |   |   |   |   |   |   |   |   [228] wind_gust_speed <= 30: No (n =
## |   |   |   |   |   |   |   |   |   |   |   [229] wind_gust_speed > 30: No (n = 2
## |   |   |   |   |   |   |   |   |   |   [230] max_temp > 22.6
## |   |   |   |   |   |   |   |   |   |   |   [231] humidity_3pm <= 54
## |   |   |   |   |   |   |   |   |   |   |   |   [232] wind_gust_dir <= SSE
## |   |   |   |   |   |   |   |   |   |   |   |   |   [233] humidity_3pm <= 51: No
## |   |   |   |   |   |   |   |   |   |   |   |   |   [234] humidity_3pm > 51: No (
## |   |   |   |   |   |   |   |   |   |   |   |   [235] wind_gust_dir > SSE: No (n
## |   |   |   |   |   |   |   |   |   |   |   [236] humidity_3pm > 54: No (n = 1099
## |   |   |   |   |   |   |   |   [237] humidity_9am > 78
## |   |   |   |   |   |   |   |   |   [238] pressure_3pm <= 1005.7: No (n = 136, er
## |   |   |   |   |   |   |   |   |   [239] pressure_3pm > 1005.7
## |   |   |   |   |   |   |   |   |   |   [240] max_temp <= 21
## |   |   |   |   |   |   |   |   |   |   |   [241] pressure_9am <= 1014.6
## |   |   |   |   |   |   |   |   |   |   |   |   [242] wind_dir_3pm <= SE: Yes (n
```

```
## |   |   |   |   |   |   |   |   |   |   |   [243] wind_dir_3pm > SE: No (n =
## |   |   |   |   |   |   |   |   |   |   [244] pressure_9am > 1014.6
## |   |   |   |   |   |   |   |   |   |   |   [245] humidity_3pm <= 60: No (n =
## |   |   |   |   |   |   |   |   |   |   |   [246] humidity_3pm > 60: Yes (n =
## |   |   |   |   |   |   |   |   |   [247] max_temp > 21
## |   |   |   |   |   |   |   |   |   |   [248] wind_gust_dir <= W: No (n = 44
## |   |   |   |   |   |   |   |   |   |   [249] wind_gust_dir > W
## |   |   |   |   |   |   |   |   |   |   |   [250] wind_dir_9am <= SE: Yes (n
## |   |   |   |   |   |   |   |   |   |   |   [251] wind_dir_9am > SE: No (n =
## |   |   |   |   |   [252] pressure_3pm > 1013.3
## |   |   |   |   |   |   [253] sunshine <= 4.6
## |   |   |   |   |   |   |   [254] pressure_3pm <= 1021.8
## |   |   |   |   |   |   |   |   [255] humidity_3pm <= 57: No (n = 463, err = 17.
## |   |   |   |   |   |   |   |   [256] humidity_3pm > 57
## |   |   |   |   |   |   |   |   |   [257] min_temp <= 13.5: No (n = 586, err = 3
## |   |   |   |   |   |   |   |   |   [258] min_temp > 13.5: No (n = 385, err = 23
## |   |   |   |   |   |   |   [259] pressure_3pm > 1021.8
## |   |   |   |   |   |   |   |   [260] temp_9am <= 20
## |   |   |   |   |   |   |   |   |   [261] wind_gust_speed <= 28: No (n = 494, er
## |   |   |   |   |   |   |   |   |   [262] wind_gust_speed > 28: No (n = 555, err
## |   |   |   |   |   |   |   |   [263] temp_9am > 20: No (n = 24, err = 33.3%)
## |   |   |   |   |   |   [264] sunshine > 4.6
## |   |   |   |   |   |   |   [265] humidity_3pm <= 65
## |   |   |   |   |   |   |   |   [266] pressure_3pm <= 1018.2
## |   |   |   |   |   |   |   |   |   [267] cloud_3pm <= 5
## |   |   |   |   |   |   |   |   |   |   [268] rainfall <= 0.1
## |   |   |   |   |   |   |   |   |   |   |   [269] humidity_9am <= 78
## |   |   |   |   |   |   |   |   |   |   |   |   [270] evaporation <= 5.8
## |   |   |   |   |   |   |   |   |   |   |   |   |   [271] humidity_3pm <= 55
## |   |   |   |   |   |   |   |   |   |   |   |   |   |   [272] cloud_9am <= 2: No
## |   |   |   |   |   |   |   |   |   |   |   |   |   |   [273] cloud_9am > 2: No
## |   |   |   |   |   |   |   |   |   |   |   |   |   [274] humidity_3pm > 55
## |   |   |   |   |   |   |   |   |   |   |   |   |   |   [275] wind_dir_3pm <= SS
## |   |   |   |   |   |   |   |   |   |   |   |   |   |   [276] wind_dir_3pm > SSW
## |   |   |   |   |   |   |   |   |   |   |   |   [277] evaporation > 5.8
## |   |   |   |   |   |   |   |   |   |   |   |   |   [278] sunshine <= 10.2
```

```
## |   |   |   |   |   |   |   |   |   |   |   |   |   [279] humidity_3pm <= 58
## |   |   |   |   |   |   |   |   |   |   |   |   |   [280] humidity_3pm > 58:
## |   |   |   |   |   |   |   |   |   |   |   |   |   [281] sunshine > 10.2: No (n
## |   |   |   |   |   |   |   |   |   |   |   [282] humidity_9am > 78
## |   |   |   |   |   |   |   |   |   |   |   |   [283] pressure_9am <= 1020.5
## |   |   |   |   |   |   |   |   |   |   |   |   |   [284] wind_gust_speed <= 33:
## |   |   |   |   |   |   |   |   |   |   |   |   |   [285] wind_gust_speed > 33: N
## |   |   |   |   |   |   |   |   |   |   |   |   [286] pressure_9am > 1020.5: No (
## |   |   |   |   |   |   |   |   |   |   [287] rainfall > 0.1
## |   |   |   |   |   |   |   |   |   |   |   [288] wind_gust_speed <= 35: No (n =
## |   |   |   |   |   |   |   |   |   |   |   [289] wind_gust_speed > 35
## |   |   |   |   |   |   |   |   |   |   |   |   [290] cloud_3pm <= 4: No (n = 139
## |   |   |   |   |   |   |   |   |   |   |   |   [291] cloud_3pm > 4: No (n = 546,
## |   |   |   |   |   |   |   |   |   [292] cloud_3pm > 5
## |   |   |   |   |   |   |   |   |   |   [293] evaporation <= 4.8: No (n = 909, er
## |   |   |   |   |   |   |   |   |   |   [294] evaporation > 4.8: No (n = 316, err
## |   |   |   |   |   |   |   |   [295] pressure_3pm > 1018.2
## |   |   |   |   |   |   |   |   |   [296] wind_gust_speed <= 26
## |   |   |   |   |   |   |   |   |   |   [297] pressure_3pm <= 1024.7: No (n = 134
## |   |   |   |   |   |   |   |   |   |   [298] pressure_3pm > 1024.7: No (n = 1138
## |   |   |   |   |   |   |   |   |   [299] wind_gust_speed > 26
## |   |   |   |   |   |   |   |   |   |   [300] sunshine <= 10.2
## |   |   |   |   |   |   |   |   |   |   |   [301] cloud_3pm <= 5
## |   |   |   |   |   |   |   |   |   |   |   |   [302] wind_dir_3pm <= ESE: No (n
## |   |   |   |   |   |   |   |   |   |   |   |   [303] wind_dir_3pm > ESE
## |   |   |   |   |   |   |   |   |   |   |   |   |   [304] temp_3pm <= 21.2
## |   |   |   |   |   |   |   |   |   |   |   |   |   |   [305] wind_dir_3pm <= WSW
## |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   [306] wind_gust_dir <
## |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   [307] wind_gust_dir >
## |   |   |   |   |   |   |   |   |   |   |   |   |   |   [308] wind_dir_3pm > WSW
## |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   [309] pressure_3pm <=
## |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   [310] temp_3pm <=
## |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   [311] temp_3pm >
## |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   [312] pressure_3pm >
## |   |   |   |   |   |   |   |   |   |   |   |   |   [313] temp_3pm > 21.2
## |   |   |   |   |   |   |   |   |   |   |   |   |   |   [314] rainfall <= 0.8: No
```

```
## |   |   |   |   |   |   |   |   |   |   |   |   |   [315] rainfall > 0.8: Yes
## |   |   |   |   |   |   |   |   |   |   |   [316] cloud_3pm > 5
## |   |   |   |   |   |   |   |   |   |   |   |   [317] temp_3pm <= 19.7: No (n = 7
## |   |   |   |   |   |   |   |   |   |   |   |   [318] temp_3pm > 19.7
## |   |   |   |   |   |   |   |   |   |   |   |   |   [319] pressure_3pm <= 1028: N
## |   |   |   |   |   |   |   |   |   |   |   |   |   [320] pressure_3pm > 1028: Ye
## |   |   |   |   |   |   |   |   |   |   [321] sunshine > 10.2: No (n = 550, err =
## |   |   |   |   |   |   |   [322] humidity_3pm > 65
## |   |   |   |   |   |   |   |   [323] pressure_3pm <= 1024.5
## |   |   |   |   |   |   |   |   |   [324] cloud_3pm <= 6
## |   |   |   |   |   |   |   |   |   |   [325] wind_dir_3pm <= ESE
## |   |   |   |   |   |   |   |   |   |   |   [326] wind_speed_3pm <= 15: No (n = 3
## |   |   |   |   |   |   |   |   |   |   |   [327] wind_speed_3pm > 15
## |   |   |   |   |   |   |   |   |   |   |   |   [328] max_temp <= 18.2: No (n = 4
## |   |   |   |   |   |   |   |   |   |   |   |   [329] max_temp > 18.2: No (n = 59
## |   |   |   |   |   |   |   |   |   |   [330] wind_dir_3pm > ESE: No (n = 1412, e
## |   |   |   |   |   |   |   |   |   |   [331] cloud_3pm > 6: No (n = 609, err = 24.1%
## |   |   |   |   |   |   |   |   |   [332] pressure_3pm > 1024.5
## |   |   |   |   |   |   |   |   |   |   [333] min_temp <= 4.3: No (n = 200, err = 1.0
## |   |   |   |   |   |   |   |   |   |   [334] min_temp > 4.3: No (n = 433, err = 12.2
## |   |   |   |   [335] rain_today in Yes
## |   |   |   |   |   [336] pressure_3pm <= 1015.4
## |   |   |   |   |   |   [337] sunshine <= 7.2
## |   |   |   |   |   |   |   [338] pressure_3pm <= 1007.4: Yes (n = 345, err = 40
## |   |   |   |   |   |   |   [339] pressure_3pm > 1007.4
## |   |   |   |   |   |   |   |   [340] wind_gust_speed <= 37: No (n = 372, err = 3
## |   |   |   |   |   |   |   |   [341] wind_gust_speed > 37: No (n = 476, err = 48
## |   |   |   |   |   |   [342] sunshine > 7.2
## |   |   |   |   |   |   |   [343] wind_gust_speed <= 37
## |   |   |   |   |   |   |   |   [344] humidity_3pm <= 61: No (n = 760, err = 20.8
## |   |   |   |   |   |   |   |   [345] humidity_3pm > 61
## |   |   |   |   |   |   |   |   |   [346] humidity_9am <= 76: No (n = 204, err =
## |   |   |   |   |   |   |   |   |   [347] humidity_9am > 76: No (n = 486, err = 3
## |   |   |   |   |   |   |   [348] wind_gust_speed > 37
## |   |   |   |   |   |   |   |   [349] humidity_9am <= 82
## |   |   |   |   |   |   |   |   |   [350] cloud_3pm <= 4: No (n = 188, err = 16.5
```

```
## |   |   |   |   |   |   |   |   [351] cloud_3pm > 4: No (n = 851, err = 34.0%
## |   |   |   |   |   |   |   [352] humidity_9am > 82
## |   |   |   |   |   |   |   |   [353] pressure_3pm <= 1007.1: Yes (n = 124, e
## |   |   |   |   |   |   |   |   [354] pressure_3pm > 1007.1
## |   |   |   |   |   |   |   |   |   [355] min_temp <= 9.2
## |   |   |   |   |   |   |   |   |   |   [356] pressure_3pm <= 1013.8: No (n =
## |   |   |   |   |   |   |   |   |   |   [357] pressure_3pm > 1013.8: No (n =
## |   |   |   |   |   |   |   |   |   [358] min_temp > 9.2: No (n = 515, err =
## |   |   |   |   [359] pressure_3pm > 1015.4
## |   |   |   |   |   [360] wind_gust_speed <= 33
## |   |   |   |   |   |   [361] min_temp <= 11
## |   |   |   |   |   |   |   [362] pressure_3pm <= 1022.3: No (n = 541, err =
## |   |   |   |   |   |   |   [363] pressure_3pm > 1022.3: No (n = 553, err = 1
## |   |   |   |   |   |   [364] min_temp > 11
## |   |   |   |   |   |   |   [365] sunshine <= 4.7: No (n = 166, err = 33.7%)
## |   |   |   |   |   |   |   [366] sunshine > 4.7
## |   |   |   |   |   |   |   |   [367] humidity_3pm <= 68: No (n = 506, err =
## |   |   |   |   |   |   |   |   [368] humidity_3pm > 68: No (n = 115, err = 2
## |   |   |   |   |   [369] wind_gust_speed > 33
## |   |   |   |   |   |   [370] humidity_3pm <= 63
## |   |   |   |   |   |   |   [371] rainfall <= 7.5
## |   |   |   |   |   |   |   |   [372] pressure_3pm <= 1018.8: No (n = 436, er
## |   |   |   |   |   |   |   |   [373] pressure_3pm > 1018.8: No (n = 795, err
## |   |   |   |   |   |   |   [374] rainfall > 7.5
## |   |   |   |   |   |   |   |   [375] temp_3pm <= 21: No (n = 315, err = 28.3
## |   |   |   |   |   |   |   |   [376] temp_3pm > 21: No (n = 123, err = 48.0%
## |   |   |   |   |   |   [377] humidity_3pm > 63
## |   |   |   |   |   |   |   [378] sunshine <= 4.8: No (n = 272, err = 49.3%)
## |   |   |   |   |   |   |   [379] sunshine > 4.8: No (n = 659, err = 34.9%)
## |   |   |   [380] wind_gust_speed > 48
## |   |   |   |   [381] pressure_3pm <= 1012.1
## |   |   |   |   |   [382] sunshine <= 6.8
## |   |   |   |   |   |   [383] wind_gust_speed <= 65
## |   |   |   |   |   |   |   [384] humidity_9am <= 70: Yes (n = 315, err = 42.2%)
## |   |   |   |   |   |   |   [385] humidity_9am > 70: Yes (n = 471, err = 29.7%)
## |   |   |   |   |   |   [386] wind_gust_speed > 65
```

```
## |   |   |   |   |   |   |   [387] pressure_3pm <= 1006.1: Yes (n = 227, err = 12
## |   |   |   |   |   |   |   [388] pressure_3pm > 1006.1
## |   |   |   |   |   |   |   |   [389] humidity_3pm <= 58: Yes (n = 64, err = 45.3
## |   |   |   |   |   |   |   |   [390] humidity_3pm > 58: Yes (n = 111, err = 17.1
## |   |   |   |   |   [391] sunshine > 6.8
## |   |   |   |   |   |   [392] humidity_9am <= 76
## |   |   |   |   |   |   |   [393] wind_dir_3pm <= SW
## |   |   |   |   |   |   |   |   [394] wind_speed_3pm <= 24: No (n = 290, err = 44
## |   |   |   |   |   |   |   |   [395] wind_speed_3pm > 24
## |   |   |   |   |   |   |   |   |   [396] wind_gust_speed <= 54: No (n = 252, err
## |   |   |   |   |   |   |   |   |   [397] wind_gust_speed > 54
## |   |   |   |   |   |   |   |   |   |   [398] cloud_3pm <= 5
## |   |   |   |   |   |   |   |   |   |   |   [399] rainfall <= 15: No (n = 230, er
## |   |   |   |   |   |   |   |   |   |   |   [400] rainfall > 15: Yes (n = 7, err
## |   |   |   |   |   |   |   |   |   |   [401] cloud_3pm > 5
## |   |   |   |   |   |   |   |   |   |   |   [402] wind_speed_3pm <= 35: Yes (n =
## |   |   |   |   |   |   |   |   |   |   |   [403] wind_speed_3pm > 35: No (n = 37
## |   |   |   |   |   |   |   [404] wind_dir_3pm > SW
## |   |   |   |   |   |   |   |   [405] humidity_3pm <= 57
## |   |   |   |   |   |   |   |   |   [406] rainfall <= 0.5: No (n = 185, err = 36
## |   |   |   |   |   |   |   |   |   [407] rainfall > 0.5: Yes (n = 128, err = 40
## |   |   |   |   |   |   |   |   [408] humidity_3pm > 57: Yes (n = 372, err = 35.2
## |   |   |   |   |   |   [409] humidity_9am > 76
## |   |   |   |   |   |   |   [410] wind_gust_speed <= 59: Yes (n = 488, err = 42.2
## |   |   |   |   |   |   |   [411] wind_gust_speed > 59: Yes (n = 354, err = 25.4%
## |   |   |   |   [412] pressure_3pm > 1012.1
## |   |   |   |   |   [413] rain_today in No
## |   |   |   |   |   |   [414] humidity_3pm <= 66
## |   |   |   |   |   |   |   [415] wind_gust_speed <= 56
## |   |   |   |   |   |   |   |   [416] sunshine <= 6.9: No (n = 319, err = 26.3%)
## |   |   |   |   |   |   |   |   [417] sunshine > 6.9
## |   |   |   |   |   |   |   |   |   [418] wind_speed_3pm <= 22: No (n = 285, err
## |   |   |   |   |   |   |   |   |   [419] wind_speed_3pm > 22
## |   |   |   |   |   |   |   |   |   |   [420] wind_dir_3pm <= SSE: No (n = 520, e
## |   |   |   |   |   |   |   |   |   |   [421] wind_dir_3pm > SSE: No (n = 295, er
## |   |   |   |   |   |   |   [422] wind_gust_speed > 56
```

```
## |   |   |   |   |   |   |   [423] wind_gust_dir <= W
## |   |   |   |   |   |   |   |   [424] sunshine <= 8.7
## |   |   |   |   |   |   |   |   |   [425] rainfall <= 0: No (n = 487, err = 2
## |   |   |   |   |   |   |   |   |   [426] rainfall > 0: No (n = 156, err = 42
## |   |   |   |   |   |   |   |   [427] sunshine > 8.7: No (n = 110, err = 10.0
## |   |   |   |   |   |   |   [428] wind_gust_dir > W: No (n = 135, err = 49.6%
## |   |   |   |   |   |   [429] humidity_3pm > 66
## |   |   |   |   |   |   |   [430] sunshine <= 7.1
## |   |   |   |   |   |   |   [431] wind_speed_9am <= 17: Yes (n = 50, err = 20
## |   |   |   |   |   |   |   [432] wind_speed_9am > 17
## |   |   |   |   |   |   |   |   [433] pressure_3pm <= 1017.2: Yes (n = 43, er
## |   |   |   |   |   |   |   |   [434] pressure_3pm > 1017.2: No (n = 46, err
## |   |   |   |   |   |   |   [435] sunshine > 7.1
## |   |   |   |   |   |   |   [436] rainfall <= 0.2
## |   |   |   |   |   |   |   |   [437] wind_gust_speed <= 54: No (n = 179, err
## |   |   |   |   |   |   |   |   [438] wind_gust_speed > 54: No (n = 175, err
## |   |   |   |   |   |   |   [439] rainfall > 0.2: No (n = 90, err = 50.0%)
## |   |   |   |   |   [440] rain_today in Yes
## |   |   |   |   |   |   [441] humidity_9am <= 84
## |   |   |   |   |   |   |   [442] humidity_3pm <= 66
## |   |   |   |   |   |   |   |   [443] humidity_9am <= 68: No (n = 402, err = 25.9
## |   |   |   |   |   |   |   |   [444] humidity_9am > 68: No (n = 605, err = 42.0%
## |   |   |   |   |   |   |   [445] humidity_3pm > 66: Yes (n = 288, err = 44.8%)
## |   |   |   |   |   |   [446] humidity_9am > 84
## |   |   |   |   |   |   |   [447] temp_9am <= 6.5: No (n = 45, err = 22.2%)
## |   |   |   |   |   |   |   [448] temp_9am > 6.5
## |   |   |   |   |   |   |   |   [449] sunshine <= 7.1: Yes (n = 186, err = 25.3%)
## |   |   |   |   |   |   |   |   [450] sunshine > 7.1
## |   |   |   |   |   |   |   |   |   [451] temp_9am <= 18.4: Yes (n = 245, err = 4
## |   |   |   |   |   |   |   |   |   [452] temp_9am > 18.4
## |   |   |   |   |   |   |   |   |   |   [453] cloud_3pm <= 5: Yes (n = 36, err =
## |   |   |   |   |   |   |   |   |   |   [454] cloud_3pm > 5: Yes (n = 12, err = 0
## |   [455] humidity_3pm > 72
## |   |   [456] humidity_3pm <= 83
## |   |   |   [457] wind_gust_speed <= 41
## |   |   |   |   [458] rain_today in No
```

```
## |   |   |   |   |   [459] cloud_3pm <= 6
## |   |   |   |   |   |   [460] pressure_3pm <= 1012.1
## |   |   |   |   |   |   |   [461] wind_speed_3pm <= 13
## |   |   |   |   |   |   |   |   [462] pressure_9am <= 1015: No (n = 125, err = 4
## |   |   |   |   |   |   |   |   [463] pressure_9am > 1015: Yes (n = 15, err = 6.
## |   |   |   |   |   |   |   [464] wind_speed_3pm > 13
## |   |   |   |   |   |   |   |   [465] rainfall <= 0
## |   |   |   |   |   |   |   |   |   [466] sunshine <= 5.7: Yes (n = 20, err = 45
## |   |   |   |   |   |   |   |   |   [467] sunshine > 5.7: No (n = 207, err = 22.
## |   |   |   |   |   |   |   |   [468] rainfall > 0: No (n = 59, err = 49.2%)
## |   |   |   |   |   |   [469] pressure_3pm > 1012.1
## |   |   |   |   |   |   |   [470] cloud_3pm <= 3
## |   |   |   |   |   |   |   |   [471] wind_gust_dir <= S: No (n = 99, err = 10.1
## |   |   |   |   |   |   |   |   [472] wind_gust_dir > S: No (n = 112, err = 0.0%
## |   |   |   |   |   |   |   [473] cloud_3pm > 3
## |   |   |   |   |   |   |   |   [474] pressure_3pm <= 1023.1
## |   |   |   |   |   |   |   |   |   [475] wind_speed_3pm <= 19
## |   |   |   |   |   |   |   |   |   |   [476] humidity_3pm <= 81
## |   |   |   |   |   |   |   |   |   |   |   [477] rainfall <= 0: No (n = 583, er
## |   |   |   |   |   |   |   |   |   |   |   [478] rainfall > 0
## |   |   |   |   |   |   |   |   |   |   |   |   [479] wind_gust_speed <= 30: No
## |   |   |   |   |   |   |   |   |   |   |   |   [480] wind_gust_speed > 30: No (
## |   |   |   |   |   |   |   |   |   |   [481] humidity_3pm > 81: No (n = 116, er
## |   |   |   |   |   |   |   |   |   [482] wind_speed_3pm > 19: No (n = 298, err =
## |   |   |   |   |   |   |   |   [483] pressure_3pm > 1023.1: No (n = 368, err = 1
## |   |   |   |   |   [484] cloud_3pm > 6
## |   |   |   |   |   |   [485] pressure_3pm <= 1020.4
## |   |   |   |   |   |   |   [486] pressure_3pm <= 1008.6: Yes (n = 165, err = 32
## |   |   |   |   |   |   |   [487] pressure_3pm > 1008.6
## |   |   |   |   |   |   |   |   [488] humidity_3pm <= 78: No (n = 519, err = 37.8
## |   |   |   |   |   |   |   |   [489] humidity_3pm > 78
## |   |   |   |   |   |   |   |   |   [490] humidity_9am <= 74: Yes (n = 80, err =
## |   |   |   |   |   |   |   |   |   [491] humidity_9am > 74
## |   |   |   |   |   |   |   |   |   |   [492] rainfall <= 0.4: No (n = 191, err =
## |   |   |   |   |   |   |   |   |   |   [493] rainfall > 0.4: Yes (n = 40, err =
## |   |   |   |   |   |   [494] pressure_3pm > 1020.4
```

```
## |   |   |   |   |   |   |   [495] humidity_3pm <= 77: No (n = 262, err = 18.3%)
## |   |   |   |   |   |   |   [496] humidity_3pm > 77: No (n = 179, err = 34.1%)
## |   |   |   |   [497] rain_today in Yes
## |   |   |   |   |   [498] pressure_3pm <= 1013.8
## |   |   |   |   |   |   [499] sunshine <= 6.1
## |   |   |   |   |   |   |   [500] evaporation <= 1.6: No (n = 79, err = 48.1%)
## |   |   |   |   |   |   |   [501] evaporation > 1.6
## |   |   |   |   |   |   |   |   [502] humidity_3pm <= 77: Yes (n = 150, err = 31
## |   |   |   |   |   |   |   |   [503] humidity_3pm > 77: Yes (n = 152, err = 13.2
## |   |   |   |   |   |   [504] sunshine > 6.1
## |   |   |   |   |   |   |   [505] cloud_3pm <= 6: No (n = 313, err = 49.8%)
## |   |   |   |   |   |   |   [506] cloud_3pm > 6: Yes (n = 150, err = 34.7%)
## |   |   |   |   |   [507] pressure_3pm > 1013.8
## |   |   |   |   |   |   [508] min_temp <= 10.8
## |   |   |   |   |   |   |   [509] wind_gust_speed <= 37: No (n = 472, err = 31.1%
## |   |   |   |   |   |   |   [510] wind_gust_speed > 37: Yes (n = 177, err = 49.2%
## |   |   |   |   |   |   [511] min_temp > 10.8
## |   |   |   |   |   |   |   [512] cloud_3pm <= 6
## |   |   |   |   |   |   |   |   [513] humidity_3pm <= 76: No (n = 227, err = 35.2
## |   |   |   |   |   |   |   |   [514] humidity_3pm > 76: Yes (n = 226, err = 45.1
## |   |   |   |   |   |   |   [515] cloud_3pm > 6: Yes (n = 354, err = 39.3%)
## |   |   |   [516] wind_gust_speed > 41
## |   |   |   |   [517] rain_today in No
## |   |   |   |   |   [518] sunshine <= 8.4
## |   |   |   |   |   |   [519] pressure_3pm <= 1013.8: Yes (n = 389, err = 22.6%)
## |   |   |   |   |   |   [520] pressure_3pm > 1013.8: Yes (n = 309, err = 42.1%)
## |   |   |   |   |   [521] sunshine > 8.4
## |   |   |   |   |   |   [522] wind_dir_3pm <= S
## |   |   |   |   |   |   |   [523] cloud_3pm <= 5
## |   |   |   |   |   |   |   |   [524] humidity_3pm <= 81
## |   |   |   |   |   |   |   |   |   [525] wind_gust_speed <= 59: No (n = 534, err
## |   |   |   |   |   |   |   |   |   [526] wind_gust_speed > 59: No (n = 112, err
## |   |   |   |   |   |   |   |   [527] humidity_3pm > 81: Yes (n = 66, err = 45.5%
## |   |   |   |   |   |   |   [528] cloud_3pm > 5
## |   |   |   |   |   |   |   |   [529] wind_gust_speed <= 46: No (n = 89, err = 38
## |   |   |   |   |   |   |   |   [530] wind_gust_speed > 46: Yes (n = 176, err = 3
```

```
## |   |   |   |   |   |   [531] wind_dir_3pm > S: Yes (n = 457, err = 42.5%)
## |   |   |   |   [532] rain_today in Yes
## |   |   |   |   |   [533] wind_gust_speed <= 57
## |   |   |   |   |   |   [534] sunshine <= 3: Yes (n = 282, err = 19.9%)
## |   |   |   |   |   |   [535] sunshine > 3
## |   |   |   |   |   |   |   [536] wind_speed_3pm <= 24
## |   |   |   |   |   |   |   |   [537] humidity_3pm <= 79: Yes (n = 444, err = 35
## |   |   |   |   |   |   |   |   [538] humidity_3pm > 79: Yes (n = 208, err = 22.(
## |   |   |   |   |   |   |   [539] wind_speed_3pm > 24
## |   |   |   |   |   |   |   |   [540] humidity_9am <= 86
## |   |   |   |   |   |   |   |   |   [541] cloud_3pm <= 6: No (n = 135, err = 36.:
## |   |   |   |   |   |   |   |   |   [542] cloud_3pm > 6: Yes (n = 86, err = 39.5%
## |   |   |   |   |   |   |   |   [543] humidity_9am > 86: Yes (n = 198, err = 33.:
## |   |   |   |   |   [544] wind_gust_speed > 57
## |   |   |   |   |   |   [545] sunshine <= 3.7: Yes (n = 154, err = 3.9%)
## |   |   |   |   |   |   [546] sunshine > 3.7: Yes (n = 526, err = 16.3%)
## |   |   [547] humidity_3pm > 83
## |   |   |   [548] pressure_3pm <= 1015.2
## |   |   |   |   [549] rain_today in No
## |   |   |   |   |   [550] temp_3pm <= 29.3
## |   |   |   |   |   |   [551] cloud_3pm <= 4: No (n = 24, err = 29.2%)
## |   |   |   |   |   |   [552] cloud_3pm > 4
## |   |   |   |   |   |   |   [553] sunshine <= 2.8
## |   |   |   |   |   |   |   |   [554] humidity_3pm <= 88: Yes (n = 194, err = 14
## |   |   |   |   |   |   |   |   [555] humidity_3pm > 88: Yes (n = 322, err = 6.8%
## |   |   |   |   |   |   |   [556] sunshine > 2.8
## |   |   |   |   |   |   |   |   [557] temp_3pm <= 16.4
## |   |   |   |   |   |   |   |   |   [558] pressure_3pm <= 1005.1: Yes (n = 111, e
## |   |   |   |   |   |   |   |   |   [559] pressure_3pm > 1005.1: Yes (n = 390, er
## |   |   |   |   |   |   |   |   [560] temp_3pm > 16.4
## |   |   |   |   |   |   |   |   |   [561] humidity_9am <= 80: Yes (n = 303, err =
## |   |   |   |   |   |   |   |   |   [562] humidity_9am > 80
## |   |   |   |   |   |   |   |   |   |   [563] temp_3pm <= 21.7: Yes (n = 150, err
## |   |   |   |   |   |   |   |   |   |   [564] temp_3pm > 21.7: No (n = 120, err =
## |   |   |   |   |   [565] temp_3pm > 29.3: No (n = 23, err = 4.3%)
## |   |   |   |   [566] rain_today in Yes
```

```
## |   |   |   |   |   [567] humidity_3pm <= 88
## |   |   |   |   |   |   [568] pressure_3pm <= 1005.1: Yes (n = 220, err = 2.7%)
## |   |   |   |   |   |   [569] pressure_3pm > 1005.1
## |   |   |   |   |   |   |   [570] wind_gust_speed <= 31: Yes (n = 100, err = 33.0
## |   |   |   |   |   |   |   [571] wind_gust_speed > 31: Yes (n = 505, err = 17.6%
## |   |   |   |   |   [572] humidity_3pm > 88
## |   |   |   |   |   |   [573] sunshine <= 3.9: Yes (n = 746, err = 3.1%)
## |   |   |   |   |   |   [574] sunshine > 3.9
## |   |   |   |   |   |   |   [575] wind_gust_speed <= 46
## |   |   |   |   |   |   |   |   [576] humidity_3pm <= 91: Yes (n = 122, err = 25
## |   |   |   |   |   |   |   |   [577] humidity_3pm > 91
## |   |   |   |   |   |   |   |   |   [578] wind_speed_3pm <= 24
## |   |   |   |   |   |   |   |   |   |   [579] sunshine <= 5: No (n = 8, err = 50
## |   |   |   |   |   |   |   |   |   |   [580] sunshine > 5
## |   |   |   |   |   |   |   |   |   |   |   [581] evaporation <= 6.1: Yes (n = 25
## |   |   |   |   |   |   |   |   |   |   |   [582] evaporation > 6.1: Yes (n = 8,
## |   |   |   |   |   |   |   |   |   [583] wind_speed_3pm > 24: Yes (n = 24, err =
## |   |   |   |   |   |   |   [584] wind_gust_speed > 46: Yes (n = 402, err = 3.5%)
## |   |   |   [585] pressure_3pm > 1015.2
## |   |   |   |   [586] min_temp <= 5
## |   |   |   |   |   [587] humidity_3pm <= 94
## |   |   |   |   |   |   [588] temp_9am <= 6.3: No (n = 278, err = 27.7%)
## |   |   |   |   |   |   [589] temp_9am > 6.3
## |   |   |   |   |   |   |   [590] rain_today in No: Yes (n = 135, err = 48.9%)
## |   |   |   |   |   |   |   [591] rain_today in Yes: Yes (n = 52, err = 21.2%)
## |   |   |   |   |   [592] humidity_3pm > 94
## |   |   |   |   |   |   [593] wind_gust_speed <= 46: Yes (n = 197, err = 47.2%)
## |   |   |   |   |   |   [594] wind_gust_speed > 46: Yes (n = 257, err = 30.7%)
## |   |   |   |   [595] min_temp > 5
## |   |   |   |   |   [596] rain_today in No
## |   |   |   |   |   |   [597] humidity_9am <= 96
## |   |   |   |   |   |   |   [598] humidity_3pm <= 88
## |   |   |   |   |   |   |   |   [599] humidity_9am <= 78: Yes (n = 182, err = 29
## |   |   |   |   |   |   |   |   [600] humidity_9am > 78: Yes (n = 255, err = 45.5
## |   |   |   |   |   |   |   [601] humidity_3pm > 88
## |   |   |   |   |   |   |   |   [602] cloud_3pm <= 3: No (n = 12, err = 16.7%)
```

```
## |   |   |   |   |   |   |   [603] cloud_3pm > 3: Yes (n = 593, err = 21.2%)
## |   |   |   |   |   [604] humidity_9am > 96
## |   |   |   |   |   |   [605] humidity_3pm <= 90
## |   |   |   |   |   |   |   [606] pressure_3pm <= 1015.5: Yes (n = 20, err =
## |   |   |   |   |   |   |   [607] pressure_3pm > 1015.5: No (n = 70, err = 18
## |   |   |   |   |   |   [608] humidity_3pm > 90: Yes (n = 146, err = 41.1%)
## |   |   |   |   [609] rain_today in Yes
## |   |   |   |   |   [610] humidity_3pm <= 90
## |   |   |   |   |   |   [611] wind_gust_speed <= 31: Yes (n = 201, err = 31.8
## |   |   |   |   |   |   [612] wind_gust_speed > 31
## |   |   |   |   |   |   |   [613] rainfall <= 12: Yes (n = 437, err = 21.1%)
## |   |   |   |   |   |   |   [614] rainfall > 12: Yes (n = 207, err = 6.8%)
## |   |   |   |   |   [615] humidity_3pm > 90
## |   |   |   |   |   |   [616] min_temp <= 10.5: Yes (n = 318, err = 15.7%)
## |   |   |   |   |   |   [617] min_temp > 10.5: Yes (n = 630, err = 8.6%)
##
## Number of inner nodes:    308
## Number of terminal nodes: 309
```

References

Hothorn, Torsten, and Achim Zeileis. 2021. *Partykit: A Toolkit for Recursive Partytioning*. http://partykit.r-forge.r-project.org/partykit/.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.64 Conditional Decision Tree Performance

Here we plot the performance of the decision tree, showing a risk chart. The areas under the recall and risk curves are also reported.

```
predicted <- predict(model, ds[te, vars], type="prob")[,2]
riskchart(predicted, actual_te, risk_te)
```

An error matrix shows, clockwise from the top left, the percentages of true negatives, false positives, true positives, and false negatives.

```
predicted <- predict(model, ds[te, vars], type="response")
sum(actual_te != predicted)/length(predicted) # Overall error rate
```

```
## [1] 0.1522272
```

```
round(100*table(actual_te, predicted, dnn=c("Actual", "Predicted"))/length(predicte
```

```
##       Predicted
## Actual No Yes
##    No  76   4
##    Yes 11   9
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.65 CTree Plot

```
plot(model)
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.66 Weka Decision Tree

The suite of algorithms implemented in Weka are also available to R thanks to (Hornik 2020).

```
library(RWeka)
model <- J48(formula=form, data=ds[tr, vars])
```

```
model
```

```
## J48 pruned tree
## ------------------
##
## humidity_3pm <= 66
## |   rain_today = No
## |   |   sunshine <= 7.7
## |   |   |   pressure_3pm <= 1013.7
## |   |   |   |   wind_gust_speed <= 54
## |   |   |   |   |   humidity_3pm <= 27: No (285.0/24.0)
## |   |   |   |   |   humidity_3pm > 27
## |   |   |   |   |   |   cloud_3pm <= 4: No (360.0/68.0)
## |   |   |   |   |   |   cloud_3pm > 4
## |   |   |   |   |   |   |   wind_gust_speed <= 37: No (960.0/267.0)
## |   |   |   |   |   |   |   wind_gust_speed > 37
## |   |   |   |   |   |   |   |   wind_dir_3pm = N
## |   |   |   |   |   |   |   |   |   wind_gust_dir = N
## |   |   |   |   |   |   |   |   |   |   cloud_3pm <= 7
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = N
## |   |   |   |   |   |   |   |   |   |   |   |   pressure_9am <= 1016.5: No (10.0/
## |   |   |   |   |   |   |   |   |   |   |   |   pressure_9am > 1016.5: Yes (8.0/1
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = NNE
## |   |   |   |   |   |   |   |   |   |   |   |   cloud_9am <= 6: Yes (6.0)
## |   |   |   |   |   |   |   |   |   |   |   |   cloud_9am > 6: No (4.0/1.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = NE: No (7.0/1.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = ENE: No (2.0/1.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = E: No (3.0/1.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = ESE: No (0.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = SE: No (0.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = SSE: No (1.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = S: No (0.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = SSW: No (0.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = SW: No (0.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = WSW: No (0.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = W: No (2.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = WNW: No (0.0)
```

```
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = NW: No (3.0/1.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = NNW
## |   |   |   |   |   |   |   |   |   |   |   cloud_9am <= 6: No (2.0)
## |   |   |   |   |   |   |   |   |   |   |   cloud_9am > 6
## |   |   |   |   |   |   |   |   |   |   |   |   humidity_3pm <= 42: No (3.0/1
## |   |   |   |   |   |   |   |   |   |   |   |   humidity_3pm > 42: Yes (2.0)
## |   |   |   |   |   |   |   |   |   cloud_3pm > 7: Yes (6.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NNE: Yes (16.0/6.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NE
## |   |   |   |   |   |   |   |   |   temp_3pm <= 23.1: No (2.0)
## |   |   |   |   |   |   |   |   |   temp_3pm > 23.1: Yes (4.0/1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = ENE: No (3.0/1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = E: No (1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = ESE: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SE: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SSE: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = S: Yes (2.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SSW: Yes (1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SW: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = WSW: Yes (3.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = W
## |   |   |   |   |   |   |   |   |   wind_speed_9am <= 20
## |   |   |   |   |   |   |   |   |   |   cloud_9am <= 6
## |   |   |   |   |   |   |   |   |   |   |   sunshine <= 4.8: No (3.0)
## |   |   |   |   |   |   |   |   |   |   |   sunshine > 4.8: Yes (3.0)
## |   |   |   |   |   |   |   |   |   |   cloud_9am > 6: Yes (10.0)
## |   |   |   |   |   |   |   |   |   wind_speed_9am > 20: No (4.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = WNW: No (2.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NW: Yes (9.0/3.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NNW
## |   |   |   |   |   |   |   |   |   pressure_9am <= 1010: Yes (4.0)
## |   |   |   |   |   |   |   |   |   pressure_9am > 1010: No (16.0/4.0)
## |   |   |   |   |   |   |   wind_dir_3pm = NNE: Yes (79.0/30.0)
## |   |   |   |   |   |   |   wind_dir_3pm = NE
## |   |   |   |   |   |   |   |   wind_dir_9am = N
## |   |   |   |   |   |   |   |   |   wind_speed_9am <= 7: Yes (2.0)
```

```
## |   |   |   |   |   |   |   |   |   wind_speed_9am > 7: No (6.0/1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = NNE: No (9.0/4.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = NE: Yes (10.0/1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = ENE
## |   |   |   |   |   |   |   |   |   wind_speed_9am <= 20: No (6.0)
## |   |   |   |   |   |   |   |   |   wind_speed_9am > 20: Yes (2.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = E
## |   |   |   |   |   |   |   |   |   temp_3pm <= 31.8: No (7.0)
## |   |   |   |   |   |   |   |   |   temp_3pm > 31.8: Yes (4.0/1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = ESE: No (2.0/1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = SE: No (2.0/1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = SSE: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = S: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = SSW: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = SW: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = WSW: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = W: Yes (3.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = WNW: Yes (1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = NW: Yes (3.0/1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = NNW
## |   |   |   |   |   |   |   |   |   wind_speed_3pm <= 9: Yes (2.0)
## |   |   |   |   |   |   |   |   |   wind_speed_3pm > 9: No (4.0/1.0)
## |   |   |   |   |   |   |   wind_dir_3pm = ENE: No (79.0/27.0)
## |   |   |   |   |   |   |   wind_dir_3pm = E: No (68.0/17.0)
## |   |   |   |   |   |   |   wind_dir_3pm = ESE: No (63.0/13.0)
## |   |   |   |   |   |   |   wind_dir_3pm = SE
## |   |   |   |   |   |   |   |   wind_dir_9am = N
## |   |   |   |   |   |   |   |   |   pressure_3pm <= 1010.7: Yes (5.0)
## |   |   |   |   |   |   |   |   |   pressure_3pm > 1010.7: No (3.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = NNE: No (3.0/1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = NE: No (1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = ENE: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = E
## |   |   |   |   |   |   |   |   |   cloud_3pm <= 6: No (5.0)
## |   |   |   |   |   |   |   |   |   cloud_3pm > 6: Yes (4.0/1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = ESE
```

```
## |   |   |   |   |   |   |   |   |   cloud_9am <= 5: Yes (3.0/1.0)
## |   |   |   |   |   |   |   |   |   cloud_9am > 5: No (5.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = SE: No (8.0/1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = SSE
## |   |   |   |   |   |   |   |   |   wind_gust_speed <= 39: No (8.0)
## |   |   |   |   |   |   |   |   |   wind_gust_speed > 39
## |   |   |   |   |   |   |   |   |   |   cloud_9am <= 5: Yes (3.0)
## |   |   |   |   |   |   |   |   |   |   cloud_9am > 5
## |   |   |   |   |   |   |   |   |   |   |   cloud_9am <= 6: No (5.0)
## |   |   |   |   |   |   |   |   |   |   |   cloud_9am > 6
## |   |   |   |   |   |   |   |   |   |   |   |   cloud_3pm <= 6: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   |   |   cloud_3pm > 6
## |   |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = N: No (0
## |   |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = NNE: No
## |   |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = NE: No (
## |   |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = ENE: No
## |   |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = E: No (0
## |   |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = ESE: No
## |   |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = SE: No (
## |   |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = SSE: Yes
## |   |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = S: No (0
## |   |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = SSW: No
## |   |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = SW: No (
## |   |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = WSW: No
## |   |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = W: No (0
## |   |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = WNW: No
## |   |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = NW: No (
## |   |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = NNW: No
## |   |   |   |   |   |   |   |   |   wind_dir_9am = S
## |   |   |   |   |   |   |   |   |   rainfall <= 0.1
## |   |   |   |   |   |   |   |   |   |   humidity_9am <= 75: Yes (5.0/1.0)
## |   |   |   |   |   |   |   |   |   |   humidity_9am > 75: No (3.0)
## |   |   |   |   |   |   |   |   |   rainfall > 0.1: No (2.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = SSW
## |   |   |   |   |   |   |   |   |   temp_9am <= 17.1: Yes (7.0)
## |   |   |   |   |   |   |   |   |   temp_9am > 17.1: No (6.0)
```

```
## |   |   |   |   |   |   |   |   wind_dir_9am = SW: Yes (4.0/1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = WSW: No (1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = W: No (2.0/1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = WNW: No (2.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = NW: No (3.0/1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = NNW
## |   |   |   |   |   |   |   |   |   humidity_3pm <= 49: No (2.0)
## |   |   |   |   |   |   |   |   |   humidity_3pm > 49: Yes (3.0)
## |   |   |   |   |   |   |   wind_dir_3pm = SSE: No (51.0/18.0)
## |   |   |   |   |   |   |   wind_dir_3pm = S: No (65.0/17.0)
## |   |   |   |   |   |   |   wind_dir_3pm = SSW: No (60.0/8.0)
## |   |   |   |   |   |   |   wind_dir_3pm = SW: No (88.0/24.0)
## |   |   |   |   |   |   |   wind_dir_3pm = WSW: No (82.0/20.0)
## |   |   |   |   |   |   |   wind_dir_3pm = W
## |   |   |   |   |   |   |   |   wind_gust_dir = N: Yes (4.0/1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NNE: No (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NE: Yes (1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = ENE: Yes (3.0/1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = E: No (2.0/1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = ESE: Yes (1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SE: No (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SSE: Yes (1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = S: No (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SSW: No (3.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SW: Yes (6.0/1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = WSW
## |   |   |   |   |   |   |   |   |   wind_speed_9am <= 13: Yes (4.0)
## |   |   |   |   |   |   |   |   |   wind_speed_9am > 13: No (16.0/4.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = W: No (40.0/13.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = WNW
## |   |   |   |   |   |   |   |   |   temp_3pm <= 13.4: Yes (4.0)
## |   |   |   |   |   |   |   |   |   temp_3pm > 13.4: No (10.0/2.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NW: No (4.0/1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NNW
## |   |   |   |   |   |   |   |   |   wind_speed_9am <= 13: No (2.0)
## |   |   |   |   |   |   |   |   |   wind_speed_9am > 13: Yes (2.0)
```

```
## |   |   |   |   |   |   |   wind_dir_3pm = WNW
## |   |   |   |   |   |   |   |   wind_gust_dir = N
## |   |   |   |   |   |   |   |   |   rainfall <= 0.2
## |   |   |   |   |   |   |   |   |   |   pressure_9am <= 1012.7: Yes (5.0/1.0
## |   |   |   |   |   |   |   |   |   |   pressure_9am > 1012.7: No (2.0)
## |   |   |   |   |   |   |   |   |   rainfall > 0.2: No (2.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NNE: No (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NE: Yes (1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = ENE: No (2.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = E: Yes (1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = ESE: Yes (4.0/1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SE: No (1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SSE: No (2.0/1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = S: No (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SSW: No (1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SW: Yes (4.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = WSW
## |   |   |   |   |   |   |   |   |   cloud_3pm <= 6: Yes (2.0)
## |   |   |   |   |   |   |   |   |   cloud_3pm > 6: No (3.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = W
## |   |   |   |   |   |   |   |   |   rainfall <= 0.1: No (20.0/7.0)
## |   |   |   |   |   |   |   |   |   rainfall > 0.1: Yes (4.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = WNW: No (31.0/10.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NW
## |   |   |   |   |   |   |   |   |   wind_speed_9am <= 17
## |   |   |   |   |   |   |   |   |   |   wind_speed_9am <= 4: No (2.0)
## |   |   |   |   |   |   |   |   |   |   wind_speed_9am > 4: Yes (7.0)
## |   |   |   |   |   |   |   |   |   wind_speed_9am > 17: No (7.0/1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NNW: Yes (3.0)
## |   |   |   |   |   |   |   wind_dir_3pm = NW
## |   |   |   |   |   |   |   |   wind_dir_9am = N
## |   |   |   |   |   |   |   |   |   wind_gust_speed <= 50: Yes (14.0/2.0)
## |   |   |   |   |   |   |   |   |   wind_gust_speed > 50: No (4.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = NNE
## |   |   |   |   |   |   |   |   |   rainfall <= 0.2: Yes (13.0/1.0)
## |   |   |   |   |   |   |   |   |   rainfall > 0.2: No (2.0)
```

```
## |   |   |   |   |   |   |   |   wind_dir_9am = NE: No (4.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = ENE
## |   |   |   |   |   |   |   |   |   temp_3pm <= 26.3: Yes (2.0)
## |   |   |   |   |   |   |   |   |   temp_3pm > 26.3: No (2.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = E: No (2.0/1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = ESE: Yes (4.0/1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = SE: No (1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = SSE: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = S: No (2.0/1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = SSW: Yes (1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = SW: Yes (1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = WSW: Yes (3.0/1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = W: Yes (3.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = WNW
## |   |   |   |   |   |   |   |   |   humidity_3pm <= 46: No (5.0)
## |   |   |   |   |   |   |   |   |   humidity_3pm > 46: Yes (6.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = NW: No (22.0/6.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = NNW
## |   |   |   |   |   |   |   |   |   wind_gust_dir = N
## |   |   |   |   |   |   |   |   |   |   temp_3pm <= 20.8: No (2.0)
## |   |   |   |   |   |   |   |   |   |   temp_3pm > 20.8: Yes (2.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = NNE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = NE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = ENE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = E: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = ESE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = SE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = SSE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = S: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = SSW: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = SW: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = WSW: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = W: Yes (3.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = WNW: No (3.0/1.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = NW: No (8.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = NNW
```

```
## |   |   |   |   |   |   |   |   |   |   cloud_9am <= 5: No (6.0)
## |   |   |   |   |   |   |   |   |   |   cloud_9am > 5: Yes (5.0/1.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = NNW
## |   |   |   |   |   |   |   |   |   wind_dir_9am = N
## |   |   |   |   |   |   |   |   |   |   max_temp <= 16.9
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_speed <= 43: Yes (8.0/3.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_speed > 43: No (14.0/3.0)
## |   |   |   |   |   |   |   |   |   |   max_temp > 16.9: Yes (29.0/4.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = NNE
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = N
## |   |   |   |   |   |   |   |   |   |   |   temp_3pm <= 18.1: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   |   temp_3pm > 18.1: No (4.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = NNE
## |   |   |   |   |   |   |   |   |   |   |   pressure_3pm <= 1011.4: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   |   pressure_3pm > 1011.4: No (2.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = NE: No (1.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = ENE: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = E: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = ESE: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = SE: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = SSE: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = S: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = SSW: No (1.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = SW: Yes (1.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = WSW: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = W: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = WNW: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = NW: Yes (3.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = NNW
## |   |   |   |   |   |   |   |   |   |   |   evaporation <= 2.3: Yes (3.0)
## |   |   |   |   |   |   |   |   |   |   |   evaporation > 2.3: No (7.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = NE
## |   |   |   |   |   |   |   |   |   |   humidity_3pm <= 49: Yes (5.0)
## |   |   |   |   |   |   |   |   |   |   humidity_3pm > 49: No (4.0/1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = ENE: Yes (7.0/1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = E: Yes (1.0)
```

```
## |   |   |   |   |   |   |   |   wind_dir_9am = ESE: Yes (1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = SE: Yes (1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = SSE: Yes (2.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = S: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = SSW: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = SW: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = WSW: Yes (2.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = W: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = WNW: Yes (3.0/1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = NW
## |   |   |   |   |   |   |   |   |   pressure_9am <= 1014.2: No (6.0)
## |   |   |   |   |   |   |   |   |   pressure_9am > 1014.2: Yes (2.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = NNW: No (24.0/6.0)
## |   |   |   |   wind_gust_speed > 54
## |   |   |   |   |   humidity_3pm <= 49
## |   |   |   |   |   |   pressure_3pm <= 1006.1
## |   |   |   |   |   |   |   wind_dir_9am = N
## |   |   |   |   |   |   |   |   min_temp <= 19
## |   |   |   |   |   |   |   |   wind_gust_dir = N
## |   |   |   |   |   |   |   |   |   rainfall <= 0.3
## |   |   |   |   |   |   |   |   |   |   evaporation <= 4.5: No (9.0)
## |   |   |   |   |   |   |   |   |   |   evaporation > 4.5
## |   |   |   |   |   |   |   |   |   |   |   cloud_3pm <= 5: No (7.0/1.0)
## |   |   |   |   |   |   |   |   |   |   |   cloud_3pm > 5
## |   |   |   |   |   |   |   |   |   |   |   |   temp_3pm <= 25.4: Yes (7.0)
## |   |   |   |   |   |   |   |   |   |   |   |   temp_3pm > 25.4
## |   |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_speed <= 72: Ye
## |   |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_speed > 72: No
## |   |   |   |   |   |   |   |   |   rainfall > 0.3: Yes (3.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NNE
## |   |   |   |   |   |   |   |   |   wind_speed_3pm <= 33: Yes (3.0)
## |   |   |   |   |   |   |   |   |   wind_speed_3pm > 33: No (4.0/1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NE: No (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = ENE: No (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = E: No (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = ESE: No (0.0)
```

```
## |   |   |   |   |   |   |   |   wind_gust_dir = SE: No (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SSE: No (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = S: No (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SSW: Yes (1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SW: No (2.0/1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = WSW: Yes (3.0/1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = W
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = N: No (2.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = NNE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = NE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = ENE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = E: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = ESE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = SE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = SSE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = S: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = SSW: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = SW: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = WSW: No (1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = W: No (4.0/1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = WNW: Yes (2.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = NW: No (1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = NNW: Yes (3.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = WNW
## |   |   |   |   |   |   |   |   |   temp_3pm <= 21.1: Yes (6.0)
## |   |   |   |   |   |   |   |   |   temp_3pm > 21.1: No (6.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NW
## |   |   |   |   |   |   |   |   |   humidity_3pm <= 29: No (3.0)
## |   |   |   |   |   |   |   |   |   humidity_3pm > 29: Yes (15.0/2.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NNW
## |   |   |   |   |   |   |   |   |   rainfall <= 0.1
## |   |   |   |   |   |   |   |   |   |   wind_speed_3pm <= 46
## |   |   |   |   |   |   |   |   |   |   |   wind_speed_9am <= 11: Yes (3.0/1
## |   |   |   |   |   |   |   |   |   |   |   wind_speed_9am > 11: No (10.0)
## |   |   |   |   |   |   |   |   |   |   wind_speed_3pm > 46: Yes (3.0)
## |   |   |   |   |   |   |   |   |   rainfall > 0.1: Yes (2.0)
```

```
## |   |   |   |   |   |   |   min_temp > 19: Yes (24.0/3.0)
## |   |   |   |   |   |   wind_dir_9am = NNE
## |   |   |   |   |   |   |   wind_dir_3pm = N
## |   |   |   |   |   |   |   |   humidity_3pm <= 22: No (5.0)
## |   |   |   |   |   |   |   |   humidity_3pm > 22
## |   |   |   |   |   |   |   |   |   pressure_3pm <= 1004.8: Yes (7.0)
## |   |   |   |   |   |   |   |   |   pressure_3pm > 1004.8: No (3.0)
## |   |   |   |   |   |   |   wind_dir_3pm = NNE: Yes (7.0)
## |   |   |   |   |   |   |   wind_dir_3pm = NE: Yes (1.0)
## |   |   |   |   |   |   |   wind_dir_3pm = ENE: Yes (0.0)
## |   |   |   |   |   |   |   wind_dir_3pm = E: Yes (0.0)
## |   |   |   |   |   |   |   wind_dir_3pm = ESE: Yes (0.0)
## |   |   |   |   |   |   |   wind_dir_3pm = SE: Yes (0.0)
## |   |   |   |   |   |   |   wind_dir_3pm = SSE: Yes (0.0)
## |   |   |   |   |   |   |   wind_dir_3pm = S: Yes (0.0)
## |   |   |   |   |   |   |   wind_dir_3pm = SSW: Yes (0.0)
## |   |   |   |   |   |   |   wind_dir_3pm = SW: Yes (1.0)
## |   |   |   |   |   |   |   wind_dir_3pm = WSW: Yes (2.0)
## |   |   |   |   |   |   |   wind_dir_3pm = W: Yes (2.0)
## |   |   |   |   |   |   |   wind_dir_3pm = WNW: No (5.0/1.0)
## |   |   |   |   |   |   |   wind_dir_3pm = NW
## |   |   |   |   |   |   |   |   pressure_3pm <= 1003.6: No (5.0)
## |   |   |   |   |   |   |   |   pressure_3pm > 1003.6: Yes (2.0)
## |   |   |   |   |   |   |   wind_dir_3pm = NNW
## |   |   |   |   |   |   |   |   temp_9am <= 25.1: Yes (3.0)
## |   |   |   |   |   |   |   |   temp_9am > 25.1: No (3.0)
## |   |   |   |   |   |   wind_dir_9am = NE: No (12.0/4.0)
## |   |   |   |   |   |   wind_dir_9am = ENE
## |   |   |   |   |   |   |   pressure_9am <= 1006.8: No (2.0)
## |   |   |   |   |   |   |   pressure_9am > 1006.8: Yes (4.0)
## |   |   |   |   |   |   wind_dir_9am = E
## |   |   |   |   |   |   |   cloud_3pm <= 6: Yes (4.0/1.0)
## |   |   |   |   |   |   |   cloud_3pm > 6: No (5.0)
## |   |   |   |   |   |   wind_dir_9am = ESE: No (6.0/1.0)
## |   |   |   |   |   |   wind_dir_9am = SE: Yes (0.0)
## |   |   |   |   |   |   wind_dir_9am = SSE: No (4.0/1.0)
```

```
## |   |   |   |   |   |   |   wind_dir_9am = S: No (3.0/1.0)
## |   |   |   |   |   |   |   wind_dir_9am = SSW: No (3.0)
## |   |   |   |   |   |   |   wind_dir_9am = SW: No (3.0/1.0)
## |   |   |   |   |   |   |   wind_dir_9am = WSW
## |   |   |   |   |   |   |   |   rainfall <= 0.1: No (3.0/1.0)
## |   |   |   |   |   |   |   |   rainfall > 0.1: Yes (3.0)
## |   |   |   |   |   |   |   wind_dir_9am = W
## |   |   |   |   |   |   |   |   rainfall <= 0.1: No (8.0/1.0)
## |   |   |   |   |   |   |   |   rainfall > 0.1: Yes (4.0)
## |   |   |   |   |   |   |   wind_dir_9am = WNW: Yes (17.0/7.0)
## |   |   |   |   |   |   |   wind_dir_9am = NW
## |   |   |   |   |   |   |   |   wind_gust_dir = N: No (4.0/1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NNE: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NE: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = ENE: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = E: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = ESE: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SE: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SSE: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = S: Yes (1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SSW: Yes (2.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SW: Yes (4.0/1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = WSW: No (2.0/1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = W: No (4.0/1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = WNW
## |   |   |   |   |   |   |   |   |   cloud_3pm <= 6
## |   |   |   |   |   |   |   |   |   |   cloud_9am <= 4: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   cloud_9am > 4: No (7.0)
## |   |   |   |   |   |   |   |   |   cloud_3pm > 6: Yes (3.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NW
## |   |   |   |   |   |   |   |   |   cloud_3pm <= 5: No (5.0/1.0)
## |   |   |   |   |   |   |   |   |   cloud_3pm > 5: Yes (10.0/3.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NNW: Yes (3.0)
## |   |   |   |   |   |   |   wind_dir_9am = NNW
## |   |   |   |   |   |   |   |   wind_gust_dir = N
## |   |   |   |   |   |   |   |   |   sunshine <= 5.9: Yes (3.0)
```

```
## |   |   |   |   |   |   |   |   sunshine > 5.9: No (4.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NNE: No (1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NE: No (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = ENE: No (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = E: No (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = ESE: No (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SE: No (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SSE: No (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = S: No (1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SSW: No (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SW: No (2.0/1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = WSW: No (1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = W
## |   |   |   |   |   |   |   |   |   humidity_3pm <= 32: No (4.0)
## |   |   |   |   |   |   |   |   |   humidity_3pm > 32: Yes (5.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = WNW: Yes (8.0/1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NW
## |   |   |   |   |   |   |   |   |   temp_9am <= 21.8
## |   |   |   |   |   |   |   |   |   |   cloud_3pm <= 3: No (2.0)
## |   |   |   |   |   |   |   |   |   |   cloud_3pm > 3
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_speed <= 76
## |   |   |   |   |   |   |   |   |   |   |   |   humidity_9am <= 71: No (4.0/1.0)
## |   |   |   |   |   |   |   |   |   |   |   |   humidity_9am > 71: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_speed > 76: Yes (6.0)
## |   |   |   |   |   |   |   |   |   temp_9am > 21.8: No (5.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NNW
## |   |   |   |   |   |   |   |   |   wind_speed_9am <= 11: Yes (2.0)
## |   |   |   |   |   |   |   |   |   wind_speed_9am > 11: No (11.0/1.0)
## |   |   |   |   |   |   pressure_3pm > 1006.1
## |   |   |   |   |   |   |   pressure_9am <= 1009.7: No (87.0/11.0)
## |   |   |   |   |   |   |   pressure_9am > 1009.7
## |   |   |   |   |   |   |   |   cloud_3pm <= 7
## |   |   |   |   |   |   |   |   |   humidity_3pm <= 19: No (58.0/7.0)
## |   |   |   |   |   |   |   |   |   humidity_3pm > 19
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = N
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = N
```

```
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = N
## |   |   |   |   |   |   |   |   |   |   |   |   sunshine <= 3.2
## |   |   |   |   |   |   |   |   |   |   |   |   |   cloud_3pm <= 6: Yes (3.0
## |   |   |   |   |   |   |   |   |   |   |   |   |   cloud_3pm > 6
## |   |   |   |   |   |   |   |   |   |   |   |   |   |   temp_3pm <= 17.2: Yes
## |   |   |   |   |   |   |   |   |   |   |   |   |   |   temp_3pm > 17.2: No (
## |   |   |   |   |   |   |   |   |   |   |   |   sunshine > 3.2
## |   |   |   |   |   |   |   |   |   |   |   |   |   evaporation <= 9.1: No (2
## |   |   |   |   |   |   |   |   |   |   |   |   |   evaporation > 9.1: Yes (2
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = NNE: No (5.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = NE: Yes (1.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = ENE: No (0.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = E: No (0.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = ESE: No (0.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = SE: No (0.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = SSE: No (0.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = S: No (0.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = SSW: No (0.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = SW: No (0.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = WSW: No (0.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = W: No (0.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = WNW: No (2.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = NW: Yes (1.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = NNW
## |   |   |   |   |   |   |   |   |   |   |   |   wind_speed_3pm <= 26: No (3.0
## |   |   |   |   |   |   |   |   |   |   |   |   wind_speed_3pm > 26: Yes (4.0
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = NNE
## |   |   |   |   |   |   |   |   |   |   |   wind_speed_3pm <= 22: Yes (4.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_speed_3pm > 22
## |   |   |   |   |   |   |   |   |   |   |   |   cloud_3pm <= 4: Yes (3.0/1.0)
## |   |   |   |   |   |   |   |   |   |   |   |   cloud_3pm > 4: No (10.0/1.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = NE
## |   |   |   |   |   |   |   |   |   |   |   pressure_9am <= 1013.5: Yes (2.0
## |   |   |   |   |   |   |   |   |   |   |   pressure_9am > 1013.5: No (4.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = ENE: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = E: No (0.0)
```

```
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = ESE: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = SE: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = SSE: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = S: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = SSW: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = SW: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = WSW: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = W: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = WNW: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = NW: Yes (3.0/1.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = NNW: No (1.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = NNE
## |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = N: No (6.0/2.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = NNE
## |   |   |   |   |   |   |   |   |   |   |   temp_9am <= 11: No (2.0)
## |   |   |   |   |   |   |   |   |   |   |   temp_9am > 11: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = NE: Yes (4.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = ENE: No (2.0/1.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = E: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = ESE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = SE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = SSE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = S: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = SSW: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = SW: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = WSW: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = W: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = WNW: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = NW: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = NNW: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = NE: No (5.0/1.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = ENE: No (2.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = E: No (11.0/3.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = ESE
## |   |   |   |   |   |   |   |   |   |   pressure_3pm <= 1010.2: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   pressure_3pm > 1010.2: No (2.0)
```

```
## |   |   |   |   |   |   |   |   |   wind_gust_dir = SE: No (2.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = SSE: Yes (3.0/1.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = S
## |   |   |   |   |   |   |   |   |   |   wind_speed_9am <= 17: Yes (3.0/1.0)
## |   |   |   |   |   |   |   |   |   |   wind_speed_9am > 17: No (5.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = SSW
## |   |   |   |   |   |   |   |   |   |   temp_3pm <= 24.2: No (2.0)
## |   |   |   |   |   |   |   |   |   |   temp_3pm > 24.2: Yes (4.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = SW: Yes (14.0/6.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = WSW: No (15.0/4.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = W
## |   |   |   |   |   |   |   |   |   |   pressure_9am <= 1011.3: No (4.0/1.0)
## |   |   |   |   |   |   |   |   |   |   pressure_9am > 1011.3: Yes (12.0/2.0
## |   |   |   |   |   |   |   |   |   wind_gust_dir = WNW
## |   |   |   |   |   |   |   |   |   |   pressure_9am <= 1014.3: No (13.0)
## |   |   |   |   |   |   |   |   |   |   pressure_9am > 1014.3: Yes (4.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = NW
## |   |   |   |   |   |   |   |   |   |   min_temp <= 7.1: Yes (6.0/1.0)
## |   |   |   |   |   |   |   |   |   |   min_temp > 7.1: No (21.0/6.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = NNW
## |   |   |   |   |   |   |   |   |   |   sunshine <= 3.3: Yes (8.0/1.0)
## |   |   |   |   |   |   |   |   |   |   sunshine > 3.3
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_speed <= 65: No (12.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_speed > 65
## |   |   |   |   |   |   |   |   |   |   |   |   pressure_3pm <= 1010.3: No (2
## |   |   |   |   |   |   |   |   |   |   |   |   pressure_3pm > 1010.3: Yes (3
## |   |   |   |   |   |   |   cloud_3pm > 7: Yes (27.0/8.0)
## |   |   |   |   |   humidity_3pm > 49
## |   |   |   |   |   |   wind_speed_9am <= 28: Yes (345.0/104.0)
## |   |   |   |   |   |   wind_speed_9am > 28
## |   |   |   |   |   |   |   wind_dir_9am = N
## |   |   |   |   |   |   |   |   wind_gust_dir = N
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = N
## |   |   |   |   |   |   |   |   |   cloud_3pm <= 4: No (5.0)
## |   |   |   |   |   |   |   |   |   cloud_3pm > 4
## |   |   |   |   |   |   |   |   |   |   humidity_3pm <= 53: Yes (5.0)
```

```
## |   |   |   |   |   |   |   |   |   |   humidity_3pm > 53
## |   |   |   |   |   |   |   |   |   |   |   max_temp <= 13.8: Yes (3.0)
## |   |   |   |   |   |   |   |   |   |   |   max_temp > 13.8
## |   |   |   |   |   |   |   |   |   |   |   |   evaporation <= 1.5: Yes (2.0
## |   |   |   |   |   |   |   |   |   |   |   |   evaporation > 1.5: No (13.0/1
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = NNE: No (2.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = NE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = ENE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = E: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = ESE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = SE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = SSE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = S: No (2.0/1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = SSW: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = SW: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = WSW: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = W: No (2.0/1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = WNW: No (3.0/1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = NW: Yes (3.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = NNW: Yes (2.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NNE: Yes (1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NE: No (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = ENE: No (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = E: No (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = ESE: No (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SE: No (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SSE: No (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = S: No (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SSW: No (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SW: No (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = WSW: Yes (1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = W: Yes (1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = WNW: Yes (2.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NW: No (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NNW: No (3.0/1.0)
## |   |   |   |   |   |   |   wind_dir_9am = NNE: Yes (5.0/1.0)
```

```
## |   |   |   |   |   |   |   wind_dir_9am = NE: No (3.0)
## |   |   |   |   |   |   |   wind_dir_9am = ENE: No (2.0/1.0)
## |   |   |   |   |   |   |   wind_dir_9am = E: Yes (4.0)
## |   |   |   |   |   |   |   wind_dir_9am = ESE: No (0.0)
## |   |   |   |   |   |   |   wind_dir_9am = SE: Yes (2.0)
## |   |   |   |   |   |   |   wind_dir_9am = SSE: No (2.0)
## |   |   |   |   |   |   |   wind_dir_9am = S: No (1.0)
## |   |   |   |   |   |   |   wind_dir_9am = SSW: Yes (1.0)
## |   |   |   |   |   |   |   wind_dir_9am = SW: No (3.0)
## |   |   |   |   |   |   |   wind_dir_9am = WSW
## |   |   |   |   |   |   |   |   temp_3pm <= 16.7: Yes (2.0)
## |   |   |   |   |   |   |   |   temp_3pm > 16.7: No (2.0)
## |   |   |   |   |   |   |   wind_dir_9am = W
## |   |   |   |   |   |   |   |   cloud_9am <= 6: No (7.0/1.0)
## |   |   |   |   |   |   |   |   cloud_9am > 6: Yes (3.0)
## |   |   |   |   |   |   |   wind_dir_9am = WNW: Yes (5.0/1.0)
## |   |   |   |   |   |   |   wind_dir_9am = NW
## |   |   |   |   |   |   |   |   rainfall <= 0.5: No (6.0/1.0)
## |   |   |   |   |   |   |   |   rainfall > 0.5: Yes (5.0)
## |   |   |   |   |   |   |   wind_dir_9am = NNW
## |   |   |   |   |   |   |   |   wind_gust_speed <= 78: No (11.0/1.0)
## |   |   |   |   |   |   |   |   wind_gust_speed > 78: Yes (3.0)
## |   |   |   pressure_3pm > 1013.7
## |   |   |   |   pressure_3pm <= 1021.8
## |   |   |   |   |   wind_gust_speed <= 46: No (3669.0/577.0)
## |   |   |   |   |   wind_gust_speed > 46
## |   |   |   |   |   |   wind_dir_9am = N
## |   |   |   |   |   |   |   wind_dir_3pm = N
## |   |   |   |   |   |   |   |   wind_speed_3pm <= 22
## |   |   |   |   |   |   |   |   |   wind_speed_3pm <= 17: No (4.0)
## |   |   |   |   |   |   |   |   |   wind_speed_3pm > 17: Yes (12.0/3.0)
## |   |   |   |   |   |   |   |   wind_speed_3pm > 22
## |   |   |   |   |   |   |   |   |   cloud_3pm <= 5: No (29.0/1.0)
## |   |   |   |   |   |   |   |   |   cloud_3pm > 5
## |   |   |   |   |   |   |   |   |   |   wind_speed_3pm <= 35: No (44.0/8.0)
## |   |   |   |   |   |   |   |   |   |   wind_speed_3pm > 35
```

```
## |   |   |   |   |   |   |   |   |   |   cloud_9am <= 6: No (6.0/1.0)
## |   |   |   |   |   |   |   |   |   |   cloud_9am > 6: Yes (9.0/1.0)
## |   |   |   |   |   |   |   wind_dir_3pm = NNE
## |   |   |   |   |   |   |   |   min_temp <= 12.2: No (13.0/2.0)
## |   |   |   |   |   |   |   |   min_temp > 12.2: Yes (3.0)
## |   |   |   |   |   |   |   wind_dir_3pm = NE: Yes (5.0/1.0)
## |   |   |   |   |   |   |   wind_dir_3pm = ENE: No (0.0)
## |   |   |   |   |   |   |   wind_dir_3pm = E: No (1.0)
## |   |   |   |   |   |   |   wind_dir_3pm = ESE: No (0.0)
## |   |   |   |   |   |   |   wind_dir_3pm = SE: No (2.0/1.0)
## |   |   |   |   |   |   |   wind_dir_3pm = SSE: No (4.0)
## |   |   |   |   |   |   |   wind_dir_3pm = S: No (3.0)
## |   |   |   |   |   |   |   wind_dir_3pm = SSW: No (5.0/1.0)
## |   |   |   |   |   |   |   wind_dir_3pm = SW
## |   |   |   |   |   |   |   |   wind_speed_3pm <= 20: Yes (2.0)
## |   |   |   |   |   |   |   |   wind_speed_3pm > 20: No (5.0)
## |   |   |   |   |   |   |   wind_dir_3pm = WSW: No (7.0)
## |   |   |   |   |   |   |   wind_dir_3pm = W
## |   |   |   |   |   |   |   |   wind_speed_9am <= 26
## |   |   |   |   |   |   |   |   |   evaporation <= 1.8: Yes (2.0)
## |   |   |   |   |   |   |   |   |   evaporation > 1.8: No (6.0)
## |   |   |   |   |   |   |   |   wind_speed_9am > 26: Yes (2.0)
## |   |   |   |   |   |   |   wind_dir_3pm = WNW
## |   |   |   |   |   |   |   |   humidity_3pm <= 52: No (7.0)
## |   |   |   |   |   |   |   |   humidity_3pm > 52: Yes (3.0)
## |   |   |   |   |   |   |   wind_dir_3pm = NW
## |   |   |   |   |   |   |   |   wind_gust_speed <= 61
## |   |   |   |   |   |   |   |   |   wind_speed_9am <= 9: No (2.0)
## |   |   |   |   |   |   |   |   |   wind_speed_9am > 9
## |   |   |   |   |   |   |   |   |   |   wind_speed_9am <= 26: Yes (12.0)
## |   |   |   |   |   |   |   |   |   |   wind_speed_9am > 26
## |   |   |   |   |   |   |   |   |   |   |   max_temp <= 15.2: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   |   max_temp > 15.2: No (2.0)
## |   |   |   |   |   |   |   |   wind_gust_speed > 61: No (7.0)
## |   |   |   |   |   |   |   wind_dir_3pm = NNW
## |   |   |   |   |   |   |   |   sunshine <= 0.3: Yes (3.0)
```

```
## |   |   |   |   |   |   |   sunshine > 0.3
## |   |   |   |   |   |   |   |   wind_speed_9am <= 28
## |   |   |   |   |   |   |   |   |   temp_9am <= 17.3
## |   |   |   |   |   |   |   |   |   |   pressure_9am <= 1018: No (5.0/1.0)
## |   |   |   |   |   |   |   |   |   |   pressure_9am > 1018: Yes (12.0/3.0)
## |   |   |   |   |   |   |   |   |   temp_9am > 17.3: No (4.0)
## |   |   |   |   |   |   |   |   wind_speed_9am > 28: No (10.0)
## |   |   |   |   |   |   wind_dir_9am = NNE
## |   |   |   |   |   |   |   cloud_3pm <= 7: No (64.0/14.0)
## |   |   |   |   |   |   |   cloud_3pm > 7: Yes (10.0/1.0)
## |   |   |   |   |   |   wind_dir_9am = NE
## |   |   |   |   |   |   |   min_temp <= 8.8: Yes (11.0)
## |   |   |   |   |   |   |   min_temp > 8.8
## |   |   |   |   |   |   |   |   wind_gust_speed <= 57: No (19.0/1.0)
## |   |   |   |   |   |   |   |   wind_gust_speed > 57: Yes (4.0/1.0)
## |   |   |   |   |   |   wind_dir_9am = ENE: No (29.0/12.0)
## |   |   |   |   |   |   wind_dir_9am = E
## |   |   |   |   |   |   |   humidity_9am <= 59: No (13.0)
## |   |   |   |   |   |   |   humidity_9am > 59
## |   |   |   |   |   |   |   |   cloud_3pm <= 6: Yes (4.0)
## |   |   |   |   |   |   |   |   cloud_3pm > 6: No (5.0/1.0)
## |   |   |   |   |   |   wind_dir_9am = ESE: No (16.0/2.0)
## |   |   |   |   |   |   wind_dir_9am = SE: No (41.0/8.0)
## |   |   |   |   |   |   wind_dir_9am = SSE
## |   |   |   |   |   |   |   cloud_9am <= 4: Yes (6.0/1.0)
## |   |   |   |   |   |   |   cloud_9am > 4: No (58.0/9.0)
## |   |   |   |   |   |   wind_dir_9am = S: No (44.0/2.0)
## |   |   |   |   |   |   wind_dir_9am = SSW
## |   |   |   |   |   |   |   wind_speed_9am <= 17
## |   |   |   |   |   |   |   |   max_temp <= 19.1: Yes (5.0/1.0)
## |   |   |   |   |   |   |   |   max_temp > 19.1: No (6.0)
## |   |   |   |   |   |   |   wind_speed_9am > 17: No (22.0)
## |   |   |   |   |   |   wind_dir_9am = SW: No (35.0/8.0)
## |   |   |   |   |   |   wind_dir_9am = WSW
## |   |   |   |   |   |   |   wind_gust_speed <= 54: No (44.0/1.0)
## |   |   |   |   |   |   |   wind_gust_speed > 54
```

```
## |   |   |   |   |   |   |   temp_9am <= 16.9
## |   |   |   |   |   |   |   |   humidity_9am <= 73: No (17.0)
## |   |   |   |   |   |   |   |   humidity_9am > 73
## |   |   |   |   |   |   |   |   |   wind_gust_dir = N: No (1.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = NNE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = NE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = ENE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = E: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = ESE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = SE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = SSE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = S: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = SSW: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = SW: Yes (3.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = WSW: No (2.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = W: Yes (1.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = WNW: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = NW: No (1.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = NNW: No (0.0)
## |   |   |   |   |   |   |   |   temp_9am > 16.9: Yes (4.0)
## |   |   |   |   |   |   wind_dir_9am = W: No (65.0/18.0)
## |   |   |   |   |   |   wind_dir_9am = WNW: No (45.0/12.0)
## |   |   |   |   |   |   wind_dir_9am = NW
## |   |   |   |   |   |   |   wind_speed_3pm <= 33: No (37.0/5.0)
## |   |   |   |   |   |   |   wind_speed_3pm > 33: Yes (3.0)
## |   |   |   |   |   |   wind_dir_9am = NNW: No (70.0/18.0)
## |   |   |   |   pressure_3pm > 1021.8
## |   |   |   |   |   temp_9am <= 18.9: No (2348.0/165.0)
## |   |   |   |   |   temp_9am > 18.9
## |   |   |   |   |   |   pressure_9am <= 1029.8: No (107.0/27.0)
## |   |   |   |   |   |   pressure_9am > 1029.8: Yes (7.0)
## |   |   sunshine > 7.7
## |   |   |   wind_gust_speed <= 50: No (61551.0/3872.0)
## |   |   |   wind_gust_speed > 50
## |   |   |   |   sunshine <= 10.6
## |   |   |   |   |   humidity_3pm <= 50
```

```
## |   |   |   |   |   |   wind_gust_speed <= 63: No (3979.0/475.0)
## |   |   |   |   |   |   wind_gust_speed > 63
## |   |   |   |   |   |   |   humidity_9am <= 54
## |   |   |   |   |   |   |   |   temp_3pm <= 22.2: No (398.0/34.0)
## |   |   |   |   |   |   |   |   temp_3pm > 22.2
## |   |   |   |   |   |   |   |   |   wind_gust_speed <= 81
## |   |   |   |   |   |   |   |   |   |   humidity_3pm <= 17: No (275.0/32.0)
## |   |   |   |   |   |   |   |   |   |   humidity_3pm > 17
## |   |   |   |   |   |   |   |   |   |   |   wind_speed_3pm <= 19
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = N
## |   |   |   |   |   |   |   |   |   |   |   |   |   cloud_9am <= 3: No (2.0)
## |   |   |   |   |   |   |   |   |   |   |   |   |   cloud_9am > 3: Yes (4.0)
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = NNE: Yes (3.0)
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = NE: Yes (3.0)
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = ENE
## |   |   |   |   |   |   |   |   |   |   |   |   |   temp_3pm <= 30.7: No (3.0)
## |   |   |   |   |   |   |   |   |   |   |   |   |   temp_3pm > 30.7: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = E: No (3.0/1.0)
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = ESE
## |   |   |   |   |   |   |   |   |   |   |   |   |   temp_3pm <= 35: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   |   |   |   temp_3pm > 35: No (2.0)
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = SE: Yes (5.0/1.0)
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = SSE: Yes (5.0/1.0
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = S: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = SSW: No (3.0/1.0)
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = SW: No (3.0)
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = WSW
## |   |   |   |   |   |   |   |   |   |   |   |   |   temp_3pm <= 32.4: No (2.0)
## |   |   |   |   |   |   |   |   |   |   |   |   |   temp_3pm > 32.4: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = W: No (5.0/1.0)
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = WNW
## |   |   |   |   |   |   |   |   |   |   |   |   |   cloud_3pm <= 3: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   |   |   |   cloud_3pm > 3: No (3.0)
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = NW: No (2.0/1.0)
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = NNW: No (2.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_speed_3pm > 19: No (245.0/55.0)
```

```
## |   |   |   |   |   |   |   |   wind_gust_speed > 81
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = N
## |   |   |   |   |   |   |   |   |   |   humidity_9am <= 34: Yes (5.0)
## |   |   |   |   |   |   |   |   |   |   humidity_9am > 34: No (5.0/1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = NNE: Yes (5.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = NE: Yes (1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = ENE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = E: Yes (3.0/1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = ESE: No (1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = SE: No (2.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = SSE: Yes (1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = S: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = SSW: Yes (1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = SW: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = WSW: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = W: No (9.0/2.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = WNW: No (19.0/2.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = NW
## |   |   |   |   |   |   |   |   |   |   wind_gust_speed <= 89
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_speed <= 83: No (8.0/3
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_speed > 83: Yes (11.0/3
## |   |   |   |   |   |   |   |   |   |   wind_gust_speed > 89: No (10.0/2.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = NNW
## |   |   |   |   |   |   |   |   |   |   wind_speed_9am <= 26
## |   |   |   |   |   |   |   |   |   |   |   cloud_3pm <= 5: No (8.0/1.0)
## |   |   |   |   |   |   |   |   |   |   |   cloud_3pm > 5: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   wind_speed_9am > 26: Yes (9.0)
## |   |   |   |   |   |   |   humidity_9am > 54
## |   |   |   |   |   |   |   |   pressure_3pm <= 1009.9
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = N
## |   |   |   |   |   |   |   |   |   |   evaporation <= 5.1
## |   |   |   |   |   |   |   |   |   |   |   sunshine <= 8.6: Yes (32.0/11.0)
## |   |   |   |   |   |   |   |   |   |   |   sunshine > 8.6: No (6.0/2.0)
## |   |   |   |   |   |   |   |   |   |   evaporation > 5.1: No (12.0/5.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = NNE
## |   |   |   |   |   |   |   |   |   |   wind_gust_speed <= 67: Yes (8.0)
```

```
## |   |   |   |   |   |   |   |   |   wind_gust_speed > 67: No (15.0/6.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = NE
## |   |   |   |   |   |   |   |   |   wind_speed_3pm <= 15: Yes (6.0)
## |   |   |   |   |   |   |   |   |   wind_speed_3pm > 15: No (8.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = ENE
## |   |   |   |   |   |   |   |   |   humidity_3pm <= 35: Yes (4.0)
## |   |   |   |   |   |   |   |   |   humidity_3pm > 35: No (2.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = E: Yes (3.0/1.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = ESE: Yes (1.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = SE: No (2.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = SSE: No (3.0/1.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = S: Yes (3.0/1.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = SSW: No (1.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = SW
## |   |   |   |   |   |   |   |   |   wind_speed_3pm <= 26: Yes (2.0)
## |   |   |   |   |   |   |   |   |   wind_speed_3pm > 26: No (7.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = WSW
## |   |   |   |   |   |   |   |   |   pressure_9am <= 1000.4: Yes (4.0)
## |   |   |   |   |   |   |   |   |   pressure_9am > 1000.4: No (25.0/4.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = W
## |   |   |   |   |   |   |   |   |   wind_gust_dir = N: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = NNE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = NE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = ENE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = E: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = ESE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = SE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = SSE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = S
## |   |   |   |   |   |   |   |   |   |   temp_3pm <= 34: No (2.0)
## |   |   |   |   |   |   |   |   |   |   temp_3pm > 34: Yes (2.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = SSW: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = SW: Yes (1.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = WSW: No (5.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = W: No (17.0/6.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = WNW
```

```
## |   |   |   |   |   |   |   |   |   |   rainfall <= 0.3: No (7.0/1.0)
## |   |   |   |   |   |   |   |   |   |   rainfall > 0.3: Yes (2.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = NW: Yes (2.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = NNW: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = WNW: No (68.0/24.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = NW
## |   |   |   |   |   |   |   |   |   wind_dir_9am = N: Yes (14.0/6.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = NNE: No (4.0/1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = NE: No (3.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = ENE: Yes (3.0/1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = E: No (3.0/1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = ESE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = SE: No (2.0/1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = SSE: No (1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = S: No (3.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = SSW: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = SW: No (1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = WSW: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = W: Yes (4.0/1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = WNW: Yes (3.0/1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = NW
## |   |   |   |   |   |   |   |   |   |   sunshine <= 8.7
## |   |   |   |   |   |   |   |   |   |   |   temp_9am <= 20.4: No (3.0)
## |   |   |   |   |   |   |   |   |   |   |   temp_9am > 20.4: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   sunshine > 8.7: Yes (2.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = NNW
## |   |   |   |   |   |   |   |   |   |   rainfall <= 0.1
## |   |   |   |   |   |   |   |   |   |   |   sunshine <= 8.7
## |   |   |   |   |   |   |   |   |   |   |   |   pressure_9am <= 1009: Yes (2
## |   |   |   |   |   |   |   |   |   |   |   |   pressure_9am > 1009: No (5.0
## |   |   |   |   |   |   |   |   |   |   |   sunshine > 8.7: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   rainfall > 0.1: Yes (3.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = NNW: Yes (39.0/9.0)
## |   |   |   |   |   |   |   pressure_3pm > 1009.9
## |   |   |   |   |   |   |   |   wind_speed_3pm <= 15: Yes (32.0/13.0)
## |   |   |   |   |   |   |   |   wind_speed_3pm > 15: No (234.0/40.0)
```

```
## |   |   |   |   |   humidity_3pm > 50
## |   |   |   |   |   |   pressure_3pm <= 1011.7
## |   |   |   |   |   |   |   humidity_9am <= 69
## |   |   |   |   |   |   |   |   max_temp <= 20.5
## |   |   |   |   |   |   |   |   |   wind_gust_dir = N
## |   |   |   |   |   |   |   |   |   |   temp_3pm <= 14.4: No (4.0)
## |   |   |   |   |   |   |   |   |   |   temp_3pm > 14.4: Yes (6.0/1.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = NNE
## |   |   |   |   |   |   |   |   |   |   cloud_3pm <= 7: Yes (3.0)
## |   |   |   |   |   |   |   |   |   |   cloud_3pm > 7: No (2.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = NE: Yes (1.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = ENE: Yes (1.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = E: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = ESE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = SE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = SSE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = S: No (2.0/1.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = SSW
## |   |   |   |   |   |   |   |   |   |   cloud_3pm <= 5: Yes (3.0)
## |   |   |   |   |   |   |   |   |   |   cloud_3pm > 5: No (3.0/1.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = SW
## |   |   |   |   |   |   |   |   |   |   pressure_3pm <= 1011.3: Yes (5.0/1.0)
## |   |   |   |   |   |   |   |   |   |   pressure_3pm > 1011.3: No (3.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = WSW: Yes (14.0/6.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = W
## |   |   |   |   |   |   |   |   |   |   humidity_9am <= 54: No (4.0)
## |   |   |   |   |   |   |   |   |   |   humidity_9am > 54: Yes (22.0/6.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = WNW
## |   |   |   |   |   |   |   |   |   |   evaporation <= 4.4: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   evaporation > 4.4
## |   |   |   |   |   |   |   |   |   |   |   min_temp <= 10.2: No (13.0/1.0)
## |   |   |   |   |   |   |   |   |   |   |   min_temp > 10.2
## |   |   |   |   |   |   |   |   |   |   |   |   pressure_3pm <= 1007.8: No (3.0/1
## |   |   |   |   |   |   |   |   |   |   |   |   pressure_3pm > 1007.8: Yes (4.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = NW
## |   |   |   |   |   |   |   |   |   |   humidity_9am <= 57: No (4.0)
```

```
## |   |   |   |   |   |   |   |   |   |   humidity_9am > 57: Yes (12.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = NNW: No (5.0/1.0)
## |   |   |   |   |   |   |   |   max_temp > 20.5
## |   |   |   |   |   |   |   |   |   cloud_3pm <= 5: No (265.0/65.0)
## |   |   |   |   |   |   |   |   |   cloud_3pm > 5
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = N
## |   |   |   |   |   |   |   |   |   |   |   pressure_9am <= 1008.7: Yes (6.0/1.0
## |   |   |   |   |   |   |   |   |   |   |   pressure_9am > 1008.7: No (7.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = NNE
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = N: No (1.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = NNE: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = NE: No (1.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = ENE: No (1.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = E: No (0.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = ESE: No (0.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = SE: No (0.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = SSE: No (1.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = S: No (0.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = SSW: Yes (1.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = SW: No (0.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = WSW: No (1.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = W: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = WNW: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = NW: No (0.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = NNW: No (2.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = NE: Yes (4.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = ENE: No (4.0/1.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = E
## |   |   |   |   |   |   |   |   |   |   |   wind_speed_9am <= 15: No (2.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_speed_9am > 15: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = ESE
## |   |   |   |   |   |   |   |   |   |   |   temp_9am <= 28.9: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   |   temp_9am > 28.9: No (2.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = SE
## |   |   |   |   |   |   |   |   |   |   |   temp_3pm <= 30: Yes (3.0)
## |   |   |   |   |   |   |   |   |   |   |   temp_3pm > 30: No (2.0)
```

```
## |   |   |   |   |   |   |   |   |   wind_dir_9am = SSE
## |   |   |   |   |   |   |   |   |   |   cloud_9am <= 5: No (5.0)
## |   |   |   |   |   |   |   |   |   |   cloud_9am > 5: Yes (4.0/1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = S: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = SSW: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = SW: Yes (3.0/1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = WSW: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = W
## |   |   |   |   |   |   |   |   |   |   temp_9am <= 18.7: No (3.0)
## |   |   |   |   |   |   |   |   |   |   temp_9am > 18.7: Yes (2.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = WNW
## |   |   |   |   |   |   |   |   |   |   cloud_3pm <= 6: Yes (4.0/1.0)
## |   |   |   |   |   |   |   |   |   |   cloud_3pm > 6: No (4.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = NW: No (5.0/1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = NNW
## |   |   |   |   |   |   |   |   |   |   wind_speed_9am <= 19: Yes (5.0)
## |   |   |   |   |   |   |   |   |   |   wind_speed_9am > 19: No (3.0/1.0)
## |   |   |   |   |   |   |   humidity_9am > 69
## |   |   |   |   |   |   |   |   wind_gust_dir = N
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = N
## |   |   |   |   |   |   |   |   |   wind_speed_3pm <= 28: Yes (5.0)
## |   |   |   |   |   |   |   |   |   wind_speed_3pm > 28
## |   |   |   |   |   |   |   |   |   |   humidity_9am <= 75: No (4.0)
## |   |   |   |   |   |   |   |   |   |   humidity_9am > 75: Yes (5.0/1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = NNE: Yes (1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = NE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = ENE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = E: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = ESE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = SE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = SSE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = S: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = SSW: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = SW: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = WSW: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = W: No (2.0)
```

```
## |   |   |   |   |   |   |   |   wind_dir_3pm = WNW: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = NW: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = NNW
## |   |   |   |   |   |   |   |   |   wind_gust_speed <= 61: No (5.0)
## |   |   |   |   |   |   |   |   |   wind_gust_speed > 61: Yes (2.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NNE: No (13.0/5.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NE: Yes (8.0/1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = ENE: No (2.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = E: Yes (10.0/2.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = ESE: Yes (6.0/1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SE
## |   |   |   |   |   |   |   |   |   evaporation <= 5.6
## |   |   |   |   |   |   |   |   |   |   humidity_3pm <= 56: Yes (3.0/1.0)
## |   |   |   |   |   |   |   |   |   |   humidity_3pm > 56: No (3.0)
## |   |   |   |   |   |   |   |   |   evaporation > 5.6: Yes (2.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SSE
## |   |   |   |   |   |   |   |   |   humidity_3pm <= 64: Yes (9.0)
## |   |   |   |   |   |   |   |   |   humidity_3pm > 64: No (5.0/1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = S: No (20.0/5.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SSW
## |   |   |   |   |   |   |   |   |   wind_speed_9am <= 15: Yes (7.0/1.0)
## |   |   |   |   |   |   |   |   |   wind_speed_9am > 15: No (8.0/1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SW
## |   |   |   |   |   |   |   |   |   pressure_3pm <= 1005.7: Yes (3.0)
## |   |   |   |   |   |   |   |   |   pressure_3pm > 1005.7: No (7.0/1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = WSW
## |   |   |   |   |   |   |   |   |   pressure_9am <= 1008.8
## |   |   |   |   |   |   |   |   |   |   wind_speed_9am <= 26: Yes (12.0/1.0)
## |   |   |   |   |   |   |   |   |   |   wind_speed_9am > 26: No (2.0)
## |   |   |   |   |   |   |   |   |   pressure_9am > 1008.8: No (5.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = W
## |   |   |   |   |   |   |   |   |   humidity_9am <= 81
## |   |   |   |   |   |   |   |   |   |   wind_speed_3pm <= 28: No (11.0/3.0)
## |   |   |   |   |   |   |   |   |   |   wind_speed_3pm > 28: Yes (7.0/2.0)
## |   |   |   |   |   |   |   |   |   humidity_9am > 81: Yes (15.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = WNW: Yes (28.0/8.0)
```

```
## |   |   |   |   |   |   |   |   wind_gust_dir = NW
## |   |   |   |   |   |   |   |   |   evaporation <= 6
## |   |   |   |   |   |   |   |   |   |   cloud_3pm <= 3
## |   |   |   |   |   |   |   |   |   |   |   temp_3pm <= 15.1: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   |   temp_3pm > 15.1: No (3.0)
## |   |   |   |   |   |   |   |   |   |   cloud_3pm > 3
## |   |   |   |   |   |   |   |   |   |   |   rainfall <= 0.7: Yes (25.0/2.0)
## |   |   |   |   |   |   |   |   |   |   |   rainfall > 0.7: No (3.0/1.0)
## |   |   |   |   |   |   |   |   |   evaporation > 6: No (2.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NNW
## |   |   |   |   |   |   |   |   |   pressure_9am <= 1013.9
## |   |   |   |   |   |   |   |   |   |   pressure_9am <= 1005.6: Yes (3.0)
## |   |   |   |   |   |   |   |   |   |   pressure_9am > 1005.6: No (7.0)
## |   |   |   |   |   |   |   |   |   pressure_9am > 1013.9: Yes (10.0)
## |   |   |   |   |   |   pressure_3pm > 1011.7
## |   |   |   |   |   |   |   wind_gust_speed <= 65
## |   |   |   |   |   |   |   |   rainfall <= 0.1: No (828.0/142.0)
## |   |   |   |   |   |   |   |   rainfall > 0.1
## |   |   |   |   |   |   |   |   |   wind_dir_9am = N
## |   |   |   |   |   |   |   |   |   |   cloud_9am <= 4: No (4.0)
## |   |   |   |   |   |   |   |   |   |   cloud_9am > 4: Yes (16.0/5.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = NNE
## |   |   |   |   |   |   |   |   |   |   temp_9am <= 22.9: No (6.0)
## |   |   |   |   |   |   |   |   |   |   temp_9am > 22.9: Yes (3.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = NE: Yes (2.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = ENE: Yes (1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = E
## |   |   |   |   |   |   |   |   |   |   evaporation <= 6.3: No (8.0)
## |   |   |   |   |   |   |   |   |   |   evaporation > 6.3
## |   |   |   |   |   |   |   |   |   |   |   wind_speed_9am <= 28: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_speed_9am > 28: No (2.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = ESE: No (9.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = SE
## |   |   |   |   |   |   |   |   |   |   sunshine <= 8.8: No (17.0/3.0)
## |   |   |   |   |   |   |   |   |   |   sunshine > 8.8: Yes (2.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = SSE
```

```
## |   |   |   |   |   |   |   |   |   |   sunshine <= 8.6
## |   |   |   |   |   |   |   |   |   |   |   rainfall <= 0.7
## |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_speed <= 54: No (12.0/2
## |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_speed > 54: Yes (5.0/1
## |   |   |   |   |   |   |   |   |   |   |   rainfall > 0.7: No (12.0)
## |   |   |   |   |   |   |   |   |   |   sunshine > 8.6: Yes (2.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = S: No (22.0/4.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = SSW: No (18.0/2.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = SW
## |   |   |   |   |   |   |   |   |   |   sunshine <= 8.9
## |   |   |   |   |   |   |   |   |   |   |   humidity_9am <= 55: Yes (3.0/1.0)
## |   |   |   |   |   |   |   |   |   |   |   humidity_9am > 55: No (17.0)
## |   |   |   |   |   |   |   |   |   |   sunshine > 8.9: Yes (2.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = WSW
## |   |   |   |   |   |   |   |   |   |   humidity_3pm <= 54: Yes (6.0/2.0)
## |   |   |   |   |   |   |   |   |   |   humidity_3pm > 54: No (10.0/3.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = W
## |   |   |   |   |   |   |   |   |   |   humidity_9am <= 65: Yes (3.0/1.0)
## |   |   |   |   |   |   |   |   |   |   humidity_9am > 65: No (15.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = WNW
## |   |   |   |   |   |   |   |   |   |   sunshine <= 8.6: Yes (13.0/4.0)
## |   |   |   |   |   |   |   |   |   |   sunshine > 8.6: No (2.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = NW
## |   |   |   |   |   |   |   |   |   |   max_temp <= 14.2: Yes (5.0)
## |   |   |   |   |   |   |   |   |   |   max_temp > 14.2: No (8.0/1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = NNW
## |   |   |   |   |   |   |   |   |   |   sunshine <= 8.4: No (5.0)
## |   |   |   |   |   |   |   |   |   |   sunshine > 8.4: Yes (8.0/2.0)
## |   |   |   |   |   |   |   wind_gust_speed > 65
## |   |   |   |   |   |   |   |   wind_gust_dir = N
## |   |   |   |   |   |   |   |   |   max_temp <= 23.7: No (18.0/2.0)
## |   |   |   |   |   |   |   |   |   max_temp > 23.7: Yes (2.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NNE
## |   |   |   |   |   |   |   |   |   wind_speed_3pm <= 37: Yes (2.0)
## |   |   |   |   |   |   |   |   |   wind_speed_3pm > 37: No (7.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NE: No (2.0)
```

```
## |   |   |   |   |   |   |   wind_gust_dir = ENE: No (1.0)
## |   |   |   |   |   |   |   wind_gust_dir = E: No (0.0)
## |   |   |   |   |   |   |   wind_gust_dir = ESE: No (0.0)
## |   |   |   |   |   |   |   wind_gust_dir = SE: Yes (2.0)
## |   |   |   |   |   |   |   wind_gust_dir = SSE
## |   |   |   |   |   |   |   |   humidity_9am <= 60: No (8.0)
## |   |   |   |   |   |   |   |   humidity_9am > 60: Yes (6.0)
## |   |   |   |   |   |   |   wind_gust_dir = S
## |   |   |   |   |   |   |   |   cloud_3pm <= 5: No (26.0/9.0)
## |   |   |   |   |   |   |   |   cloud_3pm > 5: Yes (5.0)
## |   |   |   |   |   |   |   wind_gust_dir = SSW
## |   |   |   |   |   |   |   |   sunshine <= 8.7
## |   |   |   |   |   |   |   |   |   wind_speed_9am <= 24: Yes (5.0)
## |   |   |   |   |   |   |   |   |   wind_speed_9am > 24
## |   |   |   |   |   |   |   |   |   |   humidity_3pm <= 64: No (6.0)
## |   |   |   |   |   |   |   |   |   |   humidity_3pm > 64: Yes (2.0)
## |   |   |   |   |   |   |   |   sunshine > 8.7: No (2.0)
## |   |   |   |   |   |   |   wind_gust_dir = SW: No (9.0/2.0)
## |   |   |   |   |   |   |   wind_gust_dir = WSW: No (12.0/4.0)
## |   |   |   |   |   |   |   wind_gust_dir = W: No (21.0/4.0)
## |   |   |   |   |   |   |   wind_gust_dir = WNW: No (13.0/5.0)
## |   |   |   |   |   |   |   wind_gust_dir = NW: Yes (10.0/3.0)
## |   |   |   |   |   |   |   wind_gust_dir = NNW
## |   |   |   |   |   |   |   |   pressure_9am <= 1016.5: No (2.0)
## |   |   |   |   |   |   |   |   pressure_9am > 1016.5
## |   |   |   |   |   |   |   |   |   wind_gust_speed <= 67: No (2.0)
## |   |   |   |   |   |   |   |   |   wind_gust_speed > 67: Yes (7.0)
## |   |   |   |   sunshine > 10.6: No (1989.0/95.0)
## |   rain_today = Yes
## |   |   sunshine <= 9
## |   |   |   wind_gust_speed <= 46
## |   |   |   |   humidity_3pm <= 50
## |   |   |   |   |   sunshine <= 6.4
## |   |   |   |   |   |   wind_gust_speed <= 37: No (132.0/16.0)
## |   |   |   |   |   |   wind_gust_speed > 37
## |   |   |   |   |   |   |   wind_dir_9am = N
```

```
## |   |   |   |   |   |   |   |   cloud_3pm <= 4
## |   |   |   |   |   |   |   |   |   cloud_9am <= 6: No (2.0)
## |   |   |   |   |   |   |   |   |   cloud_9am > 6
## |   |   |   |   |   |   |   |   |   |   temp_3pm <= 22.5: Yes (4.0/1.0)
## |   |   |   |   |   |   |   |   |   |   temp_3pm > 22.5: No (2.0)
## |   |   |   |   |   |   |   |   cloud_3pm > 4: Yes (10.0)
## |   |   |   |   |   |   |   wind_dir_9am = NNE
## |   |   |   |   |   |   |   |   wind_speed_3pm <= 15: No (4.0)
## |   |   |   |   |   |   |   |   wind_speed_3pm > 15
## |   |   |   |   |   |   |   |   |   humidity_3pm <= 49: Yes (6.0)
## |   |   |   |   |   |   |   |   |   humidity_3pm > 49: No (2.0)
## |   |   |   |   |   |   |   wind_dir_9am = NE
## |   |   |   |   |   |   |   |   wind_gust_speed <= 41: No (4.0)
## |   |   |   |   |   |   |   |   wind_gust_speed > 41: Yes (5.0)
## |   |   |   |   |   |   |   wind_dir_9am = ENE
## |   |   |   |   |   |   |   |   wind_speed_9am <= 7: No (2.0)
## |   |   |   |   |   |   |   |   wind_speed_9am > 7
## |   |   |   |   |   |   |   |   |   wind_speed_9am <= 17: Yes (6.0)
## |   |   |   |   |   |   |   |   |   wind_speed_9am > 17: No (2.0)
## |   |   |   |   |   |   |   wind_dir_9am = E: Yes (4.0/1.0)
## |   |   |   |   |   |   |   wind_dir_9am = ESE: No (5.0)
## |   |   |   |   |   |   |   wind_dir_9am = SE: No (7.0/1.0)
## |   |   |   |   |   |   |   wind_dir_9am = SSE: No (10.0/1.0)
## |   |   |   |   |   |   |   wind_dir_9am = S
## |   |   |   |   |   |   |   |   humidity_3pm <= 45: No (2.0)
## |   |   |   |   |   |   |   |   humidity_3pm > 45: Yes (3.0)
## |   |   |   |   |   |   |   wind_dir_9am = SSW: No (13.0/1.0)
## |   |   |   |   |   |   |   wind_dir_9am = SW
## |   |   |   |   |   |   |   |   cloud_3pm <= 6: Yes (8.0/2.0)
## |   |   |   |   |   |   |   |   cloud_3pm > 6: No (6.0)
## |   |   |   |   |   |   |   wind_dir_9am = WSW
## |   |   |   |   |   |   |   |   cloud_9am <= 5: Yes (3.0/1.0)
## |   |   |   |   |   |   |   |   cloud_9am > 5: No (7.0)
## |   |   |   |   |   |   |   wind_dir_9am = W: No (12.0/3.0)
## |   |   |   |   |   |   |   wind_dir_9am = WNW
## |   |   |   |   |   |   |   |   humidity_9am <= 88: No (5.0)
```

```
## |   |   |   |   |   |   |   humidity_9am > 88: Yes (2.0)
## |   |   |   |   |   |   |   wind_dir_9am = NW: No (6.0/1.0)
## |   |   |   |   |   |   |   wind_dir_9am = NNW
## |   |   |   |   |   |   |   |   pressure_3pm <= 1005.7: Yes (2.0)
## |   |   |   |   |   |   |   |   pressure_3pm > 1005.7: No (5.0)
## |   |   |   |   |   sunshine > 6.4: No (1908.0/172.0)
## |   |   |   |   humidity_3pm > 50
## |   |   |   |   |   wind_gust_speed <= 31: No (1757.0/309.0)
## |   |   |   |   |   wind_gust_speed > 31
## |   |   |   |   |   |   pressure_3pm <= 1014.3
## |   |   |   |   |   |   |   humidity_3pm <= 51
## |   |   |   |   |   |   |   |   max_temp <= 21.5
## |   |   |   |   |   |   |   |   |   wind_gust_dir = N: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = NNE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = NE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = ENE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = E: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = ESE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = SE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = SSE: Yes (1.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = S: No (2.0/1.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = SSW: Yes (3.0/1.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = SW: Yes (2.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = WSW
## |   |   |   |   |   |   |   |   |   |   cloud_3pm <= 6: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   cloud_3pm > 6: No (2.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = W
## |   |   |   |   |   |   |   |   |   |   temp_3pm <= 14: No (5.0/1.0)
## |   |   |   |   |   |   |   |   |   |   temp_3pm > 14: Yes (74.0/11.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = WNW: Yes (4.0/1.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = NW: No (1.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = NNW: No (4.0)
## |   |   |   |   |   |   |   |   max_temp > 21.5
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = N
## |   |   |   |   |   |   |   |   |   |   pressure_3pm <= 1002.7: Yes (3.0)
## |   |   |   |   |   |   |   |   |   |   pressure_3pm > 1002.7: No (3.0)
```

```
## |   |   |   |   |   |   |   |   wind_dir_3pm = NNE
## |   |   |   |   |   |   |   |   |   temp_3pm <= 28: No (3.0)
## |   |   |   |   |   |   |   |   |   temp_3pm > 28: Yes (2.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = NE
## |   |   |   |   |   |   |   |   |   rainfall <= 6.9: No (2.0)
## |   |   |   |   |   |   |   |   |   rainfall > 6.9: Yes (2.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = ENE: Yes (3.0/1.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = E
## |   |   |   |   |   |   |   |   |   rainfall <= 11.4: Yes (4.0)
## |   |   |   |   |   |   |   |   |   rainfall > 11.4: No (2.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = ESE: No (8.0/1.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = SE
## |   |   |   |   |   |   |   |   |   min_temp <= 18.1
## |   |   |   |   |   |   |   |   |   |   min_temp <= 14.5: No (8.0/1.0)
## |   |   |   |   |   |   |   |   |   |   min_temp > 14.5: Yes (11.0/2.0)
## |   |   |   |   |   |   |   |   |   min_temp > 18.1: No (12.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = SSE: No (6.0/1.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = S: Yes (3.0/1.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = SSW: No (2.0/1.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = SW: No (4.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = WSW: No (2.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = W: No (5.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = WNW
## |   |   |   |   |   |   |   |   |   wind_dir_9am = N: Yes (2.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = NNE: Yes (1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = NE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = ENE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = E: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = ESE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = SE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = SSE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = S: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = SSW: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = SW: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = WSW: Yes (1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = W: Yes (1.0)
```

```
## |   |   |   |   |   |   |   |   |   wind_dir_9am = WNW: No (3.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = NW
## |   |   |   |   |   |   |   |   |   |   pressure_3pm <= 1007.5: No (3.0)
## |   |   |   |   |   |   |   |   |   |   pressure_3pm > 1007.5: Yes (2.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = NNW: Yes (1.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = NW
## |   |   |   |   |   |   |   |   |   pressure_3pm <= 1003.1: Yes (6.0/1.0)
## |   |   |   |   |   |   |   |   |   pressure_3pm > 1003.1: No (5.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = NNW: Yes (1.0)
## |   |   |   |   |   |   humidity_3pm > 51
## |   |   |   |   |   |   |   humidity_3pm <= 59: No (525.0/165.0)
## |   |   |   |   |   |   |   humidity_3pm > 59
## |   |   |   |   |   |   |   |   humidity_9am <= 85
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = N
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = N
## |   |   |   |   |   |   |   |   |   |   |   rainfall <= 5.3: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   |   rainfall > 5.3: No (3.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = NNE: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = NE: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = ENE: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = E: No (1.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = ESE: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = SE: Yes (1.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = SSE: Yes (1.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = S: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = SSW: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = SW: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = WSW: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = W: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = WNW: No (2.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = NW: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = NNW: No (2.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = NNE
## |   |   |   |   |   |   |   |   |   |   wind_gust_speed <= 33: No (4.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_speed > 33
## |   |   |   |   |   |   |   |   |   |   |   evaporation <= 6.1: Yes (5.0)
```

```
## |   |   |   |   |   |   |   |   |   |   |   evaporation > 6.1: No (2.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = NE
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = N: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = NNE: Yes (1.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = NE: No (3.0/1.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = ENE
## |   |   |   |   |   |   |   |   |   |   |   |   humidity_9am <= 74: Yes (3.0)
## |   |   |   |   |   |   |   |   |   |   |   |   humidity_9am > 74: No (4.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = E: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = ESE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = SE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = SSE: Yes (1.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = S: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = SSW: Yes (1.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = SW: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = WSW: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = W: No (1.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = WNW: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = NW: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = NNW: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = ENE
## |   |   |   |   |   |   |   |   |   |   |   wind_speed_9am <= 6: Yes (6.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_speed_9am > 6
## |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_speed <= 43
## |   |   |   |   |   |   |   |   |   |   |   |   |   pressure_3pm <= 1005.3: Yes (
## |   |   |   |   |   |   |   |   |   |   |   |   |   pressure_3pm > 1005.3: No (24
## |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_speed > 43: Yes (3.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = E: No (19.0/7.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = ESE
## |   |   |   |   |   |   |   |   |   |   |   rainfall <= 17: No (14.0/2.0)
## |   |   |   |   |   |   |   |   |   |   |   rainfall > 17: Yes (5.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = SE
## |   |   |   |   |   |   |   |   |   |   |   evaporation <= 6: No (43.0/10.0)
## |   |   |   |   |   |   |   |   |   |   |   evaporation > 6: Yes (11.0/2.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = SSE
## |   |   |   |   |   |   |   |   |   |   |   cloud_3pm <= 6: No (8.0)
```

```
## |   |   |   |   |   |   |   |   |   |   cloud_3pm > 6: Yes (9.0/4.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = S
## |   |   |   |   |   |   |   |   |   |   wind_gust_speed <= 37: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_speed > 37
## |   |   |   |   |   |   |   |   |   |   |   wind_speed_9am <= 11
## |   |   |   |   |   |   |   |   |   |   |   |   pressure_3pm <= 1010.6: Yes
## |   |   |   |   |   |   |   |   |   |   |   |   pressure_3pm > 1010.6: No (2
## |   |   |   |   |   |   |   |   |   |   |   wind_speed_9am > 11: No (11.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = SSW: No (10.0/3.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = SW
## |   |   |   |   |   |   |   |   |   |   rainfall <= 4.3: No (10.0/1.0)
## |   |   |   |   |   |   |   |   |   |   rainfall > 4.3
## |   |   |   |   |   |   |   |   |   |   |   cloud_9am <= 6: Yes (7.0)
## |   |   |   |   |   |   |   |   |   |   |   cloud_9am > 6
## |   |   |   |   |   |   |   |   |   |   |   |   temp_3pm <= 18.5: Yes (3.0)
## |   |   |   |   |   |   |   |   |   |   |   |   temp_3pm > 18.5: No (2.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = WSW
## |   |   |   |   |   |   |   |   |   |   humidity_3pm <= 61: No (6.0)
## |   |   |   |   |   |   |   |   |   |   humidity_3pm > 61
## |   |   |   |   |   |   |   |   |   |   |   cloud_3pm <= 4: No (3.0)
## |   |   |   |   |   |   |   |   |   |   |   cloud_3pm > 4: Yes (14.0/3.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = W
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = N: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = NNE: No (1.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = NE: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = ENE: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = E: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = ESE: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = SE: Yes (1.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = SSE: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = S: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = SSW: Yes (1.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = SW: No (1.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = WSW: No (3.0/1.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = W: Yes (6.0/1.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = WNW
```

```
## |   |   |   |   |   |   |   |   |   |   |   cloud_9am <= 5: No (2.0)
## |   |   |   |   |   |   |   |   |   |   |   cloud_9am > 5
## |   |   |   |   |   |   |   |   |   |   |   |   sunshine <= 4.6: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   |   |   sunshine > 4.6: No (2.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = NW: Yes (1.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = NNW: No (4.0/1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = WNW: No (21.0/2.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = NW: No (30.0/10.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = NNW
## |   |   |   |   |   |   |   |   |   |   min_temp <= 23
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_speed <= 37: No (3.0/1
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_speed > 37: Yes (5.0)
## |   |   |   |   |   |   |   |   |   |   min_temp > 23: No (4.0)
## |   |   |   |   |   |   |   |   |   humidity_9am > 85
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = N
## |   |   |   |   |   |   |   |   |   |   wind_speed_3pm <= 24: Yes (12.0/2.0)
## |   |   |   |   |   |   |   |   |   |   wind_speed_3pm > 24: No (2.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = NNE: Yes (3.0/1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = NE: Yes (5.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = ENE
## |   |   |   |   |   |   |   |   |   |   wind_gust_speed <= 41: No (4.0/1.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_speed > 41: Yes (2.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = E: Yes (7.0/1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = ESE
## |   |   |   |   |   |   |   |   |   |   temp_9am <= 19.4: No (2.0)
## |   |   |   |   |   |   |   |   |   |   temp_9am > 19.4: Yes (4.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = SE: No (16.0/6.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = SSE
## |   |   |   |   |   |   |   |   |   |   humidity_3pm <= 62: No (6.0)
## |   |   |   |   |   |   |   |   |   |   humidity_3pm > 62: Yes (4.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = S
## |   |   |   |   |   |   |   |   |   |   temp_9am <= 20.1: No (9.0/1.0)
## |   |   |   |   |   |   |   |   |   |   temp_9am > 20.1: Yes (5.0/1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = SSW
## |   |   |   |   |   |   |   |   |   |   cloud_3pm <= 6: Yes (4.0/1.0)
## |   |   |   |   |   |   |   |   |   |   cloud_3pm > 6: No (4.0)
```

```
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = SW
## |   |   |   |   |   |   |   |   |   |   pressure_9am <= 1008.8: Yes (7.0)
## |   |   |   |   |   |   |   |   |   |   pressure_9am > 1008.8
## |   |   |   |   |   |   |   |   |   |   |   evaporation <= 4.6: No (7.0)
## |   |   |   |   |   |   |   |   |   |   |   evaporation > 4.6: Yes (5.0/1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = WSW
## |   |   |   |   |   |   |   |   |   |   max_temp <= 17.3: Yes (9.0)
## |   |   |   |   |   |   |   |   |   |   max_temp > 17.3
## |   |   |   |   |   |   |   |   |   |   |   pressure_3pm <= 1004.8: Yes (5.0
## |   |   |   |   |   |   |   |   |   |   |   pressure_3pm > 1004.8
## |   |   |   |   |   |   |   |   |   |   |   |   cloud_9am <= 5: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   |   |   cloud_9am > 5: No (10.0/1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = W
## |   |   |   |   |   |   |   |   |   |   cloud_3pm <= 7: Yes (20.0/3.0)
## |   |   |   |   |   |   |   |   |   |   cloud_3pm > 7: No (6.0/1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = WNW: No (19.0/7.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = NW: Yes (17.0/7.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = NNW: Yes (18.0/4.0)
## |   |   |   |   |   |   pressure_3pm > 1014.3
## |   |   |   |   |   |   |   rainfall <= 6.5: No (1736.0/387.0)
## |   |   |   |   |   |   |   rainfall > 6.5
## |   |   |   |   |   |   |   |   cloud_9am <= 7
## |   |   |   |   |   |   |   |   |   wind_gust_dir = N
## |   |   |   |   |   |   |   |   |   |   cloud_3pm <= 5: Yes (5.0/1.0)
## |   |   |   |   |   |   |   |   |   |   cloud_3pm > 5: No (4.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = NNE: No (4.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = NE
## |   |   |   |   |   |   |   |   |   |   cloud_3pm <= 5: No (7.0/1.0)
## |   |   |   |   |   |   |   |   |   |   cloud_3pm > 5: Yes (2.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = ENE
## |   |   |   |   |   |   |   |   |   |   humidity_3pm <= 52: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   humidity_3pm > 52: No (11.0/1.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = E
## |   |   |   |   |   |   |   |   |   |   cloud_9am <= 6: No (8.0/2.0)
## |   |   |   |   |   |   |   |   |   |   cloud_9am > 6: Yes (4.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = ESE: No (21.0/9.0)
```

```
## |   |   |   |   |   |   |   |   wind_gust_dir = SE: No (40.0/14.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SSE: No (38.0/13.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = S
## |   |   |   |   |   |   |   |   |   wind_gust_speed <= 39: Yes (20.0/8.0)
## |   |   |   |   |   |   |   |   |   wind_gust_speed > 39
## |   |   |   |   |   |   |   |   |   |   humidity_3pm <= 63: No (35.0/4.0)
## |   |   |   |   |   |   |   |   |   |   humidity_3pm > 63: Yes (5.0/1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SSW: No (77.0/16.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SW
## |   |   |   |   |   |   |   |   |   humidity_3pm <= 60
## |   |   |   |   |   |   |   |   |   |   wind_gust_speed <= 43: No (40.0/3.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_speed > 43
## |   |   |   |   |   |   |   |   |   |   |   humidity_3pm <= 55: Yes (5.0)
## |   |   |   |   |   |   |   |   |   |   |   humidity_3pm > 55: No (3.0)
## |   |   |   |   |   |   |   |   |   humidity_3pm > 60
## |   |   |   |   |   |   |   |   |   |   wind_speed_9am <= 13: Yes (8.0)
## |   |   |   |   |   |   |   |   |   |   wind_speed_9am > 13: No (12.0/4.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = WSW: No (47.0/13.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = W
## |   |   |   |   |   |   |   |   |   cloud_3pm <= 4
## |   |   |   |   |   |   |   |   |   |   sunshine <= 6.5
## |   |   |   |   |   |   |   |   |   |   |   wind_speed_9am <= 20: No (10.0/2
## |   |   |   |   |   |   |   |   |   |   |   wind_speed_9am > 20: Yes (3.0)
## |   |   |   |   |   |   |   |   |   |   sunshine > 6.5: No (32.0)
## |   |   |   |   |   |   |   |   |   cloud_3pm > 4
## |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = N: No (2.0/1.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = NNE: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = NE: No (6.0/1.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = ENE: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = E: No (1.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = ESE: Yes (1.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = SE
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = N
## |   |   |   |   |   |   |   |   |   |   |   |   cloud_9am <= 2
## |   |   |   |   |   |   |   |   |   |   |   |   |   cloud_9am <= 1: No (4.0/1
## |   |   |   |   |   |   |   |   |   |   |   |   |   cloud_9am > 1: Yes (4.0)
```

```
## |   |   |   |   |   |   |   |   |   |   |   |   |   cloud_9am > 2
## |   |   |   |   |   |   |   |   |   |   |   |   |   |   evaporation <= 3
## |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   cloud_9am <= 6: No (3
## |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   cloud_9am > 6: Yes (3
## |   |   |   |   |   |   |   |   |   |   |   |   |   |   evaporation > 3: No (71.0
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = NNE: Yes (1.0)
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = NE
## |   |   |   |   |   |   |   |   |   |   |   |   |   temp_9am <= 21.8: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   |   |   |   temp_9am > 21.8: No (2.0)
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = ENE: No (2.0)
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = E: No (0.0)
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = ESE: No (0.0)
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = SE
## |   |   |   |   |   |   |   |   |   |   |   |   |   temp_3pm <= 22.3
## |   |   |   |   |   |   |   |   |   |   |   |   |   |   max_temp <= 22.6: Yes (8
## |   |   |   |   |   |   |   |   |   |   |   |   |   |   max_temp > 22.6: No (6.0
## |   |   |   |   |   |   |   |   |   |   |   |   |   temp_3pm > 22.3: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = SSE: No (3.0/1.0)
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = S
## |   |   |   |   |   |   |   |   |   |   |   |   |   pressure_9am <= 1016.9: No (2
## |   |   |   |   |   |   |   |   |   |   |   |   |   pressure_9am > 1016.9: Yes (3
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = SSW: Yes (3.0/1.0
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = SW
## |   |   |   |   |   |   |   |   |   |   |   |   |   humidity_3pm <= 60: Yes (25.0
## |   |   |   |   |   |   |   |   |   |   |   |   |   humidity_3pm > 60: No (3.0)
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = WSW: Yes (5.0)
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = W
## |   |   |   |   |   |   |   |   |   |   |   |   |   cloud_9am <= 6: No (5.0)
## |   |   |   |   |   |   |   |   |   |   |   |   |   cloud_9am > 6: Yes (4.0/1.0)
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = WNW: No (2.0/1.0)
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = NW
## |   |   |   |   |   |   |   |   |   |   |   |   |   temp_3pm <= 22.4
## |   |   |   |   |   |   |   |   |   |   |   |   |   |   evaporation <= 5: No (15
## |   |   |   |   |   |   |   |   |   |   |   |   |   |   evaporation > 5: Yes (2.0
## |   |   |   |   |   |   |   |   |   |   |   |   |   temp_3pm > 22.4: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = NNW
```

```
## |   |   |   |   |   |   |   |   |   |   |   |   |   rainfall <= 13.2: Yes (4.0/1
## |   |   |   |   |   |   |   |   |   |   |   |   |   rainfall > 13.2: No (4.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = SSE: Yes (3.0/1.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = S: No (14.0/5.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = SSW: Yes (8.0/3.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = SW: Yes (13.0/4.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = WSW: No (18.0/2.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = W
## |   |   |   |   |   |   |   |   |   |   |   |   rainfall <= 12.2: No (10.0)
## |   |   |   |   |   |   |   |   |   |   |   |   rainfall > 12.2
## |   |   |   |   |   |   |   |   |   |   |   |   |   humidity_3pm <= 59: Yes (4.0
## |   |   |   |   |   |   |   |   |   |   |   |   |   humidity_3pm > 59: No (2.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = WNW
## |   |   |   |   |   |   |   |   |   |   |   |   cloud_9am <= 5: No (4.0)
## |   |   |   |   |   |   |   |   |   |   |   |   cloud_9am > 5: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = NW: No (4.0/1.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = NNW: No (1.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = WNW: No (24.0/10.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = NW
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = N
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_speed <= 41: No (2.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_speed > 41: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = NNE: No (2.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = NE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = ENE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = E: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = ESE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = SE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = SSE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = S: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = SSW: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = SW: No (2.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = WSW: No (3.0/1.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = W: No (2.0/1.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = WNW: Yes (3.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = NW: Yes (4.0)
```

```
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = NNW: No (1.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = NNW
## |   |   |   |   |   |   |   |   |   |   wind_speed_3pm <= 22: Yes (5.0/1.0)
## |   |   |   |   |   |   |   |   |   |   wind_speed_3pm > 22: No (4.0)
## |   |   |   |   |   |   |   |   cloud_9am > 7
## |   |   |   |   |   |   |   |   |   wind_gust_dir = N: No (2.0/1.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = NNE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = NE: Yes (4.0/1.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = ENE
## |   |   |   |   |   |   |   |   |   |   rainfall <= 17.2: No (2.0)
## |   |   |   |   |   |   |   |   |   |   rainfall > 17.2: Yes (6.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = E: Yes (3.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = ESE
## |   |   |   |   |   |   |   |   |   |   rainfall <= 13: No (2.0)
## |   |   |   |   |   |   |   |   |   |   rainfall > 13: Yes (4.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = SE
## |   |   |   |   |   |   |   |   |   |   evaporation <= 7.6: Yes (4.0)
## |   |   |   |   |   |   |   |   |   |   evaporation > 7.6: No (2.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = SSE
## |   |   |   |   |   |   |   |   |   |   humidity_9am <= 83: No (5.0)
## |   |   |   |   |   |   |   |   |   |   humidity_9am > 83: Yes (9.0/1.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = S: No (5.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = SSW: No (12.0/3.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = SW
## |   |   |   |   |   |   |   |   |   |   cloud_3pm <= 4: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   cloud_3pm > 4: No (8.0/1.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = WSW
## |   |   |   |   |   |   |   |   |   |   temp_9am <= 15.1: No (13.0/1.0)
## |   |   |   |   |   |   |   |   |   |   temp_9am > 15.1: Yes (3.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = W
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = N: Yes (39.0/9.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = NNE: No (1.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = NE
## |   |   |   |   |   |   |   |   |   |   |   min_temp <= 19.8: Yes (4.0)
## |   |   |   |   |   |   |   |   |   |   |   min_temp > 19.8: No (2.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = ENE: Yes (0.0)
```

```
## |   |   |   |   |   |   |   |   |   wind_dir_9am = E: No (2.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = ESE: No (3.0/1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = SE
## |   |   |   |   |   |   |   |   |   |   temp_9am <= 18.6: Yes (10.0)
## |   |   |   |   |   |   |   |   |   |   temp_9am > 18.6: No (11.0/3.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = SSE: No (2.0/1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = S: No (6.0/2.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = SSW: Yes (1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = SW
## |   |   |   |   |   |   |   |   |   |   temp_3pm <= 20.7: No (5.0/1.0)
## |   |   |   |   |   |   |   |   |   |   temp_3pm > 20.7: Yes (5.0/1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = WSW: Yes (1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = W: Yes (5.0/2.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = WNW: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = NW: No (3.0/1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = NNW: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = WNW: No (4.0/1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NW: Yes (3.0/1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NNW: No (1.0)
## |   |   |   wind_gust_speed > 46
## |   |   |   |   humidity_3pm <= 48
## |   |   |   |   |   sunshine <= 6.7
## |   |   |   |   |   |   wind_dir_9am = N: No (41.0/19.0)
## |   |   |   |   |   |   wind_dir_9am = NNE: No (10.0/5.0)
## |   |   |   |   |   |   wind_dir_9am = NE
## |   |   |   |   |   |   |   wind_gust_speed <= 52: Yes (4.0)
## |   |   |   |   |   |   |   wind_gust_speed > 52: No (2.0)
## |   |   |   |   |   |   wind_dir_9am = ENE
## |   |   |   |   |   |   |   rainfall <= 1.5: No (3.0/1.0)
## |   |   |   |   |   |   |   rainfall > 1.5: Yes (2.0)
## |   |   |   |   |   |   wind_dir_9am = E
## |   |   |   |   |   |   |   wind_speed_9am <= 11: Yes (3.0)
## |   |   |   |   |   |   |   wind_speed_9am > 11: No (3.0/1.0)
## |   |   |   |   |   |   wind_dir_9am = ESE: No (2.0)
## |   |   |   |   |   |   wind_dir_9am = SE: No (4.0/1.0)
## |   |   |   |   |   |   wind_dir_9am = SSE: Yes (3.0/1.0)
```

```
## |   |   |   |   |   |   wind_dir_9am = S
## |   |   |   |   |   |   |   wind_gust_dir = N: No (0.0)
## |   |   |   |   |   |   |   wind_gust_dir = NNE: No (0.0)
## |   |   |   |   |   |   |   wind_gust_dir = NE: No (0.0)
## |   |   |   |   |   |   |   wind_gust_dir = ENE: No (0.0)
## |   |   |   |   |   |   |   wind_gust_dir = E: Yes (1.0)
## |   |   |   |   |   |   |   wind_gust_dir = ESE: No (0.0)
## |   |   |   |   |   |   |   wind_gust_dir = SE: No (0.0)
## |   |   |   |   |   |   |   wind_gust_dir = SSE: Yes (1.0)
## |   |   |   |   |   |   |   wind_gust_dir = S: No (8.0)
## |   |   |   |   |   |   |   wind_gust_dir = SSW: No (2.0/1.0)
## |   |   |   |   |   |   |   wind_gust_dir = SW: Yes (1.0)
## |   |   |   |   |   |   |   wind_gust_dir = WSW: Yes (1.0)
## |   |   |   |   |   |   |   wind_gust_dir = W: No (0.0)
## |   |   |   |   |   |   |   wind_gust_dir = WNW: No (0.0)
## |   |   |   |   |   |   |   wind_gust_dir = NW: No (0.0)
## |   |   |   |   |   |   |   wind_gust_dir = NNW: No (0.0)
## |   |   |   |   |   |   wind_dir_9am = SSW: No (6.0/1.0)
## |   |   |   |   |   |   wind_dir_9am = SW
## |   |   |   |   |   |   |   wind_speed_9am <= 20: Yes (5.0/1.0)
## |   |   |   |   |   |   |   wind_speed_9am > 20: No (11.0)
## |   |   |   |   |   |   wind_dir_9am = WSW
## |   |   |   |   |   |   |   max_temp <= 16: Yes (9.0/3.0)
## |   |   |   |   |   |   |   max_temp > 16: No (13.0/2.0)
## |   |   |   |   |   |   wind_dir_9am = W
## |   |   |   |   |   |   |   cloud_9am <= 4: No (5.0)
## |   |   |   |   |   |   |   cloud_9am > 4: Yes (20.0/6.0)
## |   |   |   |   |   |   wind_dir_9am = WNW: No (15.0/3.0)
## |   |   |   |   |   |   wind_dir_9am = NW
## |   |   |   |   |   |   |   temp_9am <= 18.7: Yes (14.0/1.0)
## |   |   |   |   |   |   |   temp_9am > 18.7: No (2.0)
## |   |   |   |   |   |   wind_dir_9am = NNW: No (20.0/7.0)
## |   |   |   |   |   sunshine > 6.7
## |   |   |   |   |   |   pressure_3pm <= 1002.8
## |   |   |   |   |   |   |   wind_dir_3pm = N: Yes (6.0/1.0)
## |   |   |   |   |   |   |   wind_dir_3pm = NNE: Yes (5.0/1.0)
```

```
## |   |   |   |   |   |   |   wind_dir_3pm = NE: No (0.0)
## |   |   |   |   |   |   |   wind_dir_3pm = ENE: No (0.0)
## |   |   |   |   |   |   |   wind_dir_3pm = E: No (0.0)
## |   |   |   |   |   |   |   wind_dir_3pm = ESE: No (0.0)
## |   |   |   |   |   |   |   wind_dir_3pm = SE: Yes (1.0)
## |   |   |   |   |   |   |   wind_dir_3pm = SSE: No (2.0/1.0)
## |   |   |   |   |   |   |   wind_dir_3pm = S: No (0.0)
## |   |   |   |   |   |   |   wind_dir_3pm = SSW: No (2.0)
## |   |   |   |   |   |   |   wind_dir_3pm = SW: No (2.0)
## |   |   |   |   |   |   |   wind_dir_3pm = WSW
## |   |   |   |   |   |   |   |   rainfall <= 9: No (13.0/1.0)
## |   |   |   |   |   |   |   |   rainfall > 9: Yes (3.0)
## |   |   |   |   |   |   |   wind_dir_3pm = W: No (25.0/8.0)
## |   |   |   |   |   |   |   wind_dir_3pm = WNW
## |   |   |   |   |   |   |   |   humidity_3pm <= 32: No (12.0)
## |   |   |   |   |   |   |   |   humidity_3pm > 32
## |   |   |   |   |   |   |   |   |   wind_dir_9am = N: Yes (2.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = NNE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = NE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = ENE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = E: No (2.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = ESE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = SE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = SSE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = S: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = SSW: No (1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = SW: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = WSW: No (1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = W: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = WNW
## |   |   |   |   |   |   |   |   |   |   pressure_9am <= 1002.6: Yes (3.0)
## |   |   |   |   |   |   |   |   |   |   pressure_9am > 1002.6: No (3.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = NW
## |   |   |   |   |   |   |   |   |   |   cloud_3pm <= 5
## |   |   |   |   |   |   |   |   |   |   |   rainfall <= 13: No (3.0)
## |   |   |   |   |   |   |   |   |   |   |   rainfall > 13: Yes (2.0)
```

```
## |   |   |   |   |   |   |   |   |   cloud_3pm > 5: Yes (2.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = NNW
## |   |   |   |   |   |   |   |   |   cloud_9am <= 6: No (4.0/1.0)
## |   |   |   |   |   |   |   |   |   cloud_9am > 6: Yes (5.0)
## |   |   |   |   |   |   wind_dir_3pm = NW
## |   |   |   |   |   |   |   wind_gust_speed <= 57: Yes (3.0)
## |   |   |   |   |   |   |   wind_gust_speed > 57: No (27.0/6.0)
## |   |   |   |   |   |   wind_dir_3pm = NNW
## |   |   |   |   |   |   |   wind_speed_9am <= 19: No (6.0/2.0)
## |   |   |   |   |   |   |   wind_speed_9am > 19: Yes (6.0/1.0)
## |   |   |   |   |   pressure_3pm > 1002.8: No (855.0/162.0)
## |   |   |   humidity_3pm > 48
## |   |   |   |   pressure_3pm <= 1012.5
## |   |   |   |   |   wind_gust_speed <= 59
## |   |   |   |   |   |   wind_speed_9am <= 26
## |   |   |   |   |   |   |   wind_dir_3pm = N: Yes (30.0/10.0)
## |   |   |   |   |   |   |   wind_dir_3pm = NNE: Yes (19.0/6.0)
## |   |   |   |   |   |   |   wind_dir_3pm = NE: Yes (20.0/5.0)
## |   |   |   |   |   |   |   wind_dir_3pm = ENE
## |   |   |   |   |   |   |   |   evaporation <= 7.1: Yes (12.0/2.0)
## |   |   |   |   |   |   |   |   evaporation > 7.1: No (5.0)
## |   |   |   |   |   |   |   wind_dir_3pm = E: No (17.0/8.0)
## |   |   |   |   |   |   |   wind_dir_3pm = ESE: No (21.0/6.0)
## |   |   |   |   |   |   |   wind_dir_3pm = SE
## |   |   |   |   |   |   |   |   wind_gust_dir = N: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NNE: No (1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NE: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = ENE: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = E: No (2.0/1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = ESE: Yes (3.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SE
## |   |   |   |   |   |   |   |   |   humidity_3pm <= 60: Yes (10.0)
## |   |   |   |   |   |   |   |   |   humidity_3pm > 60
## |   |   |   |   |   |   |   |   |   |   sunshine <= 7.9: No (3.0)
## |   |   |   |   |   |   |   |   |   |   sunshine > 7.9: Yes (3.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SSE: No (4.0/1.0)
```

```
## |   |   |   |   |   |   |   |   wind_gust_dir = S: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SSW: Yes (1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SW: No (1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = WSW: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = W: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = WNW: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NW: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NNW: No (1.0)
## |   |   |   |   |   |   |   wind_dir_3pm = SSE
## |   |   |   |   |   |   |   |   sunshine <= 4.8: Yes (6.0)
## |   |   |   |   |   |   |   |   sunshine > 4.8: No (17.0/6.0)
## |   |   |   |   |   |   |   wind_dir_3pm = S
## |   |   |   |   |   |   |   |   rainfall <= 4: Yes (5.0/1.0)
## |   |   |   |   |   |   |   |   rainfall > 4: No (13.0/4.0)
## |   |   |   |   |   |   |   wind_dir_3pm = SSW
## |   |   |   |   |   |   |   |   humidity_9am <= 79: No (13.0/3.0)
## |   |   |   |   |   |   |   |   humidity_9am > 79: Yes (8.0/3.0)
## |   |   |   |   |   |   |   wind_dir_3pm = SW
## |   |   |   |   |   |   |   |   cloud_9am <= 6: Yes (20.0/1.0)
## |   |   |   |   |   |   |   |   cloud_9am > 6
## |   |   |   |   |   |   |   |   |   wind_gust_speed <= 52: No (17.0/6.0)
## |   |   |   |   |   |   |   |   |   wind_gust_speed > 52: Yes (4.0/1.0)
## |   |   |   |   |   |   |   wind_dir_3pm = WSW
## |   |   |   |   |   |   |   |   wind_gust_dir = N: Yes (1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NNE: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NE: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = ENE: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = E: Yes (1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = ESE: No (1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SE: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SSE: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = S: No (1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SSW: Yes (2.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SW
## |   |   |   |   |   |   |   |   |   pressure_3pm <= 1007.2: No (3.0)
## |   |   |   |   |   |   |   |   |   pressure_3pm > 1007.2: Yes (5.0)
```

```
## |   |   |   |   |   |   |   |   wind_gust_dir = WSW
## |   |   |   |   |   |   |   |   |   cloud_9am <= 7
## |   |   |   |   |   |   |   |   |   |   evaporation <= 5.6: Yes (20.0/1.0)
## |   |   |   |   |   |   |   |   |   |   evaporation > 5.6: No (2.0)
## |   |   |   |   |   |   |   |   |   cloud_9am > 7: No (7.0/2.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = W
## |   |   |   |   |   |   |   |   |   cloud_3pm <= 4: No (2.0)
## |   |   |   |   |   |   |   |   |   cloud_3pm > 4
## |   |   |   |   |   |   |   |   |   |   evaporation <= 5.8: Yes (20.0/5.0)
## |   |   |   |   |   |   |   |   |   |   evaporation > 5.8: No (4.0/1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = WNW
## |   |   |   |   |   |   |   |   |   wind_gust_speed <= 54: No (3.0)
## |   |   |   |   |   |   |   |   |   wind_gust_speed > 54: Yes (2.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NW: Yes (5.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NNW: Yes (1.0)
## |   |   |   |   |   |   |   wind_dir_3pm = W
## |   |   |   |   |   |   |   |   wind_gust_dir = N: No (3.0/1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NNE: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NE: Yes (1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = ENE: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = E: Yes (1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = ESE: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SE: No (3.0/1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SSE: No (1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = S: Yes (1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SSW: Yes (3.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SW
## |   |   |   |   |   |   |   |   |   cloud_3pm <= 6: Yes (5.0)
## |   |   |   |   |   |   |   |   |   cloud_3pm > 6: No (2.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = WSW
## |   |   |   |   |   |   |   |   |   min_temp <= 8.4
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = N: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = NNE: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = NE: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = ENE: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = E: No (0.0)
```

```
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = ESE: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = SE: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = SSE: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = S: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = SSW: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = SW: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = WSW: No (2.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = W: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = WNW: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = NW: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = NNW: No (1.0)
## |   |   |   |   |   |   |   |   |   min_temp > 8.4: Yes (12.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = W
## |   |   |   |   |   |   |   |   |   wind_dir_9am = N: Yes (1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = NNE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = NE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = ENE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = E: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = ESE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = SE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = SSE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = S: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = SSW: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = SW: No (3.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = WSW
## |   |   |   |   |   |   |   |   |   |   cloud_9am <= 6: Yes (4.0)
## |   |   |   |   |   |   |   |   |   |   cloud_9am > 6: No (4.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = W
## |   |   |   |   |   |   |   |   |   |   wind_speed_3pm <= 30: Yes (8.0/1.0)
## |   |   |   |   |   |   |   |   |   |   wind_speed_3pm > 30: No (3.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = WNW
## |   |   |   |   |   |   |   |   |   |   wind_gust_speed <= 54: No (2.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_speed > 54: Yes (3.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = NW: Yes (4.0/1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = NNW: No (4.0/1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = WNW
```

```
## |   |   |   |   |   |   |   |   |   wind_speed_3pm <= 24: Yes (5.0)
## |   |   |   |   |   |   |   |   |   wind_speed_3pm > 24: No (5.0/1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NW
## |   |   |   |   |   |   |   |   |   wind_speed_9am <= 11: No (2.0)
## |   |   |   |   |   |   |   |   |   wind_speed_9am > 11: Yes (8.0/1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NNW: Yes (3.0)
## |   |   |   |   |   |   |   wind_dir_3pm = WNW
## |   |   |   |   |   |   |   |   wind_gust_dir = N: No (2.0/1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NNE: Yes (1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NE: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = ENE: Yes (2.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = E: Yes (2.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = ESE: No (1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SE: Yes (1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SSE: Yes (2.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = S: Yes (1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SSW: No (1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SW: Yes (5.0/1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = WSW: No (4.0/1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = W: Yes (21.0/6.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = WNW: No (22.0/7.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NW
## |   |   |   |   |   |   |   |   |   wind_gust_speed <= 56: No (15.0/5.0)
## |   |   |   |   |   |   |   |   |   wind_gust_speed > 56: Yes (5.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NNW
## |   |   |   |   |   |   |   |   |   temp_3pm <= 21.1: No (2.0)
## |   |   |   |   |   |   |   |   |   temp_3pm > 21.1: Yes (2.0)
## |   |   |   |   |   |   |   wind_dir_3pm = NW
## |   |   |   |   |   |   |   |   wind_gust_dir = N: No (4.0/1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NNE: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NE: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = ENE: No (1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = E: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = ESE: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SE: No (1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SSE: Yes (2.0)
```

```
## |   |   |   |   |   |   |   |   wind_gust_dir = S: Yes (1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SSW: No (1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SW: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = WSW: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = W: Yes (3.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = WNW
## |   |   |   |   |   |   |   |   |   wind_gust_speed <= 48: No (3.0)
## |   |   |   |   |   |   |   |   |   wind_gust_speed > 48
## |   |   |   |   |   |   |   |   |   |   min_temp <= 7.3: No (4.0)
## |   |   |   |   |   |   |   |   |   |   min_temp > 7.3: Yes (14.0/2.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NW
## |   |   |   |   |   |   |   |   |   pressure_9am <= 999.5: No (4.0)
## |   |   |   |   |   |   |   |   |   pressure_9am > 999.5: Yes (30.0/7.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NNW
## |   |   |   |   |   |   |   |   |   evaporation <= 1.6: No (2.0)
## |   |   |   |   |   |   |   |   |   evaporation > 1.6: Yes (9.0)
## |   |   |   |   |   |   |   wind_dir_3pm = NNW
## |   |   |   |   |   |   |   |   wind_gust_dir = N
## |   |   |   |   |   |   |   |   |   humidity_3pm <= 61: Yes (2.0)
## |   |   |   |   |   |   |   |   |   humidity_3pm > 61: No (2.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NNE: No (1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NE: No (1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = ENE: Yes (1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = E: Yes (1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = ESE: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SE: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SSE: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = S: No (1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SSW: Yes (1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SW: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = WSW: No (2.0/1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = W
## |   |   |   |   |   |   |   |   |   temp_9am <= 21.6: Yes (2.0)
## |   |   |   |   |   |   |   |   |   temp_9am > 21.6: No (2.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = WNW: Yes (6.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NW
```

```
## |   |   |   |   |   |   |   |   |   evaporation <= 4.5: No (4.0)
## |   |   |   |   |   |   |   |   |   evaporation > 4.5: Yes (3.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NNW
## |   |   |   |   |   |   |   |   |   sunshine <= 7.5
## |   |   |   |   |   |   |   |   |   |   temp_9am <= 7.3: No (2.0)
## |   |   |   |   |   |   |   |   |   |   temp_9am > 7.3: Yes (8.0/1.0)
## |   |   |   |   |   |   |   |   |   sunshine > 7.5: No (4.0)
## |   |   |   |   |   |   wind_speed_9am > 26: No (103.0/36.0)
## |   |   |   |   |   wind_gust_speed > 59
## |   |   |   |   |   |   wind_dir_3pm = N
## |   |   |   |   |   |   |   max_temp <= 15.9
## |   |   |   |   |   |   |   |   wind_speed_9am <= 26: Yes (3.0)
## |   |   |   |   |   |   |   |   wind_speed_9am > 26: No (3.0)
## |   |   |   |   |   |   |   max_temp > 15.9: Yes (18.0)
## |   |   |   |   |   |   wind_dir_3pm = NNE: Yes (10.0)
## |   |   |   |   |   |   wind_dir_3pm = NE
## |   |   |   |   |   |   |   rainfall <= 2.9: No (3.0)
## |   |   |   |   |   |   |   rainfall > 2.9: Yes (8.0)
## |   |   |   |   |   |   wind_dir_3pm = ENE: Yes (5.0/1.0)
## |   |   |   |   |   |   wind_dir_3pm = E
## |   |   |   |   |   |   |   wind_speed_3pm <= 11: No (3.0/1.0)
## |   |   |   |   |   |   |   wind_speed_3pm > 11: Yes (5.0)
## |   |   |   |   |   |   wind_dir_3pm = ESE: Yes (7.0/2.0)
## |   |   |   |   |   |   wind_dir_3pm = SE: Yes (11.0/3.0)
## |   |   |   |   |   |   wind_dir_3pm = SSE
## |   |   |   |   |   |   |   temp_3pm <= 27.2: No (7.0)
## |   |   |   |   |   |   |   temp_3pm > 27.2: Yes (4.0)
## |   |   |   |   |   |   wind_dir_3pm = S
## |   |   |   |   |   |   |   sunshine <= 7.6
## |   |   |   |   |   |   |   |   humidity_3pm <= 59: Yes (2.0)
## |   |   |   |   |   |   |   |   humidity_3pm > 59: No (3.0/1.0)
## |   |   |   |   |   |   |   sunshine > 7.6: No (7.0)
## |   |   |   |   |   |   wind_dir_3pm = SSW
## |   |   |   |   |   |   |   wind_speed_3pm <= 37: No (15.0/4.0)
## |   |   |   |   |   |   |   wind_speed_3pm > 37: Yes (4.0)
## |   |   |   |   |   |   wind_dir_3pm = SW
```

```
## |   |   |   |   |   |   |   |   wind_gust_dir = N: No (1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NNE: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NE: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = ENE: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = E: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = ESE: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SE: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SSE: Yes (1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = S: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SSW
## |   |   |   |   |   |   |   |   |   cloud_3pm <= 4: No (2.0)
## |   |   |   |   |   |   |   |   |   cloud_3pm > 4: Yes (4.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SW
## |   |   |   |   |   |   |   |   |   wind_speed_9am <= 28: Yes (8.0/1.0)
## |   |   |   |   |   |   |   |   |   wind_speed_9am > 28: No (4.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = WSW: Yes (9.0/3.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = W: Yes (4.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = WNW: No (3.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NW: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NNW: Yes (1.0)
## |   |   |   |   |   |   |   wind_dir_3pm = WSW: Yes (81.0/22.0)
## |   |   |   |   |   |   |   wind_dir_3pm = W
## |   |   |   |   |   |   |   |   humidity_9am <= 77
## |   |   |   |   |   |   |   |   |   wind_speed_3pm <= 37
## |   |   |   |   |   |   |   |   |   |   humidity_9am <= 70: Yes (29.0/5.0)
## |   |   |   |   |   |   |   |   |   |   humidity_9am > 70
## |   |   |   |   |   |   |   |   |   |   |   temp_3pm <= 9.3: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   |   temp_3pm > 9.3: No (14.0/5.0)
## |   |   |   |   |   |   |   |   |   wind_speed_3pm > 37
## |   |   |   |   |   |   |   |   |   |   cloud_3pm <= 7: No (20.0/6.0)
## |   |   |   |   |   |   |   |   |   |   cloud_3pm > 7: Yes (6.0/2.0)
## |   |   |   |   |   |   |   |   humidity_9am > 77: Yes (57.0/9.0)
## |   |   |   |   |   |   |   wind_dir_3pm = WNW: Yes (93.0/22.0)
## |   |   |   |   |   |   |   wind_dir_3pm = NW: Yes (72.0/14.0)
## |   |   |   |   |   |   |   wind_dir_3pm = NNW: Yes (35.0/4.0)
## |   |   |   |   |   pressure_3pm > 1012.5
```

```
## |   |   |   |   |   |   humidity_9am <= 68: No (457.0/120.0)
## |   |   |   |   |   |   humidity_9am > 68
## |   |   |   |   |   |   |   wind_dir_3pm = N: Yes (19.0/7.0)
## |   |   |   |   |   |   |   wind_dir_3pm = NNE: No (6.0/1.0)
## |   |   |   |   |   |   |   wind_dir_3pm = NE: Yes (10.0/4.0)
## |   |   |   |   |   |   |   wind_dir_3pm = ENE: Yes (15.0/3.0)
## |   |   |   |   |   |   |   wind_dir_3pm = E
## |   |   |   |   |   |   |   |   wind_gust_dir = N: No (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NNE: No (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NE: Yes (1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = ENE: Yes (2.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = E
## |   |   |   |   |   |   |   |   |   pressure_9am <= 1022.8: No (10.0)
## |   |   |   |   |   |   |   |   |   pressure_9am > 1022.8: Yes (2.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = ESE: No (4.0/1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SE: No (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SSE: No (3.0/1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = S: No (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SSW: No (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SW: No (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = WSW: No (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = W: No (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = WNW: No (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NW: No (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NNW: No (0.0)
## |   |   |   |   |   |   |   wind_dir_3pm = ESE
## |   |   |   |   |   |   |   |   wind_gust_dir = N: No (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NNE: No (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NE: No (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = ENE: No (1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = E: Yes (4.0/1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = ESE
## |   |   |   |   |   |   |   |   |   sunshine <= 2.4: Yes (2.0)
## |   |   |   |   |   |   |   |   |   sunshine > 2.4: No (8.0/1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SE
## |   |   |   |   |   |   |   |   |   pressure_3pm <= 1015.1: Yes (2.0)
```

```
## |   |   |   |   |   |   |   |   pressure_3pm > 1015.1: No (5.0)
## |   |   |   |   |   |   |   wind_gust_dir = SSE: Yes (2.0)
## |   |   |   |   |   |   |   wind_gust_dir = S: Yes (2.0)
## |   |   |   |   |   |   |   wind_gust_dir = SSW: Yes (1.0)
## |   |   |   |   |   |   |   wind_gust_dir = SW: No (0.0)
## |   |   |   |   |   |   |   wind_gust_dir = WSW: Yes (1.0)
## |   |   |   |   |   |   |   wind_gust_dir = W: No (0.0)
## |   |   |   |   |   |   |   wind_gust_dir = WNW: No (0.0)
## |   |   |   |   |   |   |   wind_gust_dir = NW: No (0.0)
## |   |   |   |   |   |   |   wind_gust_dir = NNW: No (0.0)
## |   |   |   |   |   |   wind_dir_3pm = SE
## |   |   |   |   |   |   |   wind_dir_9am = N: Yes (3.0)
## |   |   |   |   |   |   |   wind_dir_9am = NNE: Yes (0.0)
## |   |   |   |   |   |   |   wind_dir_9am = NE: Yes (0.0)
## |   |   |   |   |   |   |   wind_dir_9am = ENE: Yes (1.0)
## |   |   |   |   |   |   |   wind_dir_9am = E: No (5.0/1.0)
## |   |   |   |   |   |   |   wind_dir_9am = ESE: No (3.0/1.0)
## |   |   |   |   |   |   |   wind_dir_9am = SE
## |   |   |   |   |   |   |   |   max_temp <= 23.5
## |   |   |   |   |   |   |   |   |   pressure_9am <= 1020.8
## |   |   |   |   |   |   |   |   |   |   evaporation <= 5: No (9.0)
## |   |   |   |   |   |   |   |   |   |   evaporation > 5: Yes (3.0/1.0)
## |   |   |   |   |   |   |   |   |   pressure_9am > 1020.8: Yes (3.0)
## |   |   |   |   |   |   |   |   max_temp > 23.5: Yes (5.0)
## |   |   |   |   |   |   |   wind_dir_9am = SSE
## |   |   |   |   |   |   |   |   wind_speed_3pm <= 30: Yes (8.0)
## |   |   |   |   |   |   |   |   wind_speed_3pm > 30
## |   |   |   |   |   |   |   |   |   wind_gust_dir = N: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = NNE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = NE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = ENE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = E: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = ESE: Yes (1.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = SE
## |   |   |   |   |   |   |   |   |   |   cloud_9am <= 7
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_speed <= 56: Yes (5.0/1
```

```
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_speed > 56: No (2.0)
## |   |   |   |   |   |   |   |   |   |   |   cloud_9am > 7: No (4.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = SSE
## |   |   |   |   |   |   |   |   |   |   |   evaporation <= 5.5: No (3.0)
## |   |   |   |   |   |   |   |   |   |   |   evaporation > 5.5: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = S: No (2.0/1.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = SSW: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = SW: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = WSW: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = W: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = WNW: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = NW: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = NNW: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = S: Yes (21.0/4.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = SSW
## |   |   |   |   |   |   |   |   |   wind_gust_dir = N: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = NNE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = NE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = ENE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = E: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = ESE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = SE: Yes (3.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = SSE: No (1.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = S: Yes (1.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = SSW: No (2.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = SW: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = WSW: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = W: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = WNW: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = NW: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = NNW: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = SW: No (2.0/1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = WSW: No (2.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = W: Yes (3.0/1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = WNW: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = NW: No (2.0/1.0)
```

```
## |   |   |   |   |   |   |   wind_dir_9am = NNW: No (1.0)
## |   |   |   |   |   |   wind_dir_3pm = SSE
## |   |   |   |   |   |   |   wind_gust_dir = N: No (0.0)
## |   |   |   |   |   |   |   wind_gust_dir = NNE: No (0.0)
## |   |   |   |   |   |   |   wind_gust_dir = NE: No (0.0)
## |   |   |   |   |   |   |   wind_gust_dir = ENE: No (0.0)
## |   |   |   |   |   |   |   wind_gust_dir = E: No (0.0)
## |   |   |   |   |   |   |   wind_gust_dir = ESE: Yes (4.0)
## |   |   |   |   |   |   |   wind_gust_dir = SE
## |   |   |   |   |   |   |   |   pressure_3pm <= 1018.4: No (5.0/1.0)
## |   |   |   |   |   |   |   |   pressure_3pm > 1018.4: Yes (8.0)
## |   |   |   |   |   |   |   wind_gust_dir = SSE
## |   |   |   |   |   |   |   |   rainfall <= 21.6
## |   |   |   |   |   |   |   |   |   temp_3pm <= 24.4
## |   |   |   |   |   |   |   |   |   |   sunshine <= 7.4: Yes (4.0/1.0)
## |   |   |   |   |   |   |   |   |   |   sunshine > 7.4: No (27.0/3.0)
## |   |   |   |   |   |   |   |   |   temp_3pm > 24.4: Yes (7.0/1.0)
## |   |   |   |   |   |   |   |   rainfall > 21.6: Yes (6.0)
## |   |   |   |   |   |   |   wind_gust_dir = S
## |   |   |   |   |   |   |   |   pressure_3pm <= 1024.4: No (15.0/3.0)
## |   |   |   |   |   |   |   |   pressure_3pm > 1024.4: Yes (4.0/1.0)
## |   |   |   |   |   |   |   wind_gust_dir = SSW: No (9.0/3.0)
## |   |   |   |   |   |   |   wind_gust_dir = SW: No (1.0)
## |   |   |   |   |   |   |   wind_gust_dir = WSW: No (1.0)
## |   |   |   |   |   |   |   wind_gust_dir = W: Yes (1.0)
## |   |   |   |   |   |   |   wind_gust_dir = WNW: No (2.0/1.0)
## |   |   |   |   |   |   |   wind_gust_dir = NW: No (0.0)
## |   |   |   |   |   |   |   wind_gust_dir = NNW: No (0.0)
## |   |   |   |   |   |   wind_dir_3pm = S
## |   |   |   |   |   |   |   cloud_9am <= 7
## |   |   |   |   |   |   |   |   temp_9am <= 7.4: Yes (6.0)
## |   |   |   |   |   |   |   |   temp_9am > 7.4: No (72.0/20.0)
## |   |   |   |   |   |   |   cloud_9am > 7
## |   |   |   |   |   |   |   |   wind_dir_9am = N: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = NNE: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = NE: Yes (0.0)
```

```
## |   |   |   |   |   |   |   |   wind_dir_9am = ENE: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = E: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = ESE: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = SE: Yes (1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = SSE: Yes (1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = S
## |   |   |   |   |   |   |   |   |   cloud_3pm <= 6: Yes (4.0/1.0)
## |   |   |   |   |   |   |   |   |   cloud_3pm > 6: No (5.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = SSW: Yes (7.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = SW
## |   |   |   |   |   |   |   |   |   pressure_3pm <= 1020.1: Yes (2.0)
## |   |   |   |   |   |   |   |   |   pressure_3pm > 1020.1: No (2.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = WSW: Yes (4.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = W: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = WNW: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = NW: Yes (1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = NNW: Yes (0.0)
## |   |   |   |   |   |   |   wind_dir_3pm = SSW
## |   |   |   |   |   |   |   |   sunshine <= 6.6
## |   |   |   |   |   |   |   |   wind_dir_9am = N: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = NNE: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = NE: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = ENE: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = E: Yes (1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = ESE: No (1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = SE: Yes (1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = SSE: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = S
## |   |   |   |   |   |   |   |   |   evaporation <= 2.3: Yes (2.0)
## |   |   |   |   |   |   |   |   |   evaporation > 2.3: No (5.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = SSW
## |   |   |   |   |   |   |   |   |   max_temp <= 17.3
## |   |   |   |   |   |   |   |   |   |   humidity_3pm <= 53: No (2.0)
## |   |   |   |   |   |   |   |   |   |   humidity_3pm > 53: Yes (7.0)
## |   |   |   |   |   |   |   |   |   max_temp > 17.3: No (3.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = SW
```

```
## |   |   |   |   |   |   |   |   |   wind_speed_3pm <= 26: No (4.0)
## |   |   |   |   |   |   |   |   |   wind_speed_3pm > 26
## |   |   |   |   |   |   |   |   |   |   sunshine <= 3.5: Yes (6.0)
## |   |   |   |   |   |   |   |   |   |   sunshine > 3.5: No (4.0/1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = WSW: Yes (6.0/1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = W: Yes (9.0/1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = WNW: Yes (1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = NW: Yes (1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = NNW: No (2.0)
## |   |   |   |   |   |   |   sunshine > 6.6: No (86.0/24.0)
## |   |   |   |   |   |   wind_dir_3pm = SW
## |   |   |   |   |   |   |   cloud_9am <= 3: No (13.0)
## |   |   |   |   |   |   |   cloud_9am > 3
## |   |   |   |   |   |   |   |   rainfall <= 6.6: No (66.0/15.0)
## |   |   |   |   |   |   |   |   rainfall > 6.6
## |   |   |   |   |   |   |   |   |   wind_gust_dir = N: No (1.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = NNE: No (1.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = NE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = ENE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = E: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = ESE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = SE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = SSE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = S: No (1.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = SSW
## |   |   |   |   |   |   |   |   |   |   cloud_3pm <= 4: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   cloud_3pm > 4: No (8.0/2.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = SW
## |   |   |   |   |   |   |   |   |   |   min_temp <= 12: Yes (17.0/2.0)
## |   |   |   |   |   |   |   |   |   |   min_temp > 12: No (3.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = WSW
## |   |   |   |   |   |   |   |   |   |   wind_speed_9am <= 9: No (3.0/1.0)
## |   |   |   |   |   |   |   |   |   |   wind_speed_9am > 9: Yes (9.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = W
## |   |   |   |   |   |   |   |   |   |   evaporation <= 2.6: Yes (3.0)
## |   |   |   |   |   |   |   |   |   |   evaporation > 2.6: No (4.0)
```

```
## |   |   |   |   |   |   |   |   |   wind_gust_dir = WNW: No (1.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = NW: No (1.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = NNW: No (1.0)
## |   |   |   |   |   |   |   wind_dir_3pm = WSW
## |   |   |   |   |   |   |   |   wind_dir_9am = N: Yes (5.0/1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = NNE: Yes (1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = NE: Yes (2.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = ENE: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = E: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = ESE: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = SE: No (1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = SSE: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = S: Yes (1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = SSW: No (3.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = SW
## |   |   |   |   |   |   |   |   |   rainfall <= 6.1: No (9.0)
## |   |   |   |   |   |   |   |   |   rainfall > 6.1
## |   |   |   |   |   |   |   |   |   |   temp_9am <= 9: No (2.0)
## |   |   |   |   |   |   |   |   |   |   temp_9am > 9: Yes (7.0/1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = WSW
## |   |   |   |   |   |   |   |   |   evaporation <= 2.1: No (9.0)
## |   |   |   |   |   |   |   |   |   evaporation > 2.1
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = N: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = NNE: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = NE: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = ENE: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = E: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = ESE: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = SE: No (1.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = SSE: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = S: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = SSW: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = SW
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_speed <= 54: No (4.0/1.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_speed > 54: Yes (3.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = WSW
```

```
## |   |   |   |   |   |   |   |   |   |   |   humidity_3pm <= 54: No (5.0)
## |   |   |   |   |   |   |   |   |   |   |   humidity_3pm > 54
## |   |   |   |   |   |   |   |   |   |   |   |   rainfall <= 4.6: Yes (4.0)
## |   |   |   |   |   |   |   |   |   |   |   |   rainfall > 4.6: No (4.0/1.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = W
## |   |   |   |   |   |   |   |   |   |   |   humidity_9am <= 84
## |   |   |   |   |   |   |   |   |   |   |   |   temp_9am <= 11.3: Yes (3.0)
## |   |   |   |   |   |   |   |   |   |   |   |   temp_9am > 11.3: No (2.0)
## |   |   |   |   |   |   |   |   |   |   |   humidity_9am > 84: No (6.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = WNW: Yes (3.0/1.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = NW: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = NNW: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = W
## |   |   |   |   |   |   |   |   |   cloud_9am <= 5
## |   |   |   |   |   |   |   |   |   |   cloud_3pm <= 3: No (2.0)
## |   |   |   |   |   |   |   |   |   |   cloud_3pm > 3
## |   |   |   |   |   |   |   |   |   |   |   pressure_9am <= 1018.6: Yes (14.0/1.0
## |   |   |   |   |   |   |   |   |   |   |   pressure_9am > 1018.6: No (2.0)
## |   |   |   |   |   |   |   |   |   cloud_9am > 5: No (18.0/5.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = WNW
## |   |   |   |   |   |   |   |   |   humidity_9am <= 74: Yes (5.0)
## |   |   |   |   |   |   |   |   |   humidity_9am > 74
## |   |   |   |   |   |   |   |   |   |   rainfall <= 8: No (8.0)
## |   |   |   |   |   |   |   |   |   |   rainfall > 8
## |   |   |   |   |   |   |   |   |   |   |   evaporation <= 3.5: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   |   evaporation > 3.5: No (4.0/1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = NW
## |   |   |   |   |   |   |   |   |   cloud_3pm <= 5
## |   |   |   |   |   |   |   |   |   |   temp_3pm <= 14.6: No (2.0)
## |   |   |   |   |   |   |   |   |   |   temp_3pm > 14.6: Yes (4.0)
## |   |   |   |   |   |   |   |   |   cloud_3pm > 5: Yes (4.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = NNW: Yes (2.0)
## |   |   |   |   |   |   |   wind_dir_3pm = W
## |   |   |   |   |   |   |   |   wind_speed_9am <= 31
## |   |   |   |   |   |   |   |   |   wind_dir_9am = N: Yes (5.0/1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = NNE: Yes (1.0)
```

```
## |   |   |   |   |   |   |   |   wind_dir_9am = NE: No (3.0/1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = ENE: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = E: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = ESE: No (1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = SE: No (2.0/1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = SSE: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = S: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = SSW: No (1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = SW: No (7.0/3.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = WSW
## |   |   |   |   |   |   |   |   |   humidity_3pm <= 64
## |   |   |   |   |   |   |   |   |   |   pressure_3pm <= 1013.3: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   pressure_3pm > 1013.3
## |   |   |   |   |   |   |   |   |   |   |   temp_9am <= 15.3: No (7.0)
## |   |   |   |   |   |   |   |   |   |   |   temp_9am > 15.3: Yes (2.0)
## |   |   |   |   |   |   |   |   |   humidity_3pm > 64: Yes (4.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = W
## |   |   |   |   |   |   |   |   |   wind_gust_dir = N: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = NNE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = NE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = ENE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = E: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = ESE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = SE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = SSE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = S: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = SSW: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = SW
## |   |   |   |   |   |   |   |   |   |   temp_9am <= 11: No (2.0)
## |   |   |   |   |   |   |   |   |   |   temp_9am > 11: Yes (2.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = WSW
## |   |   |   |   |   |   |   |   |   |   pressure_3pm <= 1013.5: Yes (3.0)
## |   |   |   |   |   |   |   |   |   |   pressure_3pm > 1013.5: No (6.0/1.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = W
## |   |   |   |   |   |   |   |   |   |   min_temp <= 10.7
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_speed <= 54
```

```
## |   |   |   |   |   |   |   |   |   |   |   |   evaporation <= 4: No (2.0)
## |   |   |   |   |   |   |   |   |   |   |   |   evaporation > 4: Yes (5.0)
## |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_speed > 54: No (13.0/1
## |   |   |   |   |   |   |   |   |   |   |   min_temp > 10.7: Yes (5.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = WNW: No (7.0/1.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = NW: Yes (3.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = NNW: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = WNW
## |   |   |   |   |   |   |   |   |   |   cloud_3pm <= 5
## |   |   |   |   |   |   |   |   |   |   |   min_temp <= 7.1: No (3.0)
## |   |   |   |   |   |   |   |   |   |   |   min_temp > 7.1
## |   |   |   |   |   |   |   |   |   |   |   |   wind_speed_9am <= 20: Yes (5.0)
## |   |   |   |   |   |   |   |   |   |   |   |   wind_speed_9am > 20
## |   |   |   |   |   |   |   |   |   |   |   |   |   temp_3pm <= 15.7: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   |   |   |   temp_3pm > 15.7: No (2.0)
## |   |   |   |   |   |   |   |   |   |   cloud_3pm > 5: No (8.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = NW
## |   |   |   |   |   |   |   |   |   |   rainfall <= 3.1: No (8.0/3.0)
## |   |   |   |   |   |   |   |   |   |   rainfall > 3.1: Yes (10.0/2.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = NNW
## |   |   |   |   |   |   |   |   |   |   cloud_3pm <= 6: Yes (4.0/1.0)
## |   |   |   |   |   |   |   |   |   |   cloud_3pm > 6: No (2.0)
## |   |   |   |   |   |   |   |   wind_speed_9am > 31: No (9.0)
## |   |   |   |   |   |   |   wind_dir_3pm = WNW
## |   |   |   |   |   |   |   |   temp_3pm <= 12.6: No (20.0/1.0)
## |   |   |   |   |   |   |   |   temp_3pm > 12.6
## |   |   |   |   |   |   |   |   |   wind_dir_9am = N: Yes (6.0/1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = NNE: Yes (2.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = NE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = ENE: No (1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = E: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = ESE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = SE: Yes (1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = SSE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = S: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = SSW: Yes (0.0)
```

```
## |   |   |   |   |   |   |   |   wind_dir_9am = SW: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = WSW: Yes (2.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = W
## |   |   |   |   |   |   |   |   |   wind_speed_3pm <= 22: Yes (4.0)
## |   |   |   |   |   |   |   |   |   wind_speed_3pm > 22: No (6.0/1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = WNW
## |   |   |   |   |   |   |   |   |   cloud_3pm <= 3: No (2.0)
## |   |   |   |   |   |   |   |   |   cloud_3pm > 3
## |   |   |   |   |   |   |   |   |   |   sunshine <= 8.2: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   sunshine > 8.2
## |   |   |   |   |   |   |   |   |   |   |   wind_speed_9am <= 15: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_speed_9am > 15: No (5.0/1.0
## |   |   |   |   |   |   |   |   wind_dir_9am = NW
## |   |   |   |   |   |   |   |   |   rainfall <= 6.1
## |   |   |   |   |   |   |   |   |   |   humidity_3pm <= 51: Yes (3.0)
## |   |   |   |   |   |   |   |   |   |   humidity_3pm > 51: No (6.0)
## |   |   |   |   |   |   |   |   |   rainfall > 6.1: Yes (6.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = NNW
## |   |   |   |   |   |   |   |   |   temp_3pm <= 13.4: Yes (2.0)
## |   |   |   |   |   |   |   |   |   temp_3pm > 13.4: No (5.0)
## |   |   |   |   |   |   wind_dir_3pm = NW
## |   |   |   |   |   |   |   wind_speed_9am <= 24
## |   |   |   |   |   |   |   |   cloud_9am <= 5
## |   |   |   |   |   |   |   |   |   wind_dir_9am = N
## |   |   |   |   |   |   |   |   |   |   humidity_3pm <= 59: No (4.0/1.0)
## |   |   |   |   |   |   |   |   |   |   humidity_3pm > 59: Yes (3.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = NNE: Yes (1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = NE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = ENE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = E: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = ESE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = SE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = SSE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = S: No (1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = SSW: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = SW: Yes (0.0)
```

```
## |   |   |   |   |   |   |   |   |   wind_dir_9am = WSW: No (1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = W: Yes (2.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = WNW: Yes (2.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = NW
## |   |   |   |   |   |   |   |   |   |   temp_3pm <= 12.8: No (3.0)
## |   |   |   |   |   |   |   |   |   |   temp_3pm > 12.8: Yes (3.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = NNW
## |   |   |   |   |   |   |   |   |   |   wind_speed_3pm <= 22: No (3.0)
## |   |   |   |   |   |   |   |   |   |   wind_speed_3pm > 22: Yes (6.0)
## |   |   |   |   |   |   |   |   cloud_9am > 5: No (4.0)
## |   |   |   |   |   |   |   wind_speed_9am > 24: Yes (10.0)
## |   |   |   |   |   |   wind_dir_3pm = NNW
## |   |   |   |   |   |   |   wind_dir_9am = N
## |   |   |   |   |   |   |   |   humidity_3pm <= 53: Yes (2.0)
## |   |   |   |   |   |   |   |   humidity_3pm > 53: No (7.0/1.0)
## |   |   |   |   |   |   |   wind_dir_9am = NNE: Yes (0.0)
## |   |   |   |   |   |   |   wind_dir_9am = NE: No (2.0)
## |   |   |   |   |   |   |   wind_dir_9am = ENE: Yes (1.0)
## |   |   |   |   |   |   |   wind_dir_9am = E: Yes (0.0)
## |   |   |   |   |   |   |   wind_dir_9am = ESE: Yes (0.0)
## |   |   |   |   |   |   |   wind_dir_9am = SE: Yes (0.0)
## |   |   |   |   |   |   |   wind_dir_9am = SSE: Yes (0.0)
## |   |   |   |   |   |   |   wind_dir_9am = S: Yes (0.0)
## |   |   |   |   |   |   |   wind_dir_9am = SSW: Yes (0.0)
## |   |   |   |   |   |   |   wind_dir_9am = SW: Yes (0.0)
## |   |   |   |   |   |   |   wind_dir_9am = WSW: Yes (0.0)
## |   |   |   |   |   |   |   wind_dir_9am = W: Yes (0.0)
## |   |   |   |   |   |   |   wind_dir_9am = WNW: Yes (1.0)
## |   |   |   |   |   |   |   wind_dir_9am = NW: Yes (3.0)
## |   |   |   |   |   |   |   wind_dir_9am = NNW
## |   |   |   |   |   |   |   |   temp_3pm <= 16.2: Yes (6.0)
## |   |   |   |   |   |   |   |   temp_3pm > 16.2: No (2.0)
## |   |   sunshine > 9: No (1928.0/206.0)
## humidity_3pm > 66
## |   humidity_3pm <= 81
## |   |   wind_gust_speed <= 41
```

```
## |   |   |   rain_today = No
## |   |   |   |   sunshine <= 0.7
## |   |   |   |   |   pressure_3pm <= 1017.2
## |   |   |   |   |   |   wind_dir_9am = N: Yes (41.0/15.0)
## |   |   |   |   |   |   wind_dir_9am = NNE
## |   |   |   |   |   |   |   min_temp <= 17.5
## |   |   |   |   |   |   |   |   wind_speed_3pm <= 9: No (3.0/1.0)
## |   |   |   |   |   |   |   |   wind_speed_3pm > 9: Yes (12.0)
## |   |   |   |   |   |   |   min_temp > 17.5: No (3.0)
## |   |   |   |   |   |   wind_dir_9am = NE
## |   |   |   |   |   |   |   humidity_3pm <= 69: No (2.0)
## |   |   |   |   |   |   |   humidity_3pm > 69: Yes (9.0/1.0)
## |   |   |   |   |   |   wind_dir_9am = ENE
## |   |   |   |   |   |   |   cloud_3pm <= 6: No (3.0)
## |   |   |   |   |   |   |   cloud_3pm > 6
## |   |   |   |   |   |   |   |   cloud_9am <= 7
## |   |   |   |   |   |   |   |   |   humidity_3pm <= 77: No (2.0)
## |   |   |   |   |   |   |   |   |   humidity_3pm > 77: Yes (2.0)
## |   |   |   |   |   |   |   |   cloud_9am > 7: Yes (4.0)
## |   |   |   |   |   |   wind_dir_9am = E
## |   |   |   |   |   |   |   cloud_9am <= 7: Yes (11.0/1.0)
## |   |   |   |   |   |   |   cloud_9am > 7
## |   |   |   |   |   |   |   |   pressure_3pm <= 1012.6: Yes (2.0)
## |   |   |   |   |   |   |   |   pressure_3pm > 1012.6: No (4.0)
## |   |   |   |   |   |   wind_dir_9am = ESE: Yes (7.0)
## |   |   |   |   |   |   wind_dir_9am = SE
## |   |   |   |   |   |   |   cloud_3pm <= 7: No (4.0)
## |   |   |   |   |   |   |   cloud_3pm > 7: Yes (5.0)
## |   |   |   |   |   |   wind_dir_9am = SSE
## |   |   |   |   |   |   |   temp_3pm <= 19.1: No (3.0/1.0)
## |   |   |   |   |   |   |   temp_3pm > 19.1: Yes (8.0)
## |   |   |   |   |   |   wind_dir_9am = S
## |   |   |   |   |   |   |   rainfall <= 0.7
## |   |   |   |   |   |   |   |   min_temp <= 20: No (11.0)
## |   |   |   |   |   |   |   |   min_temp > 20
## |   |   |   |   |   |   |   |   |   humidity_3pm <= 71: No (3.0)
```

```
## |   |   |   |   |   |   |   |   humidity_3pm > 71: Yes (4.0)
## |   |   |   |   |   |   |   rainfall > 0.7: Yes (3.0)
## |   |   |   |   |   |   wind_dir_9am = SSW
## |   |   |   |   |   |   |   humidity_3pm <= 75: No (7.0/1.0)
## |   |   |   |   |   |   |   humidity_3pm > 75: Yes (4.0)
## |   |   |   |   |   |   wind_dir_9am = SW
## |   |   |   |   |   |   |   cloud_3pm <= 7: No (3.0)
## |   |   |   |   |   |   |   cloud_3pm > 7
## |   |   |   |   |   |   |   |   pressure_3pm <= 1013.9: Yes (3.0)
## |   |   |   |   |   |   |   |   pressure_3pm > 1013.9: No (2.0)
## |   |   |   |   |   |   wind_dir_9am = WSW
## |   |   |   |   |   |   |   wind_speed_3pm <= 11: Yes (3.0)
## |   |   |   |   |   |   |   wind_speed_3pm > 11: No (5.0/1.0)
## |   |   |   |   |   |   wind_dir_9am = W: Yes (13.0/4.0)
## |   |   |   |   |   |   wind_dir_9am = WNW: No (7.0)
## |   |   |   |   |   |   wind_dir_9am = NW
## |   |   |   |   |   |   |   max_temp <= 12.5: No (2.0)
## |   |   |   |   |   |   |   max_temp > 12.5: Yes (8.0)
## |   |   |   |   |   |   wind_dir_9am = NNW
## |   |   |   |   |   |   |   pressure_3pm <= 1014.3: No (2.0)
## |   |   |   |   |   |   |   pressure_3pm > 1014.3: Yes (2.0)
## |   |   |   |   |   pressure_3pm > 1017.2
## |   |   |   |   |   |   wind_gust_dir = N
## |   |   |   |   |   |   |   evaporation <= 1.7: No (17.0/1.0)
## |   |   |   |   |   |   |   evaporation > 1.7
## |   |   |   |   |   |   |   |   rainfall <= 0.1
## |   |   |   |   |   |   |   |   |   wind_speed_3pm <= 13: Yes (10.0/2.0)
## |   |   |   |   |   |   |   |   |   wind_speed_3pm > 13: No (2.0)
## |   |   |   |   |   |   |   |   rainfall > 0.1: No (2.0)
## |   |   |   |   |   |   wind_gust_dir = NNE
## |   |   |   |   |   |   |   wind_speed_9am <= 11
## |   |   |   |   |   |   |   |   humidity_3pm <= 78: No (10.0)
## |   |   |   |   |   |   |   |   humidity_3pm > 78: Yes (2.0)
## |   |   |   |   |   |   |   wind_speed_9am > 11: Yes (3.0)
## |   |   |   |   |   |   wind_gust_dir = NE
## |   |   |   |   |   |   |   max_temp <= 17: Yes (4.0)
```

```
## |   |   |   |   |   |   |   max_temp > 17: No (3.0)
## |   |   |   |   |   |   wind_gust_dir = ENE
## |   |   |   |   |   |   |   temp_9am <= 11.2: Yes (6.0)
## |   |   |   |   |   |   |   temp_9am > 11.2: No (6.0/1.0)
## |   |   |   |   |   |   wind_gust_dir = E
## |   |   |   |   |   |   |   evaporation <= 1.7: No (5.0)
## |   |   |   |   |   |   |   evaporation > 1.7: Yes (6.0/1.0)
## |   |   |   |   |   |   wind_gust_dir = ESE: No (15.0/5.0)
## |   |   |   |   |   |   wind_gust_dir = SE: No (16.0)
## |   |   |   |   |   |   wind_gust_dir = SSE
## |   |   |   |   |   |   |   evaporation <= 1.2: No (7.0)
## |   |   |   |   |   |   |   evaporation > 1.2: Yes (11.0/3.0)
## |   |   |   |   |   |   wind_gust_dir = S
## |   |   |   |   |   |   |   wind_speed_9am <= 17: No (16.0/1.0)
## |   |   |   |   |   |   |   wind_speed_9am > 17: Yes (2.0)
## |   |   |   |   |   |   wind_gust_dir = SSW: No (25.0/4.0)
## |   |   |   |   |   |   wind_gust_dir = SW
## |   |   |   |   |   |   |   cloud_3pm <= 7: No (13.0/2.0)
## |   |   |   |   |   |   |   cloud_3pm > 7: Yes (5.0/1.0)
## |   |   |   |   |   |   wind_gust_dir = WSW: No (18.0/1.0)
## |   |   |   |   |   |   wind_gust_dir = W: No (27.0/9.0)
## |   |   |   |   |   |   wind_gust_dir = WNW: No (8.0/1.0)
## |   |   |   |   |   |   wind_gust_dir = NW: No (6.0/1.0)
## |   |   |   |   |   |   wind_gust_dir = NNW
## |   |   |   |   |   |   |   temp_9am <= 11.1: No (9.0)
## |   |   |   |   |   |   |   temp_9am > 11.1: Yes (3.0)
## |   |   |   |   sunshine > 0.7
## |   |   |   |   |   pressure_3pm <= 1022.9
## |   |   |   |   |   |   cloud_3pm <= 5
## |   |   |   |   |   |   |   wind_speed_3pm <= 20
## |   |   |   |   |   |   |   |   max_temp <= 10: No (88.0/4.0)
## |   |   |   |   |   |   |   |   max_temp > 10
## |   |   |   |   |   |   |   |   |   temp_3pm <= 19.6
## |   |   |   |   |   |   |   |   |   |   pressure_3pm <= 1013.9
## |   |   |   |   |   |   |   |   |   |   |   pressure_9am <= 1016.7
## |   |   |   |   |   |   |   |   |   |   |   |   evaporation <= 2.5: No (32.0/3.0
```

```
## |   |   |   |   |   |   |   |   |   |   |   evaporation > 2.5
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = N: No (12.0/5
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = NNE: No (7.0/2
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = NE: Yes (1.0)
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = ENE: No (4.0)
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = E
## |   |   |   |   |   |   |   |   |   |   |   |   |   pressure_9am <= 1012.4: Y
## |   |   |   |   |   |   |   |   |   |   |   |   |   pressure_9am > 1012.4: No
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = ESE
## |   |   |   |   |   |   |   |   |   |   |   |   |   wind_speed_9am <= 15: Yes
## |   |   |   |   |   |   |   |   |   |   |   |   |   wind_speed_9am > 15: No (
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = SE
## |   |   |   |   |   |   |   |   |   |   |   |   |   min_temp <= 12.5
## |   |   |   |   |   |   |   |   |   |   |   |   |   |   wind_speed_9am <= 6:
## |   |   |   |   |   |   |   |   |   |   |   |   |   |   wind_speed_9am > 6: N
## |   |   |   |   |   |   |   |   |   |   |   |   |   min_temp > 12.5: Yes (6.0
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = SSE: Yes (8.0/
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = S
## |   |   |   |   |   |   |   |   |   |   |   |   |   wind_speed_9am <= 13: No
## |   |   |   |   |   |   |   |   |   |   |   |   |   wind_speed_9am > 13: Yes
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = SSW: No (13.0/
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = SW
## |   |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_speed <= 35: No
## |   |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_speed > 35: Yes
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = WSW
## |   |   |   |   |   |   |   |   |   |   |   |   |   pressure_9am <= 1013.7: N
## |   |   |   |   |   |   |   |   |   |   |   |   |   pressure_9am > 1013.7: Ye
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = W
## |   |   |   |   |   |   |   |   |   |   |   |   |   cloud_3pm <= 4: No (3.0)
## |   |   |   |   |   |   |   |   |   |   |   |   |   cloud_3pm > 4: Yes (6.0/1
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = WNW
## |   |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_speed <= 35: Ye
## |   |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_speed > 35: No
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = NW: No (11.0/4
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = NNW
## |   |   |   |   |   |   |   |   |   |   |   |   |   humidity_3pm <= 73
```

```
## |   |   |   |   |   |   |   |   |   |   |   |   |   |   temp_3pm <= 14.6: No
## |   |   |   |   |   |   |   |   |   |   |   |   |   |   temp_3pm > 14.6: Yes
## |   |   |   |   |   |   |   |   |   |   |   |   |   |   humidity_3pm > 73: Yes (5
## |   |   |   |   |   |   |   |   |   |   |   pressure_9am > 1016.7: Yes (31.0/8.0
## |   |   |   |   |   |   |   |   |   |   pressure_3pm > 1013.9: No (1021.0/257.0)
## |   |   |   |   |   |   |   |   |   temp_3pm > 19.6: No (1366.0/246.0)
## |   |   |   |   |   |   |   wind_speed_3pm > 20: No (852.0/117.0)
## |   |   |   |   |   |   cloud_3pm > 5
## |   |   |   |   |   |   |   pressure_3pm <= 1008.5
## |   |   |   |   |   |   |   |   humidity_3pm <= 72
## |   |   |   |   |   |   |   |   |   wind_dir_9am = N
## |   |   |   |   |   |   |   |   |   |   pressure_3pm <= 1006.1
## |   |   |   |   |   |   |   |   |   |   |   cloud_3pm <= 6: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   |   cloud_3pm > 6: No (6.0/1.0)
## |   |   |   |   |   |   |   |   |   |   pressure_3pm > 1006.1: Yes (7.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = NNE: No (3.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = NE: No (8.0/2.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = ENE
## |   |   |   |   |   |   |   |   |   |   cloud_9am <= 5: No (3.0/1.0)
## |   |   |   |   |   |   |   |   |   |   cloud_9am > 5: Yes (5.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = E: No (4.0/1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = ESE: No (9.0/3.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = SE: No (12.0/1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = SSE: No (10.0/5.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = S
## |   |   |   |   |   |   |   |   |   |   wind_gust_speed <= 33: Yes (3.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_speed > 33
## |   |   |   |   |   |   |   |   |   |   |   humidity_3pm <= 68: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   |   humidity_3pm > 68: No (2.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = SSW: No (10.0/3.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = SW: No (4.0/1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = WSW: No (7.0/1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = W
## |   |   |   |   |   |   |   |   |   |   wind_speed_3pm <= 13: No (5.0)
## |   |   |   |   |   |   |   |   |   |   wind_speed_3pm > 13: Yes (6.0/1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = WNW
```

```
## |   |   |   |   |   |   |   |   |   humidity_3pm <= 70: Yes (8.0/1.0)
## |   |   |   |   |   |   |   |   |   humidity_3pm > 70: No (4.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = NW
## |   |   |   |   |   |   |   |   |   wind_speed_3pm <= 13: Yes (2.0)
## |   |   |   |   |   |   |   |   |   wind_speed_3pm > 13: No (3.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = NNW
## |   |   |   |   |   |   |   |   |   rainfall <= 0.2: No (6.0)
## |   |   |   |   |   |   |   |   |   rainfall > 0.2: Yes (3.0/1.0)
## |   |   |   |   |   |   |   humidity_3pm > 72: Yes (132.0/44.0)
## |   |   |   |   |   |   pressure_3pm > 1008.5
## |   |   |   |   |   |   |   wind_dir_3pm = N: No (69.0/19.0)
## |   |   |   |   |   |   |   wind_dir_3pm = NNE
## |   |   |   |   |   |   |   |   wind_speed_9am <= 6: No (15.0/1.0)
## |   |   |   |   |   |   |   |   wind_speed_9am > 6
## |   |   |   |   |   |   |   |   |   wind_gust_dir = N: Yes (17.0/6.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = NNE
## |   |   |   |   |   |   |   |   |   |   wind_speed_9am <= 7: Yes (6.0/1.0)
## |   |   |   |   |   |   |   |   |   |   wind_speed_9am > 7: No (11.0/3.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = NE
## |   |   |   |   |   |   |   |   |   |   rainfall <= 0.1
## |   |   |   |   |   |   |   |   |   |   |   sunshine <= 4.1: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   |   sunshine > 4.1: No (8.0/1.0)
## |   |   |   |   |   |   |   |   |   |   rainfall > 0.1: Yes (4.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = ENE: No (3.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = E: No (1.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = ESE: Yes (1.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = SE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = SSE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = S: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = SSW: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = SW: No (2.0/1.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = WSW: Yes (1.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = W: Yes (2.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = WNW: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = NW: Yes (2.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = NNW: No (2.0)
```

```
## |   |   |   |   |   |   |   wind_dir_3pm = NE: No (95.0/37.0)
## |   |   |   |   |   |   |   wind_dir_3pm = ENE
## |   |   |   |   |   |   |   |   rainfall <= 0.5: No (117.0/35.0)
## |   |   |   |   |   |   |   |   rainfall > 0.5: Yes (10.0/3.0)
## |   |   |   |   |   |   |   wind_dir_3pm = E: No (133.0/35.0)
## |   |   |   |   |   |   |   wind_dir_3pm = ESE: No (117.0/37.0)
## |   |   |   |   |   |   |   wind_dir_3pm = SE
## |   |   |   |   |   |   |   |   max_temp <= 27.1: No (236.0/71.0)
## |   |   |   |   |   |   |   |   max_temp > 27.1: Yes (33.0/13.0)
## |   |   |   |   |   |   |   wind_dir_3pm = SSE: No (125.0/30.0)
## |   |   |   |   |   |   |   wind_dir_3pm = S: No (146.0/37.0)
## |   |   |   |   |   |   |   wind_dir_3pm = SSW
## |   |   |   |   |   |   |   |   humidity_3pm <= 78: No (60.0/11.0)
## |   |   |   |   |   |   |   |   humidity_3pm > 78: Yes (16.0/6.0)
## |   |   |   |   |   |   |   wind_dir_3pm = SW
## |   |   |   |   |   |   |   |   wind_speed_3pm <= 4: Yes (5.0)
## |   |   |   |   |   |   |   |   wind_speed_3pm > 4: No (83.0/16.0)
## |   |   |   |   |   |   |   wind_dir_3pm = WSW: No (96.0/27.0)
## |   |   |   |   |   |   |   wind_dir_3pm = W
## |   |   |   |   |   |   |   |   wind_gust_speed <= 35: No (57.0/16.0)
## |   |   |   |   |   |   |   |   wind_gust_speed > 35: Yes (34.0/12.0)
## |   |   |   |   |   |   |   wind_dir_3pm = WNW
## |   |   |   |   |   |   |   |   wind_dir_9am = N
## |   |   |   |   |   |   |   |   |   wind_speed_3pm <= 11: No (4.0)
## |   |   |   |   |   |   |   |   |   wind_speed_3pm > 11
## |   |   |   |   |   |   |   |   |   |   rainfall <= 0.1: Yes (6.0)
## |   |   |   |   |   |   |   |   |   |   rainfall > 0.1
## |   |   |   |   |   |   |   |   |   |   |   evaporation <= 3.7: No (2.0)
## |   |   |   |   |   |   |   |   |   |   |   evaporation > 3.7: Yes (4.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = NNE: Yes (3.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = NE: No (3.0/1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = ENE: No (1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = E: No (2.0/1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = ESE: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = SE: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = SSE: No (3.0/1.0)
```

```
## |   |   |   |   |   |   |   |   wind_dir_9am = S: Yes (1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = SSW: No (1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = SW: Yes (2.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = WSW
## |   |   |   |   |   |   |   |   |   rainfall <= 0.1: No (2.0)
## |   |   |   |   |   |   |   |   |   rainfall > 0.1: Yes (2.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = W: No (4.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = WNW
## |   |   |   |   |   |   |   |   |   cloud_9am <= 6: Yes (4.0/1.0)
## |   |   |   |   |   |   |   |   |   cloud_9am > 6: No (7.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = NW
## |   |   |   |   |   |   |   |   |   evaporation <= 1.2: No (3.0)
## |   |   |   |   |   |   |   |   |   evaporation > 1.2
## |   |   |   |   |   |   |   |   |   |   wind_speed_3pm <= 11: No (4.0/1.0)
## |   |   |   |   |   |   |   |   |   |   wind_speed_3pm > 11: Yes (6.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = NNW
## |   |   |   |   |   |   |   |   |   temp_9am <= 8.3: No (2.0)
## |   |   |   |   |   |   |   |   |   temp_9am > 8.3: Yes (4.0)
## |   |   |   |   |   |   |   wind_dir_3pm = NW
## |   |   |   |   |   |   |   |   humidity_9am <= 76: Yes (23.0/6.0)
## |   |   |   |   |   |   |   |   humidity_9am > 76: No (45.0/12.0)
## |   |   |   |   |   |   |   wind_dir_3pm = NNW: No (74.0/18.0)
## |   |   |   |   pressure_3pm > 1022.9: No (1421.0/180.0)
## |   |   rain_today = Yes
## |   |   |   pressure_3pm <= 1013.8
## |   |   |   |   sunshine <= 9.4
## |   |   |   |   |   humidity_3pm <= 75
## |   |   |   |   |   |   wind_gust_speed <= 30
## |   |   |   |   |   |   |   wind_dir_3pm = N
## |   |   |   |   |   |   |   |   sunshine <= 5.1: Yes (11.0/4.0)
## |   |   |   |   |   |   |   |   sunshine > 5.1: No (16.0/7.0)
## |   |   |   |   |   |   |   wind_dir_3pm = NNE: No (20.0/10.0)
## |   |   |   |   |   |   |   wind_dir_3pm = NE
## |   |   |   |   |   |   |   |   wind_speed_3pm <= 9
## |   |   |   |   |   |   |   |   |   rainfall <= 4.8: No (2.0)
## |   |   |   |   |   |   |   |   |   rainfall > 4.8: Yes (5.0)
```

```
## |   |   |   |   |   |   |   |   wind_speed_3pm > 9: No (20.0/3.0)
## |   |   |   |   |   |   |   wind_dir_3pm = ENE: No (27.0/7.0)
## |   |   |   |   |   |   |   wind_dir_3pm = E
## |   |   |   |   |   |   |   |   humidity_9am <= 78: No (8.0)
## |   |   |   |   |   |   |   |   humidity_9am > 78
## |   |   |   |   |   |   |   |   |   sunshine <= 3.9: No (2.0)
## |   |   |   |   |   |   |   |   |   sunshine > 3.9: Yes (7.0)
## |   |   |   |   |   |   |   wind_dir_3pm = ESE
## |   |   |   |   |   |   |   |   pressure_3pm <= 1006.2: Yes (2.0)
## |   |   |   |   |   |   |   |   pressure_3pm > 1006.2: No (14.0/1.0)
## |   |   |   |   |   |   |   wind_dir_3pm = SE
## |   |   |   |   |   |   |   |   cloud_3pm <= 5: Yes (7.0)
## |   |   |   |   |   |   |   |   cloud_3pm > 5: No (4.0/1.0)
## |   |   |   |   |   |   |   wind_dir_3pm = SSE
## |   |   |   |   |   |   |   |   cloud_9am <= 6: No (5.0)
## |   |   |   |   |   |   |   |   cloud_9am > 6: Yes (8.0/3.0)
## |   |   |   |   |   |   |   wind_dir_3pm = S
## |   |   |   |   |   |   |   |   cloud_3pm <= 6
## |   |   |   |   |   |   |   |   |   temp_9am <= 16.3: No (3.0)
## |   |   |   |   |   |   |   |   |   temp_9am > 16.3: Yes (3.0)
## |   |   |   |   |   |   |   |   cloud_3pm > 6: No (4.0)
## |   |   |   |   |   |   |   wind_dir_3pm = SSW: No (11.0/1.0)
## |   |   |   |   |   |   |   wind_dir_3pm = SW
## |   |   |   |   |   |   |   |   temp_3pm <= 18.5: Yes (2.0)
## |   |   |   |   |   |   |   |   temp_3pm > 18.5: No (2.0)
## |   |   |   |   |   |   |   wind_dir_3pm = WSW: No (10.0/4.0)
## |   |   |   |   |   |   |   wind_dir_3pm = W
## |   |   |   |   |   |   |   |   temp_3pm <= 21.4: Yes (3.0)
## |   |   |   |   |   |   |   |   temp_3pm > 21.4: No (2.0)
## |   |   |   |   |   |   |   wind_dir_3pm = WNW
## |   |   |   |   |   |   |   |   humidity_3pm <= 71: No (8.0/1.0)
## |   |   |   |   |   |   |   |   humidity_3pm > 71: Yes (3.0)
## |   |   |   |   |   |   |   wind_dir_3pm = NW: No (10.0/4.0)
## |   |   |   |   |   |   |   wind_dir_3pm = NNW
## |   |   |   |   |   |   |   |   wind_gust_speed <= 28
## |   |   |   |   |   |   |   |   |   temp_9am <= 8.6: No (2.0)
```

```
## |   |   |   |   |   |   |   |   |   temp_9am > 8.6: Yes (6.0/1.0)
## |   |   |   |   |   |   |   |   wind_gust_speed > 28: No (5.0)
## |   |   |   |   |   |   wind_gust_speed > 30
## |   |   |   |   |   |   |   wind_dir_9am = N: Yes (75.0/28.0)
## |   |   |   |   |   |   |   wind_dir_9am = NNE: Yes (25.0/12.0)
## |   |   |   |   |   |   |   wind_dir_9am = NE
## |   |   |   |   |   |   |   |   humidity_3pm <= 68: No (4.0)
## |   |   |   |   |   |   |   |   humidity_3pm > 68: Yes (19.0/4.0)
## |   |   |   |   |   |   |   wind_dir_9am = ENE: Yes (24.0/9.0)
## |   |   |   |   |   |   |   wind_dir_9am = E
## |   |   |   |   |   |   |   |   humidity_3pm <= 69: Yes (11.0/2.0)
## |   |   |   |   |   |   |   |   humidity_3pm > 69: No (14.0/5.0)
## |   |   |   |   |   |   |   wind_dir_9am = ESE: No (24.0/8.0)
## |   |   |   |   |   |   |   wind_dir_9am = SE
## |   |   |   |   |   |   |   |   sunshine <= 6.7: Yes (24.0/9.0)
## |   |   |   |   |   |   |   |   sunshine > 6.7: No (29.0/7.0)
## |   |   |   |   |   |   |   wind_dir_9am = SSE
## |   |   |   |   |   |   |   |   wind_speed_3pm <= 17: Yes (9.0)
## |   |   |   |   |   |   |   |   wind_speed_3pm > 17
## |   |   |   |   |   |   |   |   |   evaporation <= 3.9: No (7.0)
## |   |   |   |   |   |   |   |   |   evaporation > 3.9
## |   |   |   |   |   |   |   |   |   |   rainfall <= 6: No (11.0/4.0)
## |   |   |   |   |   |   |   |   |   |   rainfall > 6: Yes (19.0/4.0)
## |   |   |   |   |   |   |   wind_dir_9am = S: Yes (40.0/12.0)
## |   |   |   |   |   |   |   wind_dir_9am = SSW
## |   |   |   |   |   |   |   |   evaporation <= 3.3: Yes (9.0/2.0)
## |   |   |   |   |   |   |   |   evaporation > 3.3: No (30.0/12.0)
## |   |   |   |   |   |   |   wind_dir_9am = SW
## |   |   |   |   |   |   |   |   wind_gust_dir = N: No (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NNE: No (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NE: No (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = ENE: No (2.0/1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = E: Yes (1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = ESE: No (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SE: No (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SSE: No (0.0)
```

```
## |   |   |   |   |   |   |   |   wind_gust_dir = S
## |   |   |   |   |   |   |   |   |   cloud_9am <= 6: Yes (3.0)
## |   |   |   |   |   |   |   |   |   cloud_9am > 6: No (3.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SSW: No (3.0/1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SW: No (5.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = WSW
## |   |   |   |   |   |   |   |   |   humidity_9am <= 94: Yes (2.0)
## |   |   |   |   |   |   |   |   |   humidity_9am > 94: No (3.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = W
## |   |   |   |   |   |   |   |   |   min_temp <= 16.8: Yes (7.0)
## |   |   |   |   |   |   |   |   |   min_temp > 16.8: No (5.0/1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = WNW: No (2.0/1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NW: No (1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NNW: No (0.0)
## |   |   |   |   |   |   |   wind_dir_9am = WSW
## |   |   |   |   |   |   |   |   humidity_3pm <= 73: Yes (15.0/4.0)
## |   |   |   |   |   |   |   |   humidity_3pm > 73: No (3.0)
## |   |   |   |   |   |   |   wind_dir_9am = W: No (45.0/17.0)
## |   |   |   |   |   |   |   wind_dir_9am = WNW: Yes (31.0/12.0)
## |   |   |   |   |   |   |   wind_dir_9am = NW
## |   |   |   |   |   |   |   |   wind_gust_dir = N: Yes (3.0/1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NNE: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NE
## |   |   |   |   |   |   |   |   |   pressure_9am <= 1011.4: No (3.0)
## |   |   |   |   |   |   |   |   |   pressure_9am > 1011.4: Yes (3.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = ENE: No (2.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = E: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = ESE: Yes (1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SE: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SSE: Yes (1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = S: Yes (1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SSW: Yes (2.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SW: No (1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = WSW: Yes (3.0/1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = W: Yes (13.0/4.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = WNW
```

```
## |   |   |   |   |   |   |   |   |   cloud_9am <= 7
## |   |   |   |   |   |   |   |   |   |   wind_speed_9am <= 9: No (2.0)
## |   |   |   |   |   |   |   |   |   |   wind_speed_9am > 9: Yes (5.0)
## |   |   |   |   |   |   |   |   |   cloud_9am > 7: No (2.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NW: No (5.0/1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NNW: No (3.0)
## |   |   |   |   |   |   |   wind_dir_9am = NNW: No (38.0/17.0)
## |   |   |   |   |   |   humidity_3pm > 75
## |   |   |   |   |   |   |   cloud_3pm <= 6
## |   |   |   |   |   |   |   wind_dir_3pm = N: No (13.0/5.0)
## |   |   |   |   |   |   |   wind_dir_3pm = NNE
## |   |   |   |   |   |   |   |   wind_dir_9am = N: Yes (4.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = NNE
## |   |   |   |   |   |   |   |   |   temp_3pm <= 24.6: No (3.0)
## |   |   |   |   |   |   |   |   |   temp_3pm > 24.6: Yes (2.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = NE: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = ENE: No (2.0/1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = E: Yes (1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = ESE: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = SE: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = SSE: No (1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = S: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = SSW: No (2.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = SW: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = WSW: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = W: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = WNW: Yes (1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = NW: No (4.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = NNW: No (3.0)
## |   |   |   |   |   |   |   wind_dir_3pm = NE: No (16.0/5.0)
## |   |   |   |   |   |   |   wind_dir_3pm = ENE
## |   |   |   |   |   |   |   |   sunshine <= 6: Yes (4.0)
## |   |   |   |   |   |   |   |   sunshine > 6: No (6.0/1.0)
## |   |   |   |   |   |   |   wind_dir_3pm = E
## |   |   |   |   |   |   |   |   pressure_9am <= 1015.5: No (6.0/1.0)
## |   |   |   |   |   |   |   |   pressure_9am > 1015.5: Yes (3.0)
```

```
## |   |   |   |   |   |   |   wind_dir_3pm = ESE: Yes (15.0/6.0)
## |   |   |   |   |   |   |   wind_dir_3pm = SE
## |   |   |   |   |   |   |   |   sunshine <= 6: No (3.0)
## |   |   |   |   |   |   |   |   sunshine > 6: Yes (19.0/6.0)
## |   |   |   |   |   |   |   wind_dir_3pm = SSE
## |   |   |   |   |   |   |   |   pressure_3pm <= 1010.4: Yes (3.0)
## |   |   |   |   |   |   |   |   pressure_3pm > 1010.4: No (3.0)
## |   |   |   |   |   |   |   wind_dir_3pm = S: Yes (16.0/6.0)
## |   |   |   |   |   |   |   wind_dir_3pm = SSW
## |   |   |   |   |   |   |   |   wind_gust_speed <= 30: No (2.0)
## |   |   |   |   |   |   |   |   wind_gust_speed > 30: Yes (6.0/1.0)
## |   |   |   |   |   |   |   wind_dir_3pm = SW: Yes (9.0/2.0)
## |   |   |   |   |   |   |   wind_dir_3pm = WSW: Yes (5.0/1.0)
## |   |   |   |   |   |   |   wind_dir_3pm = W: Yes (9.0/2.0)
## |   |   |   |   |   |   |   wind_dir_3pm = WNW: No (9.0/4.0)
## |   |   |   |   |   |   |   wind_dir_3pm = NW: Yes (8.0/2.0)
## |   |   |   |   |   |   |   wind_dir_3pm = NNW
## |   |   |   |   |   |   |   |   humidity_9am <= 85: No (4.0/1.0)
## |   |   |   |   |   |   |   |   humidity_9am > 85: Yes (7.0)
## |   |   |   |   |   |   |   cloud_3pm > 6: Yes (250.0/68.0)
## |   |   |   |   |   sunshine > 9.4: No (64.0/15.0)
## |   |   |   |   pressure_3pm > 1013.8
## |   |   |   |   |   cloud_3pm <= 2: No (95.0/14.0)
## |   |   |   |   |   cloud_3pm > 2
## |   |   |   |   |   |   temp_3pm <= 12.4
## |   |   |   |   |   |   |   wind_gust_speed <= 33: No (177.0/22.0)
## |   |   |   |   |   |   |   wind_gust_speed > 33
## |   |   |   |   |   |   |   |   wind_gust_dir = N
## |   |   |   |   |   |   |   |   |   temp_3pm <= 10.5: Yes (2.0)
## |   |   |   |   |   |   |   |   |   temp_3pm > 10.5: No (2.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NNE: Yes (2.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NE: No (1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = ENE: No (1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = E: No (1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = ESE: No (2.0/1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SE: No (4.0/1.0)
```

```
## |   |   |   |   |   |   |   wind_gust_dir = SSE: No (3.0/1.0)
## |   |   |   |   |   |   |   wind_gust_dir = S
## |   |   |   |   |   |   |   |   evaporation <= 1: Yes (3.0)
## |   |   |   |   |   |   |   |   evaporation > 1: No (22.0/4.0)
## |   |   |   |   |   |   |   wind_gust_dir = SSW
## |   |   |   |   |   |   |   |   rainfall <= 3.5: No (4.0)
## |   |   |   |   |   |   |   |   rainfall > 3.5: Yes (7.0/1.0)
## |   |   |   |   |   |   |   wind_gust_dir = SW
## |   |   |   |   |   |   |   |   humidity_3pm <= 78
## |   |   |   |   |   |   |   |   |   temp_3pm <= 11.1: No (10.0/1.0)
## |   |   |   |   |   |   |   |   |   temp_3pm > 11.1
## |   |   |   |   |   |   |   |   |   |   wind_speed_9am <= 9: No (2.0)
## |   |   |   |   |   |   |   |   |   |   wind_speed_9am > 9: Yes (5.0)
## |   |   |   |   |   |   |   |   humidity_3pm > 78: Yes (5.0)
## |   |   |   |   |   |   |   wind_gust_dir = WSW: No (20.0/2.0)
## |   |   |   |   |   |   |   wind_gust_dir = W
## |   |   |   |   |   |   |   |   cloud_3pm <= 6
## |   |   |   |   |   |   |   |   |   min_temp <= 5.3: No (11.0/3.0)
## |   |   |   |   |   |   |   |   |   min_temp > 5.3
## |   |   |   |   |   |   |   |   |   |   temp_9am <= 8.7: Yes (3.0)
## |   |   |   |   |   |   |   |   |   |   temp_9am > 8.7: No (7.0/2.0)
## |   |   |   |   |   |   |   |   cloud_3pm > 6: Yes (13.0/5.0)
## |   |   |   |   |   |   |   wind_gust_dir = WNW
## |   |   |   |   |   |   |   |   pressure_3pm <= 1021.3: No (8.0)
## |   |   |   |   |   |   |   |   pressure_3pm > 1021.3: Yes (2.0)
## |   |   |   |   |   |   |   wind_gust_dir = NW: No (6.0/1.0)
## |   |   |   |   |   |   |   wind_gust_dir = NNW: No (4.0)
## |   |   |   |   |   |   temp_3pm > 12.4
## |   |   |   |   |   |   |   sunshine <= 0.5: Yes (126.0/48.0)
## |   |   |   |   |   |   |   sunshine > 0.5
## |   |   |   |   |   |   |   wind_gust_speed <= 28
## |   |   |   |   |   |   |   |   min_temp <= 14: No (358.0/79.0)
## |   |   |   |   |   |   |   |   min_temp > 14
## |   |   |   |   |   |   |   |   |   wind_dir_9am = N: Yes (16.0/7.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = NNE
## |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = N: No (0.0)
```

```
## |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = NNE: Yes (1.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = NE: No (1.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = ENE: No (2.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = E: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = ESE: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = SE: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = SSE: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = S: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = SSW: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = SW: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = WSW: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = W: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = WNW: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = NW: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = NNW: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = NE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = ENE
## |   |   |   |   |   |   |   |   |   |   rainfall <= 12.5: Yes (3.0)
## |   |   |   |   |   |   |   |   |   |   rainfall > 12.5: No (2.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = E: No (6.0/2.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = ESE: No (5.0/1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = SE: No (7.0/3.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = SSE
## |   |   |   |   |   |   |   |   |   |   cloud_3pm <= 7: No (11.0/1.0)
## |   |   |   |   |   |   |   |   |   |   cloud_3pm > 7: Yes (3.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = S
## |   |   |   |   |   |   |   |   |   |   wind_speed_9am <= 7: Yes (8.0)
## |   |   |   |   |   |   |   |   |   |   wind_speed_9am > 7
## |   |   |   |   |   |   |   |   |   |   |   sunshine <= 5.6: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   |   sunshine > 5.6: No (11.0/3.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = SSW: No (32.0/12.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = SW: No (21.0/7.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = WSW: No (18.0/5.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = W: Yes (14.0/5.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = WNW: No (2.0/1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = NW
```

```
## |   |   |   |   |   |   |   |   |   |   humidity_3pm <= 75: No (3.0)
## |   |   |   |   |   |   |   |   |   |   humidity_3pm > 75: Yes (2.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = NNW: Yes (2.0)
## |   |   |   |   |   |   |   wind_gust_speed > 28
## |   |   |   |   |   |   |   |   humidity_3pm <= 77
## |   |   |   |   |   |   |   |   |   rainfall <= 4.5
## |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = N
## |   |   |   |   |   |   |   |   |   |   |   rainfall <= 3.3: No (10.0/1.0)
## |   |   |   |   |   |   |   |   |   |   |   rainfall > 3.3: Yes (3.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = NNE
## |   |   |   |   |   |   |   |   |   |   |   wind_speed_9am <= 11: No (7.0/1.0
## |   |   |   |   |   |   |   |   |   |   |   wind_speed_9am > 11: Yes (3.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = NE: No (24.0/4.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = ENE
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = N: Yes (1.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = NNE: No (0.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = NE: No (0.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = ENE
## |   |   |   |   |   |   |   |   |   |   |   |   sunshine <= 8.3: No (2.0)
## |   |   |   |   |   |   |   |   |   |   |   |   sunshine > 8.3: Yes (3.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = E: No (4.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = ESE: Yes (1.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = SE: No (0.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = SSE: Yes (1.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = S: No (0.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = SSW: No (1.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = SW: Yes (1.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = WSW: Yes (1.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = W: No (2.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = WNW: No (0.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = NW: No (0.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = NNW: No (1.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = E: No (22.0/3.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = ESE: No (25.0/11.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = SE
## |   |   |   |   |   |   |   |   |   |   |   evaporation <= 4.3: Yes (18.0/7.0
```

```
## |   |   |   |   |   |   |   |   |   |   |   evaporation > 4.3
## |   |   |   |   |   |   |   |   |   |   |   |   wind_speed_9am <= 6
## |   |   |   |   |   |   |   |   |   |   |   |   |   max_temp <= 20.3: Yes (9
## |   |   |   |   |   |   |   |   |   |   |   |   |   max_temp > 20.3: No (9.0/
## |   |   |   |   |   |   |   |   |   |   |   |   wind_speed_9am > 6: No (69.0/
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = SSE: No (28.0/11.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = S: No (52.0/15.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = SSW: No (32.0/9.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = SW
## |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_speed <= 33: No (17.0/2
## |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_speed > 33
## |   |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = N: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = NNE: Yes (0.0
## |   |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = NE: Yes (0.0
## |   |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = ENE: Yes (0.0
## |   |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = E: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = ESE: Yes (0.0
## |   |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = SE: Yes (0.0
## |   |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = SSE: Yes (0.0
## |   |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = S: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = SSW: Yes (5.0
## |   |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = SW
## |   |   |   |   |   |   |   |   |   |   |   |   |   |   pressure_9am <= 1022.9: N
## |   |   |   |   |   |   |   |   |   |   |   |   |   |   pressure_9am > 1022.9: Ye
## |   |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = WSW: No (2.0/
## |   |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = W: Yes (7.0/1
## |   |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = WNW: No (1.0
## |   |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = NW: Yes (0.0
## |   |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = NNW: Yes (0.0
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = WSW
## |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_speed <= 31: No (14.0)
## |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_speed > 31
## |   |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = N
## |   |   |   |   |   |   |   |   |   |   |   |   |   |   pressure_3pm <= 1019.7: Y
## |   |   |   |   |   |   |   |   |   |   |   |   |   |   pressure_3pm > 1019.7: No
## |   |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = NNE: No (1.0)
```

```
## |   |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = NE: No (0.0)
## |   |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = ENE: No (0.0)
## |   |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = E: No (0.0)
## |   |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = ESE: No (1.0)
## |   |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = SE: No (0.0)
## |   |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = SSE: No (0.0)
## |   |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = S: No (1.0)
## |   |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = SSW: No (0.0)
## |   |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = SW: No (1.0)
## |   |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = WSW: No (4.0)
## |   |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = W
## |   |   |   |   |   |   |   |   |   |   |   |   |   |   humidity_9am <= 89: Yes (
## |   |   |   |   |   |   |   |   |   |   |   |   |   |   humidity_9am > 89: No (2
## |   |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = WNW: Yes (3.0/
## |   |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = NW: Yes (4.0/1
## |   |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = NNW: No (2.0/1
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = W: No (38.0/8.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = WNW
## |   |   |   |   |   |   |   |   |   |   |   |   cloud_9am <= 6: Yes (14.0/4.0)
## |   |   |   |   |   |   |   |   |   |   |   |   cloud_9am > 6: No (16.0/2.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = NW
## |   |   |   |   |   |   |   |   |   |   |   |   pressure_3pm <= 1020.6: Yes (12.0
## |   |   |   |   |   |   |   |   |   |   |   |   pressure_3pm > 1020.6: No (5.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = NNW: No (20.0/7.0)
## |   |   |   |   |   |   |   |   |   |   rainfall > 4.5
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = N: No (93.0/44.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = NNE
## |   |   |   |   |   |   |   |   |   |   |   |   humidity_9am <= 94: Yes (8.0)
## |   |   |   |   |   |   |   |   |   |   |   |   humidity_9am > 94
## |   |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_speed <= 37: No (5
## |   |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_speed > 37: Yes (3
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = NE
## |   |   |   |   |   |   |   |   |   |   |   |   pressure_3pm <= 1014.9: Yes (4.0
## |   |   |   |   |   |   |   |   |   |   |   |   pressure_3pm > 1014.9: No (13.0/2
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = ENE: Yes (7.0/2.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = E
```

```
## |   |   |   |   |   |   |   |   |   |   |   |   rainfall <= 11.5: No (12.0/5.0)
## |   |   |   |   |   |   |   |   |   |   |   |   rainfall > 11.5: Yes (10.0/3.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = ESE: Yes (12.0/4.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = SE: Yes (29.0/9.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = SSE: No (25.0/9.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = S: No (39.0/17.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = SSW
## |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = N: Yes (1.0)
## |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = NNE: No (0.0)
## |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = NE: No (3.0)
## |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = ENE: Yes (1.0)
## |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = E: Yes (1.0)
## |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = ESE: No (0.0)
## |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = SE: Yes (4.0/1.0
## |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = SSE: Yes (4.0/1.(
## |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = S
## |   |   |   |   |   |   |   |   |   |   |   |   |   pressure_3pm <= 1015.6: Yes (
## |   |   |   |   |   |   |   |   |   |   |   |   |   pressure_3pm > 1015.6: No (6
## |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = SSW
## |   |   |   |   |   |   |   |   |   |   |   |   |   evaporation <= 3.5: Yes (3.0
## |   |   |   |   |   |   |   |   |   |   |   |   |   evaporation > 3.5
## |   |   |   |   |   |   |   |   |   |   |   |   |   |   min_temp <= 15.1: No (6.(
## |   |   |   |   |   |   |   |   |   |   |   |   |   |   min_temp > 15.1: Yes (2.(
## |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = SW
## |   |   |   |   |   |   |   |   |   |   |   |   |   temp_9am <= 14: No (2.0)
## |   |   |   |   |   |   |   |   |   |   |   |   |   temp_9am > 14: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = WSW: No (3.0)
## |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = W
## |   |   |   |   |   |   |   |   |   |   |   |   |   cloud_3pm <= 5: No (4.0)
## |   |   |   |   |   |   |   |   |   |   |   |   |   cloud_3pm > 5
## |   |   |   |   |   |   |   |   |   |   |   |   |   |   humidity_9am <= 78: No (3
## |   |   |   |   |   |   |   |   |   |   |   |   |   |   humidity_9am > 78: Yes (1
## |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = WNW: No (0.0)
## |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = NW: Yes (1.0)
## |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = NNW: No (0.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = SW
```

```
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = N: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = NNE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = NE: No (2.0/1.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = ENE: Yes (1.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = E: Yes (1.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = ESE: No (2.0/1.0
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = SE: Yes (5.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = SSE
## |   |   |   |   |   |   |   |   |   |   |   |   temp_3pm <= 20.4: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   |   |   temp_3pm > 20.4: No (2.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = S
## |   |   |   |   |   |   |   |   |   |   |   |   cloud_9am <= 5: No (3.0)
## |   |   |   |   |   |   |   |   |   |   |   |   cloud_9am > 5: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = SSW
## |   |   |   |   |   |   |   |   |   |   |   |   sunshine <= 7.3: Yes (3.0)
## |   |   |   |   |   |   |   |   |   |   |   |   sunshine > 7.3
## |   |   |   |   |   |   |   |   |   |   |   |   |   humidity_3pm <= 73: No (
## |   |   |   |   |   |   |   |   |   |   |   |   |   humidity_3pm > 73: Yes (
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = SW: No (8.0/1.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = WSW
## |   |   |   |   |   |   |   |   |   |   |   |   wind_speed_3pm <= 15: Yes (6
## |   |   |   |   |   |   |   |   |   |   |   |   wind_speed_3pm > 15: No (4.0
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = W
## |   |   |   |   |   |   |   |   |   |   |   |   wind_speed_3pm <= 35: No (10
## |   |   |   |   |   |   |   |   |   |   |   |   wind_speed_3pm > 35: Yes (3.
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = WNW: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = NW: Yes (1.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = NNW: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = WSW
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = N: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = NNE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = NE: No (2.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = ENE: Yes (1.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = E: Yes (1.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = ESE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = SE: Yes (2.0)
```

```
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = SSE: Yes (6.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = S
## |   |   |   |   |   |   |   |   |   |   |   |   temp_3pm <= 20.7: No (2.0)
## |   |   |   |   |   |   |   |   |   |   |   |   temp_3pm > 20.7: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = SSW
## |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_speed <= 37: Yes (4
## |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_speed > 37: No (4.0
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = SW: No (9.0/1.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = WSW
## |   |   |   |   |   |   |   |   |   |   |   |   wind_speed_3pm <= 15: Yes (3
## |   |   |   |   |   |   |   |   |   |   |   |   wind_speed_3pm > 15: No (2.0
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = W: No (3.0/1.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = WNW: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = NW: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = NNW: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = W: No (33.0/14.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = WNW: No (30.0/11.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = NW
## |   |   |   |   |   |   |   |   |   |   |   wind_speed_9am <= 13: No (20.0/6
## |   |   |   |   |   |   |   |   |   |   |   wind_speed_9am > 13: Yes (9.0/3.0
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = NNW: No (19.0/8.0)
## |   |   |   |   |   |   |   |   |   humidity_3pm > 77
## |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = N: Yes (9.0/3.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = NNE: Yes (7.0/1.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = NE
## |   |   |   |   |   |   |   |   |   |   |   wind_speed_9am <= 17
## |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_speed <= 37: No (5.0)
## |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_speed > 37: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_speed_9am > 17: Yes (6.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = ENE
## |   |   |   |   |   |   |   |   |   |   |   wind_speed_3pm <= 17: Yes (4.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_speed_3pm > 17: No (3.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = E
## |   |   |   |   |   |   |   |   |   |   |   sunshine <= 7.1: Yes (7.0/1.0)
## |   |   |   |   |   |   |   |   |   |   |   sunshine > 7.1: No (8.0/1.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = ESE
```

```
## |   |   |   |   |   |   |   |   |   |   rainfall <= 4.4: No (3.0/1.0)
## |   |   |   |   |   |   |   |   |   |   rainfall > 4.4: Yes (6.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = SE
## |   |   |   |   |   |   |   |   |   |   cloud_3pm <= 7
## |   |   |   |   |   |   |   |   |   |   |   rainfall <= 2.2: No (6.0/1.0)
## |   |   |   |   |   |   |   |   |   |   |   rainfall > 2.2: Yes (26.0/10.0)
## |   |   |   |   |   |   |   |   |   |   cloud_3pm > 7: Yes (8.0/1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = SSE
## |   |   |   |   |   |   |   |   |   |   temp_9am <= 11: No (3.0/1.0)
## |   |   |   |   |   |   |   |   |   |   temp_9am > 11: Yes (22.0/3.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = S: Yes (22.0/7.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = SSW: Yes (18.0/7.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = SW: No (10.0/5.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = WSW: No (12.0/4.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = W
## |   |   |   |   |   |   |   |   |   |   cloud_3pm <= 6: No (8.0/3.0)
## |   |   |   |   |   |   |   |   |   |   cloud_3pm > 6: Yes (9.0/3.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = WNW
## |   |   |   |   |   |   |   |   |   |   humidity_3pm <= 78: Yes (4.0/1.0)
## |   |   |   |   |   |   |   |   |   |   humidity_3pm > 78: No (11.0/1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = NW
## |   |   |   |   |   |   |   |   |   |   temp_3pm <= 13.7: Yes (4.0)
## |   |   |   |   |   |   |   |   |   |   temp_3pm > 13.7: No (5.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = NNW
## |   |   |   |   |   |   |   |   |   |   wind_speed_9am <= 7: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   wind_speed_9am > 7: No (5.0)
## |   |   wind_gust_speed > 41
## |   |   |   rainfall <= 0.2
## |   |   |   |   sunshine <= 7.5
## |   |   |   |   |   pressure_3pm <= 1016.1
## |   |   |   |   |   |   min_temp <= 16.1: Yes (313.0/76.0)
## |   |   |   |   |   |   min_temp > 16.1
## |   |   |   |   |   |   |   wind_speed_3pm <= 28
## |   |   |   |   |   |   |   |   wind_dir_3pm = N: Yes (7.0/1.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = NNE: Yes (5.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = NE: Yes (12.0/5.0)
```

```
## |   |   |   |   |   |   |   wind_dir_3pm = ENE
## |   |   |   |   |   |   |   |   cloud_3pm <= 6: No (2.0)
## |   |   |   |   |   |   |   |   cloud_3pm > 6: Yes (7.0/1.0)
## |   |   |   |   |   |   |   wind_dir_3pm = E: Yes (13.0/4.0)
## |   |   |   |   |   |   |   wind_dir_3pm = ESE: Yes (6.0/1.0)
## |   |   |   |   |   |   |   wind_dir_3pm = SE
## |   |   |   |   |   |   |   |   temp_9am <= 22.1: No (5.0)
## |   |   |   |   |   |   |   |   temp_9am > 22.1: Yes (13.0/4.0)
## |   |   |   |   |   |   |   wind_dir_3pm = SSE
## |   |   |   |   |   |   |   |   pressure_3pm <= 1011.2: Yes (9.0/2.0)
## |   |   |   |   |   |   |   |   pressure_3pm > 1011.2: No (5.0)
## |   |   |   |   |   |   |   wind_dir_3pm = S
## |   |   |   |   |   |   |   |   pressure_9am <= 1011.5
## |   |   |   |   |   |   |   |   |   cloud_3pm <= 5: No (2.0)
## |   |   |   |   |   |   |   |   |   cloud_3pm > 5: Yes (6.0)
## |   |   |   |   |   |   |   |   pressure_9am > 1011.5: No (8.0)
## |   |   |   |   |   |   |   wind_dir_3pm = SSW
## |   |   |   |   |   |   |   |   pressure_9am <= 1012.1: No (6.0/1.0)
## |   |   |   |   |   |   |   |   pressure_9am > 1012.1: Yes (3.0)
## |   |   |   |   |   |   |   wind_dir_3pm = SW
## |   |   |   |   |   |   |   |   sunshine <= 4: No (2.0)
## |   |   |   |   |   |   |   |   sunshine > 4: Yes (4.0)
## |   |   |   |   |   |   |   wind_dir_3pm = WSW: Yes (9.0)
## |   |   |   |   |   |   |   wind_dir_3pm = W: Yes (14.0/4.0)
## |   |   |   |   |   |   |   wind_dir_3pm = WNW
## |   |   |   |   |   |   |   |   humidity_3pm <= 71: No (3.0/1.0)
## |   |   |   |   |   |   |   |   humidity_3pm > 71: Yes (5.0)
## |   |   |   |   |   |   |   wind_dir_3pm = NW: Yes (5.0/1.0)
## |   |   |   |   |   |   |   wind_dir_3pm = NNW
## |   |   |   |   |   |   |   |   wind_gust_speed <= 56: No (3.0/1.0)
## |   |   |   |   |   |   |   |   wind_gust_speed > 56: Yes (4.0)
## |   |   |   |   |   |   wind_speed_3pm > 28
## |   |   |   |   |   |   |   wind_gust_speed <= 63: No (57.0/16.0)
## |   |   |   |   |   |   |   wind_gust_speed > 63: Yes (16.0/6.0)
## |   |   |   |   pressure_3pm > 1016.1
## |   |   |   |   |   wind_gust_dir = N: No (30.0/6.0)
```

```
## |   |   |   |   |   |   wind_gust_dir = NNE: No (10.0/3.0)
## |   |   |   |   |   |   wind_gust_dir = NE
## |   |   |   |   |   |   |   wind_speed_3pm <= 28: Yes (11.0)
## |   |   |   |   |   |   |   wind_speed_3pm > 28: No (2.0)
## |   |   |   |   |   |   wind_gust_dir = ENE: No (8.0/2.0)
## |   |   |   |   |   |   wind_gust_dir = E
## |   |   |   |   |   |   |   pressure_3pm <= 1021.2: No (8.0/1.0)
## |   |   |   |   |   |   |   pressure_3pm > 1021.2: Yes (5.0)
## |   |   |   |   |   |   wind_gust_dir = ESE
## |   |   |   |   |   |   |   wind_speed_3pm <= 22: Yes (3.0)
## |   |   |   |   |   |   |   wind_speed_3pm > 22
## |   |   |   |   |   |   |   |   wind_speed_3pm <= 31: No (5.0)
## |   |   |   |   |   |   |   |   wind_speed_3pm > 31: Yes (3.0/1.0)
## |   |   |   |   |   |   wind_gust_dir = SE
## |   |   |   |   |   |   |   rainfall <= 0
## |   |   |   |   |   |   |   |   cloud_3pm <= 6
## |   |   |   |   |   |   |   |   |   humidity_3pm <= 67: Yes (2.0)
## |   |   |   |   |   |   |   |   |   humidity_3pm > 67: No (2.0)
## |   |   |   |   |   |   |   |   cloud_3pm > 6: Yes (3.0)
## |   |   |   |   |   |   |   rainfall > 0: No (5.0)
## |   |   |   |   |   |   wind_gust_dir = SSE
## |   |   |   |   |   |   |   wind_dir_3pm = N: No (0.0)
## |   |   |   |   |   |   |   wind_dir_3pm = NNE: No (0.0)
## |   |   |   |   |   |   |   wind_dir_3pm = NE: No (0.0)
## |   |   |   |   |   |   |   wind_dir_3pm = ENE: No (0.0)
## |   |   |   |   |   |   |   wind_dir_3pm = E: No (0.0)
## |   |   |   |   |   |   |   wind_dir_3pm = ESE: No (1.0)
## |   |   |   |   |   |   |   wind_dir_3pm = SE: Yes (3.0)
## |   |   |   |   |   |   |   wind_dir_3pm = SSE
## |   |   |   |   |   |   |   |   wind_gust_speed <= 43: Yes (2.0)
## |   |   |   |   |   |   |   |   wind_gust_speed > 43: No (6.0)
## |   |   |   |   |   |   |   wind_dir_3pm = S
## |   |   |   |   |   |   |   |   temp_3pm <= 19.2: No (4.0)
## |   |   |   |   |   |   |   |   temp_3pm > 19.2: Yes (2.0)
## |   |   |   |   |   |   |   wind_dir_3pm = SSW: No (0.0)
## |   |   |   |   |   |   |   wind_dir_3pm = SW: No (0.0)
```

```
## |   |   |   |   |   |   |   wind_dir_3pm = WSW: No (0.0)
## |   |   |   |   |   |   |   wind_dir_3pm = W: No (0.0)
## |   |   |   |   |   |   |   wind_dir_3pm = WNW: No (0.0)
## |   |   |   |   |   |   |   wind_dir_3pm = NW: No (0.0)
## |   |   |   |   |   |   |   wind_dir_3pm = NNW: No (0.0)
## |   |   |   |   |   |   wind_gust_dir = S
## |   |   |   |   |   |   |   wind_dir_3pm = N: Yes (0.0)
## |   |   |   |   |   |   |   wind_dir_3pm = NNE: No (1.0)
## |   |   |   |   |   |   |   wind_dir_3pm = NE: Yes (0.0)
## |   |   |   |   |   |   |   wind_dir_3pm = ENE: Yes (0.0)
## |   |   |   |   |   |   |   wind_dir_3pm = E: Yes (0.0)
## |   |   |   |   |   |   |   wind_dir_3pm = ESE: No (1.0)
## |   |   |   |   |   |   |   wind_dir_3pm = SE: Yes (0.0)
## |   |   |   |   |   |   |   wind_dir_3pm = SSE: No (4.0/1.0)
## |   |   |   |   |   |   |   wind_dir_3pm = S
## |   |   |   |   |   |   |   |   humidity_3pm <= 73: No (6.0)
## |   |   |   |   |   |   |   |   humidity_3pm > 73
## |   |   |   |   |   |   |   |   |   wind_gust_speed <= 50: Yes (6.0)
## |   |   |   |   |   |   |   |   |   wind_gust_speed > 50: No (3.0)
## |   |   |   |   |   |   |   wind_dir_3pm = SSW: Yes (8.0)
## |   |   |   |   |   |   |   wind_dir_3pm = SW: Yes (1.0)
## |   |   |   |   |   |   |   wind_dir_3pm = WSW: Yes (0.0)
## |   |   |   |   |   |   |   wind_dir_3pm = W: Yes (0.0)
## |   |   |   |   |   |   |   wind_dir_3pm = WNW: Yes (0.0)
## |   |   |   |   |   |   |   wind_dir_3pm = NW: Yes (0.0)
## |   |   |   |   |   |   |   wind_dir_3pm = NNW: Yes (0.0)
## |   |   |   |   |   |   wind_gust_dir = SSW
## |   |   |   |   |   |   |   wind_gust_speed <= 52: No (23.0/8.0)
## |   |   |   |   |   |   |   wind_gust_speed > 52: Yes (9.0/2.0)
## |   |   |   |   |   |   wind_gust_dir = SW
## |   |   |   |   |   |   |   cloud_3pm <= 7: No (10.0/2.0)
## |   |   |   |   |   |   |   cloud_3pm > 7: Yes (2.0)
## |   |   |   |   |   |   wind_gust_dir = WSW: No (16.0/5.0)
## |   |   |   |   |   |   wind_gust_dir = W
## |   |   |   |   |   |   |   wind_dir_9am = N: Yes (3.0/1.0)
## |   |   |   |   |   |   |   wind_dir_9am = NNE: Yes (1.0)
```

```
## |   |   |   |   |   |   |   wind_dir_9am = NE: No (0.0)
## |   |   |   |   |   |   |   wind_dir_9am = ENE: No (0.0)
## |   |   |   |   |   |   |   wind_dir_9am = E: No (0.0)
## |   |   |   |   |   |   |   wind_dir_9am = ESE: No (0.0)
## |   |   |   |   |   |   |   wind_dir_9am = SE: No (0.0)
## |   |   |   |   |   |   |   wind_dir_9am = SSE: No (0.0)
## |   |   |   |   |   |   |   wind_dir_9am = S: No (0.0)
## |   |   |   |   |   |   |   wind_dir_9am = SSW: No (0.0)
## |   |   |   |   |   |   |   wind_dir_9am = SW: No (0.0)
## |   |   |   |   |   |   |   wind_dir_9am = WSW: No (0.0)
## |   |   |   |   |   |   |   wind_dir_9am = W: No (5.0/1.0)
## |   |   |   |   |   |   |   wind_dir_9am = WNW
## |   |   |   |   |   |   |   |   cloud_3pm <= 7: Yes (4.0)
## |   |   |   |   |   |   |   |   cloud_3pm > 7: No (3.0)
## |   |   |   |   |   |   |   wind_dir_9am = NW: No (3.0)
## |   |   |   |   |   |   |   wind_dir_9am = NNW: Yes (2.0)
## |   |   |   |   |   |   wind_gust_dir = WNW
## |   |   |   |   |   |   |   pressure_3pm <= 1019.4: No (3.0)
## |   |   |   |   |   |   |   pressure_3pm > 1019.4: Yes (5.0)
## |   |   |   |   |   |   wind_gust_dir = NW
## |   |   |   |   |   |   |   pressure_9am <= 1020.5: Yes (2.0)
## |   |   |   |   |   |   |   pressure_9am > 1020.5: No (3.0)
## |   |   |   |   |   |   wind_gust_dir = NNW: Yes (4.0/1.0)
## |   |   |   |   sunshine > 7.5
## |   |   |   |   |   wind_gust_speed <= 54
## |   |   |   |   |   |   wind_speed_3pm <= 28
## |   |   |   |   |   |   |   wind_gust_speed <= 46
## |   |   |   |   |   |   |   |   humidity_3pm <= 67: No (52.0/3.0)
## |   |   |   |   |   |   |   |   humidity_3pm > 67
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = N
## |   |   |   |   |   |   |   |   |   |   rainfall <= 0.1: No (28.0/8.0)
## |   |   |   |   |   |   |   |   |   |   rainfall > 0.1
## |   |   |   |   |   |   |   |   |   |   |   min_temp <= 4: No (3.0/1.0)
## |   |   |   |   |   |   |   |   |   |   |   min_temp > 4: Yes (5.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = NNE
## |   |   |   |   |   |   |   |   |   |   temp_9am <= 17.9
```

```
## |   |   |   |   |   |   |   |   |   |   wind_gust_speed <= 43: Yes (6.0/1.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_speed > 43: No (7.0/1.0)
## |   |   |   |   |   |   |   |   |   |   temp_9am > 17.9: No (32.0/1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = NE
## |   |   |   |   |   |   |   |   |   |   cloud_3pm <= 6: No (49.0/5.0)
## |   |   |   |   |   |   |   |   |   |   cloud_3pm > 6
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_speed <= 43: No (4.0/1.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_speed > 43: Yes (4.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = ENE: No (28.0/6.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = E
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = N: No (1.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = NNE: No (1.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = NE: Yes (1.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = ENE: No (4.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = E
## |   |   |   |   |   |   |   |   |   |   |   humidity_9am <= 70: Yes (3.0)
## |   |   |   |   |   |   |   |   |   |   |   humidity_9am > 70: No (4.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = ESE
## |   |   |   |   |   |   |   |   |   |   |   wind_speed_3pm <= 20: No (3.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_speed_3pm > 20: Yes (4.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = SE: No (2.0/1.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = SSE: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = S: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = SSW: No (2.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = SW: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = WSW: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = W: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = WNW: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = NW: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = NNW: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = ESE: No (26.0/6.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = SE: No (47.0/12.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = SSE
## |   |   |   |   |   |   |   |   |   |   wind_speed_3pm <= 22
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = N: No (0.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = NNE: No (0.0)
```

```
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = NE: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = ENE: Yes (1.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = E: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = ESE: Yes (1.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = SE
## |   |   |   |   |   |   |   |   |   |   |   humidity_3pm <= 71: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   |   humidity_3pm > 71: No (2.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = SSE: No (10.0/1.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = S
## |   |   |   |   |   |   |   |   |   |   |   humidity_3pm <= 71: No (3.0)
## |   |   |   |   |   |   |   |   |   |   |   humidity_3pm > 71: Yes (4.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = SSW: No (3.0/1.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = SW: Yes (1.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = WSW: No (2.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = W: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = WNW: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = NW: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = NNW: Yes (1.0)
## |   |   |   |   |   |   |   |   |   |   wind_speed_3pm > 22: No (14.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = S: No (34.0/12.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = SSW
## |   |   |   |   |   |   |   |   |   |   min_temp <= 17.3
## |   |   |   |   |   |   |   |   |   |   |   cloud_9am <= 6: No (13.0)
## |   |   |   |   |   |   |   |   |   |   |   cloud_9am > 6
## |   |   |   |   |   |   |   |   |   |   |   |   temp_3pm <= 19.2: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   |   |   temp_3pm > 19.2: No (2.0)
## |   |   |   |   |   |   |   |   |   |   min_temp > 17.3: Yes (3.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = SW
## |   |   |   |   |   |   |   |   |   |   humidity_9am <= 68: Yes (5.0)
## |   |   |   |   |   |   |   |   |   |   humidity_9am > 68
## |   |   |   |   |   |   |   |   |   |   |   wind_speed_9am <= 0: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_speed_9am > 0: No (11.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = WSW
## |   |   |   |   |   |   |   |   |   |   wind_speed_3pm <= 20: Yes (9.0/2.0)
## |   |   |   |   |   |   |   |   |   |   wind_speed_3pm > 20: No (4.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = W: No (31.0/12.0)
```

```
## |   |   |   |   |   |   |   |   wind_dir_3pm = WNW
## |   |   |   |   |   |   |   |   |   pressure_9am <= 1019.2
## |   |   |   |   |   |   |   |   |   |   temp_3pm <= 10.3: No (3.0)
## |   |   |   |   |   |   |   |   |   |   temp_3pm > 10.3: Yes (14.0/1.0)
## |   |   |   |   |   |   |   |   |   pressure_9am > 1019.2: No (3.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = NW
## |   |   |   |   |   |   |   |   |   rainfall <= 0.1
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = N: Yes (1.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = NNE: No (1.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = NE: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = ENE: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = E: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = ESE: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = SE: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = SSE: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = S: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = SSW: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = SW: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = WSW: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = W: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = WNW: No (1.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = NW: Yes (8.0/2.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = NNW: No (3.0)
## |   |   |   |   |   |   |   |   |   rainfall > 0.1: No (6.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = NNW
## |   |   |   |   |   |   |   |   |   wind_gust_dir = N: Yes (3.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = NNE: Yes (1.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = NE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = ENE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = E: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = ESE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = SE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = SSE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = S: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = SSW: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = SW: No (0.0)
```

```
## |   |   |   |   |   |   |   |   |   wind_gust_dir = WSW: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = W: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = WNW: No (3.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = NW
## |   |   |   |   |   |   |   |   |   |   temp_3pm <= 18: No (5.0)
## |   |   |   |   |   |   |   |   |   |   temp_3pm > 18: Yes (3.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = NNW
## |   |   |   |   |   |   |   |   |   |   wind_speed_9am <= 19: No (14.0/4.0)
## |   |   |   |   |   |   |   |   |   |   wind_speed_9am > 19: Yes (6.0/1.0)
## |   |   |   |   |   |   |   wind_gust_speed > 46
## |   |   |   |   |   |   |   |   wind_dir_3pm = N
## |   |   |   |   |   |   |   |   |   cloud_3pm <= 7: No (21.0/5.0)
## |   |   |   |   |   |   |   |   |   cloud_3pm > 7: Yes (5.0/1.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = NNE
## |   |   |   |   |   |   |   |   |   temp_3pm <= 20.6: Yes (11.0/1.0)
## |   |   |   |   |   |   |   |   |   temp_3pm > 20.6
## |   |   |   |   |   |   |   |   |   |   humidity_3pm <= 70: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   humidity_3pm > 70: No (15.0/2.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = NE
## |   |   |   |   |   |   |   |   |   humidity_9am <= 79: Yes (23.0/9.0)
## |   |   |   |   |   |   |   |   |   humidity_9am > 79: No (10.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = ENE: No (22.0/9.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = E: Yes (12.0/5.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = ESE: No (24.0/5.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = SE
## |   |   |   |   |   |   |   |   |   temp_9am <= 24.3: No (26.0/10.0)
## |   |   |   |   |   |   |   |   |   temp_9am > 24.3: Yes (4.0/1.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = SSE
## |   |   |   |   |   |   |   |   |   wind_dir_9am = N: Yes (1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = NNE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = NE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = ENE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = E: No (1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = ESE: Yes (1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = SE: No (3.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = SSE: No (5.0/1.0)
```

```
## |   |   |   |   |   |   |   |   wind_dir_9am = S: No (12.0/2.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = SSW: Yes (4.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = SW: Yes (3.0/1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = WSW: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = W: Yes (1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = WNW: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = NW: No (2.0/1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = NNW: No (0.0)
## |   |   |   |   |   |   |   wind_dir_3pm = S: No (34.0/13.0)
## |   |   |   |   |   |   |   wind_dir_3pm = SSW
## |   |   |   |   |   |   |   |   wind_gust_speed <= 52
## |   |   |   |   |   |   |   |   |   humidity_3pm <= 70: No (4.0)
## |   |   |   |   |   |   |   |   |   humidity_3pm > 70: Yes (11.0/2.0)
## |   |   |   |   |   |   |   |   wind_gust_speed > 52: No (4.0)
## |   |   |   |   |   |   |   wind_dir_3pm = SW
## |   |   |   |   |   |   |   |   wind_gust_speed <= 52: No (13.0/1.0)
## |   |   |   |   |   |   |   |   wind_gust_speed > 52: Yes (3.0)
## |   |   |   |   |   |   |   wind_dir_3pm = WSW: No (15.0/6.0)
## |   |   |   |   |   |   |   wind_dir_3pm = W
## |   |   |   |   |   |   |   |   wind_gust_dir = N: Yes (1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NNE: Yes (1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NE: No (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = ENE: No (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = E: No (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = ESE: No (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SE: No (1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SSE: No (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = S: No (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SSW: Yes (1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SW: No (2.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = WSW: Yes (3.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = W: No (6.0/1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = WNW: No (7.0/1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NW: No (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NNW: No (0.0)
## |   |   |   |   |   |   |   wind_dir_3pm = WNW
```

```
## |   |   |   |   |   |   |   |   evaporation <= 4.4: No (4.0)
## |   |   |   |   |   |   |   |   evaporation > 4.4: Yes (20.0/9.0)
## |   |   |   |   |   |   |   wind_dir_3pm = NW
## |   |   |   |   |   |   |   |   evaporation <= 4.6: No (3.0)
## |   |   |   |   |   |   |   |   evaporation > 4.6: Yes (27.0/7.0)
## |   |   |   |   |   |   |   wind_dir_3pm = NNW
## |   |   |   |   |   |   |   |   wind_dir_9am = N: Yes (7.0/1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = NNE: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = NE: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = ENE: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = E: No (1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = ESE: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = SE: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = SSE: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = S: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = SSW: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = SW: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = WSW: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = W: No (1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = WNW: No (1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = NW
## |   |   |   |   |   |   |   |   |   wind_speed_3pm <= 24: Yes (5.0)
## |   |   |   |   |   |   |   |   |   wind_speed_3pm > 24: No (2.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = NNW
## |   |   |   |   |   |   |   |   |   humidity_9am <= 87: No (5.0)
## |   |   |   |   |   |   |   |   |   humidity_9am > 87: Yes (2.0)
## |   |   |   |   |   |   wind_speed_3pm > 28: No (544.0/108.0)
## |   |   |   |   |   wind_gust_speed > 54
## |   |   |   |   |   |   cloud_3pm <= 6
## |   |   |   |   |   |   |   wind_gust_dir = N
## |   |   |   |   |   |   |   |   wind_dir_9am = N
## |   |   |   |   |   |   |   |   |   humidity_9am <= 97
## |   |   |   |   |   |   |   |   |   |   wind_gust_speed <= 59: No (3.0/1.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_speed > 59: Yes (7.0)
## |   |   |   |   |   |   |   |   |   humidity_9am > 97: No (3.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = NNE: No (5.0/1.0)
```

```
## |   |   |   |   |   |   |   wind_dir_9am = NE: Yes (0.0)
## |   |   |   |   |   |   |   wind_dir_9am = ENE: Yes (0.0)
## |   |   |   |   |   |   |   wind_dir_9am = E: Yes (0.0)
## |   |   |   |   |   |   |   wind_dir_9am = ESE: Yes (0.0)
## |   |   |   |   |   |   |   wind_dir_9am = SE: Yes (1.0)
## |   |   |   |   |   |   |   wind_dir_9am = SSE: Yes (0.0)
## |   |   |   |   |   |   |   wind_dir_9am = S: Yes (0.0)
## |   |   |   |   |   |   |   wind_dir_9am = SSW: Yes (0.0)
## |   |   |   |   |   |   |   wind_dir_9am = SW: Yes (0.0)
## |   |   |   |   |   |   |   wind_dir_9am = WSW: Yes (0.0)
## |   |   |   |   |   |   |   wind_dir_9am = W: Yes (0.0)
## |   |   |   |   |   |   |   wind_dir_9am = WNW: Yes (0.0)
## |   |   |   |   |   |   |   wind_dir_9am = NW: No (1.0)
## |   |   |   |   |   |   |   wind_dir_9am = NNW: Yes (2.0)
## |   |   |   |   |   |   wind_gust_dir = NNE
## |   |   |   |   |   |   |   temp_3pm <= 19.1: Yes (14.0/2.0)
## |   |   |   |   |   |   |   temp_3pm > 19.1
## |   |   |   |   |   |   |   |   rainfall <= 0.1: No (24.0)
## |   |   |   |   |   |   |   |   rainfall > 0.1: Yes (2.0)
## |   |   |   |   |   |   wind_gust_dir = NE: No (41.0/3.0)
## |   |   |   |   |   |   wind_gust_dir = ENE: Yes (4.0/1.0)
## |   |   |   |   |   |   wind_gust_dir = E: No (10.0/3.0)
## |   |   |   |   |   |   wind_gust_dir = ESE: No (3.0)
## |   |   |   |   |   |   wind_gust_dir = SE
## |   |   |   |   |   |   |   pressure_9am <= 1017.8: No (14.0/6.0)
## |   |   |   |   |   |   |   pressure_9am > 1017.8: Yes (10.0/1.0)
## |   |   |   |   |   |   wind_gust_dir = SSE
## |   |   |   |   |   |   |   wind_speed_3pm <= 22: No (9.0)
## |   |   |   |   |   |   |   wind_speed_3pm > 22
## |   |   |   |   |   |   |   |   temp_3pm <= 25.9
## |   |   |   |   |   |   |   |   |   pressure_9am <= 1015.9
## |   |   |   |   |   |   |   |   |   |   pressure_9am <= 1008.1: Yes (4.0/1.0
## |   |   |   |   |   |   |   |   |   |   pressure_9am > 1008.1: No (13.0)
## |   |   |   |   |   |   |   |   |   pressure_9am > 1015.9
## |   |   |   |   |   |   |   |   |   |   temp_3pm <= 18.4: No (2.0)
## |   |   |   |   |   |   |   |   |   |   temp_3pm > 18.4: Yes (12.0/1.0)
```

```
## |   |   |   |   |   |   |   |   temp_3pm > 25.9: Yes (9.0)
## |   |   |   |   |   |   wind_gust_dir = S
## |   |   |   |   |   |   |   wind_gust_speed <= 61: No (35.0/6.0)
## |   |   |   |   |   |   |   wind_gust_speed > 61
## |   |   |   |   |   |   |   |   cloud_3pm <= 2: No (3.0)
## |   |   |   |   |   |   |   |   cloud_3pm > 2: Yes (44.0/19.0)
## |   |   |   |   |   |   wind_gust_dir = SSW
## |   |   |   |   |   |   |   wind_dir_9am = N: No (3.0/1.0)
## |   |   |   |   |   |   |   wind_dir_9am = NNE
## |   |   |   |   |   |   |   |   min_temp <= 20.9: No (5.0)
## |   |   |   |   |   |   |   |   min_temp > 20.9: Yes (3.0)
## |   |   |   |   |   |   |   wind_dir_9am = NE: No (1.0)
## |   |   |   |   |   |   |   wind_dir_9am = ENE: No (3.0)
## |   |   |   |   |   |   |   wind_dir_9am = E: No (0.0)
## |   |   |   |   |   |   |   wind_dir_9am = ESE: No (0.0)
## |   |   |   |   |   |   |   wind_dir_9am = SE: No (0.0)
## |   |   |   |   |   |   |   wind_dir_9am = SSE: No (2.0/1.0)
## |   |   |   |   |   |   |   wind_dir_9am = S: No (3.0/1.0)
## |   |   |   |   |   |   |   wind_dir_9am = SSW
## |   |   |   |   |   |   |   |   wind_gust_speed <= 57: No (3.0)
## |   |   |   |   |   |   |   |   wind_gust_speed > 57: Yes (2.0)
## |   |   |   |   |   |   |   wind_dir_9am = SW
## |   |   |   |   |   |   |   |   humidity_3pm <= 68: No (2.0)
## |   |   |   |   |   |   |   |   humidity_3pm > 68: Yes (3.0)
## |   |   |   |   |   |   |   wind_dir_9am = WSW: Yes (5.0)
## |   |   |   |   |   |   |   wind_dir_9am = W: No (1.0)
## |   |   |   |   |   |   |   wind_dir_9am = WNW: No (0.0)
## |   |   |   |   |   |   |   wind_dir_9am = NW: Yes (3.0/1.0)
## |   |   |   |   |   |   |   wind_dir_9am = NNW: No (2.0)
## |   |   |   |   |   |   wind_gust_dir = SW
## |   |   |   |   |   |   |   max_temp <= 19.5: No (6.0)
## |   |   |   |   |   |   |   max_temp > 19.5: Yes (11.0/2.0)
## |   |   |   |   |   |   wind_gust_dir = WSW
## |   |   |   |   |   |   |   temp_9am <= 7.5: No (7.0)
## |   |   |   |   |   |   |   temp_9am > 7.5: Yes (21.0/6.0)
## |   |   |   |   |   |   wind_gust_dir = W
```

```
## |   |   |   |   |   |   |   humidity_9am <= 79: Yes (23.0/11.0)
## |   |   |   |   |   |   |   humidity_9am > 79: No (16.0)
## |   |   |   |   |   |   wind_gust_dir = WNW
## |   |   |   |   |   |   |   wind_speed_3pm <= 33
## |   |   |   |   |   |   |   |   humidity_9am <= 70: Yes (12.0)
## |   |   |   |   |   |   |   |   humidity_9am > 70
## |   |   |   |   |   |   |   |   |   wind_dir_9am = N: No (2.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = NNE: No (2.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = NE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = ENE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = E: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = ESE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = SE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = SSE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = S: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = SSW: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = SW: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = WSW: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = W: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = WNW: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = NW: Yes (3.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = NNW: No (1.0)
## |   |   |   |   |   |   |   |   wind_speed_3pm > 33: No (7.0)
## |   |   |   |   |   |   |   wind_gust_dir = NW
## |   |   |   |   |   |   |   |   wind_dir_9am = N: Yes (6.0/1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = NNE: No (2.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = NE: No (1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = ENE: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = E: No (1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = ESE: No (1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = SE: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = SSE: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = S: No (1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = SSW: Yes (1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = SW: No (1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = WSW: No (1.0)
```

```
## |   |   |   |   |   |   |   wind_dir_9am = W: No (1.0)
## |   |   |   |   |   |   |   wind_dir_9am = WNW: No (3.0)
## |   |   |   |   |   |   |   wind_dir_9am = NW
## |   |   |   |   |   |   |   |   pressure_9am <= 1008.5: No (5.0)
## |   |   |   |   |   |   |   |   pressure_9am > 1008.5: Yes (7.0/1.0)
## |   |   |   |   |   |   |   wind_dir_9am = NNW
## |   |   |   |   |   |   |   |   min_temp <= 9.2: No (3.0/1.0)
## |   |   |   |   |   |   |   |   min_temp > 9.2: Yes (4.0)
## |   |   |   |   |   |   wind_gust_dir = NNW: Yes (16.0/5.0)
## |   |   |   |   |   cloud_3pm > 6
## |   |   |   |   |   |   pressure_3pm <= 1004.2: Yes (16.0)
## |   |   |   |   |   |   pressure_3pm > 1004.2
## |   |   |   |   |   |   |   wind_dir_9am = N
## |   |   |   |   |   |   |   |   pressure_3pm <= 1011.3: Yes (6.0)
## |   |   |   |   |   |   |   |   pressure_3pm > 1011.3: No (16.0/5.0)
## |   |   |   |   |   |   |   wind_dir_9am = NNE
## |   |   |   |   |   |   |   |   temp_3pm <= 13.1: No (3.0/1.0)
## |   |   |   |   |   |   |   |   temp_3pm > 13.1: Yes (9.0)
## |   |   |   |   |   |   |   wind_dir_9am = NE: Yes (8.0)
## |   |   |   |   |   |   |   wind_dir_9am = ENE
## |   |   |   |   |   |   |   |   humidity_3pm <= 72: No (3.0/1.0)
## |   |   |   |   |   |   |   |   humidity_3pm > 72: Yes (5.0)
## |   |   |   |   |   |   |   wind_dir_9am = E: Yes (4.0/1.0)
## |   |   |   |   |   |   |   wind_dir_9am = ESE: Yes (0.0)
## |   |   |   |   |   |   |   wind_dir_9am = SE
## |   |   |   |   |   |   |   |   temp_3pm <= 20.3: Yes (2.0)
## |   |   |   |   |   |   |   |   temp_3pm > 20.3: No (3.0/1.0)
## |   |   |   |   |   |   |   wind_dir_9am = SSE
## |   |   |   |   |   |   |   |   cloud_9am <= 6: No (6.0/1.0)
## |   |   |   |   |   |   |   |   cloud_9am > 6: Yes (4.0)
## |   |   |   |   |   |   |   wind_dir_9am = S
## |   |   |   |   |   |   |   |   wind_dir_3pm = N: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = NNE: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = NE: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = ENE: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = E: No (0.0)
```

```
## |   |   |   |   |   |   |   |   wind_dir_3pm = ESE: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = SE: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = SSE: No (4.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = S
## |   |   |   |   |   |   |   |   |   cloud_9am <= 6: Yes (3.0)
## |   |   |   |   |   |   |   |   |   cloud_9am > 6: No (2.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = SSW: Yes (1.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = SW: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = WSW: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = W: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = WNW: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = NW: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = NNW: No (0.0)
## |   |   |   |   |   |   |   wind_dir_9am = SSW
## |   |   |   |   |   |   |   |   sunshine <= 8.1: No (2.0)
## |   |   |   |   |   |   |   |   sunshine > 8.1
## |   |   |   |   |   |   |   |   |   wind_speed_9am <= 17: No (3.0)
## |   |   |   |   |   |   |   |   |   wind_speed_9am > 17
## |   |   |   |   |   |   |   |   |   |   wind_gust_speed <= 57: No (3.0/1.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_speed > 57: Yes (6.0)
## |   |   |   |   |   |   |   wind_dir_9am = SW
## |   |   |   |   |   |   |   |   wind_speed_3pm <= 30: Yes (7.0)
## |   |   |   |   |   |   |   |   wind_speed_3pm > 30
## |   |   |   |   |   |   |   |   |   cloud_3pm <= 7: Yes (2.0)
## |   |   |   |   |   |   |   |   |   cloud_3pm > 7: No (7.0/2.0)
## |   |   |   |   |   |   |   wind_dir_9am = WSW
## |   |   |   |   |   |   |   |   humidity_3pm <= 72: Yes (4.0)
## |   |   |   |   |   |   |   |   humidity_3pm > 72: No (3.0)
## |   |   |   |   |   |   |   wind_dir_9am = W
## |   |   |   |   |   |   |   |   evaporation <= 6.1: Yes (18.0/6.0)
## |   |   |   |   |   |   |   |   evaporation > 6.1: No (3.0)
## |   |   |   |   |   |   |   wind_dir_9am = WNW
## |   |   |   |   |   |   |   |   wind_speed_9am <= 22: Yes (2.0)
## |   |   |   |   |   |   |   |   wind_speed_9am > 22: No (4.0)
## |   |   |   |   |   |   |   wind_dir_9am = NW
## |   |   |   |   |   |   |   |   wind_speed_3pm <= 31: Yes (7.0)
```

```
## |   |   |   |   |   |   |   |   wind_speed_3pm > 31
## |   |   |   |   |   |   |   |   |   wind_speed_3pm <= 43: No (4.0)
## |   |   |   |   |   |   |   |   |   wind_speed_3pm > 43: Yes (2.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = NNW: No (14.0/6.0)
## |   |   |   rainfall > 0.2
## |   |   |   |   wind_gust_speed <= 54
## |   |   |   |   |   wind_speed_3pm <= 24
## |   |   |   |   |   |   pressure_3pm <= 1013.1: Yes (555.0/151.0)
## |   |   |   |   |   |   pressure_3pm > 1013.1
## |   |   |   |   |   |   |   humidity_3pm <= 73
## |   |   |   |   |   |   |   |   wind_dir_3pm = N
## |   |   |   |   |   |   |   |   |   temp_9am <= 8.6: No (3.0/1.0)
## |   |   |   |   |   |   |   |   |   temp_9am > 8.6: Yes (7.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = NNE
## |   |   |   |   |   |   |   |   |   pressure_3pm <= 1016.2: Yes (2.0)
## |   |   |   |   |   |   |   |   |   pressure_3pm > 1016.2: No (3.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = NE
## |   |   |   |   |   |   |   |   |   evaporation <= 4: Yes (2.0)
## |   |   |   |   |   |   |   |   |   evaporation > 4
## |   |   |   |   |   |   |   |   |   |   humidity_3pm <= 72: No (5.0)
## |   |   |   |   |   |   |   |   |   |   humidity_3pm > 72: Yes (2.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = ENE: No (15.0/6.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = E
## |   |   |   |   |   |   |   |   |   humidity_9am <= 91: Yes (13.0/3.0)
## |   |   |   |   |   |   |   |   |   humidity_9am > 91: No (5.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = ESE: No (15.0/7.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = SE: No (18.0/9.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = SSE: Yes (41.0/15.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = S: Yes (43.0/15.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = SSW
## |   |   |   |   |   |   |   |   |   wind_speed_3pm <= 13: Yes (3.0)
## |   |   |   |   |   |   |   |   |   wind_speed_3pm > 13
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = N: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = NNE: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = NE: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = ENE: No (0.0)
```

```
## |   |   |   |   |   |   |   |   |   wind_dir_9am = E: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = ESE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = SE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = SSE: No (1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = S: No (7.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = SSW
## |   |   |   |   |   |   |   |   |   |   cloud_3pm <= 5: No (7.0)
## |   |   |   |   |   |   |   |   |   |   cloud_3pm > 5
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_speed <= 44: No (4.0/1
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_speed > 44: Yes (4.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = SW
## |   |   |   |   |   |   |   |   |   |   humidity_3pm <= 69
## |   |   |   |   |   |   |   |   |   |   |   wind_speed_9am <= 17: No (5.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_speed_9am > 17: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   humidity_3pm > 69: Yes (3.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = WSW
## |   |   |   |   |   |   |   |   |   |   wind_speed_3pm <= 19: No (3.0)
## |   |   |   |   |   |   |   |   |   |   wind_speed_3pm > 19: Yes (2.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = W: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = WNW: No (2.0/1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = NW: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = NNW: Yes (2.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = SW
## |   |   |   |   |   |   |   |   |   wind_dir_9am = N
## |   |   |   |   |   |   |   |   |   humidity_9am <= 91: Yes (3.0)
## |   |   |   |   |   |   |   |   |   humidity_9am > 91: No (2.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = NNE: Yes (1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = NE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = ENE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = E: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = ESE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = SE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = SSE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = S: Yes (1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = SSW
## |   |   |   |   |   |   |   |   |   |   pressure_3pm <= 1021.1: No (4.0)
```

```
## |   |   |   |   |   |   |   |   |   pressure_3pm > 1021.1
## |   |   |   |   |   |   |   |   |   |   humidity_9am <= 72: No (2.0)
## |   |   |   |   |   |   |   |   |   |   humidity_9am > 72: Yes (4.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = SW
## |   |   |   |   |   |   |   |   |   pressure_3pm <= 1020.2: No (6.0)
## |   |   |   |   |   |   |   |   |   pressure_3pm > 1020.2: Yes (2.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = WSW: No (10.0/3.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = W: No (4.0/1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = WNW: No (2.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = NW: Yes (1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = NNW: No (2.0/1.0)
## |   |   |   |   |   |   |   wind_dir_3pm = WSW
## |   |   |   |   |   |   |   |   wind_dir_9am = N: No (2.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = NNE: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = NE: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = ENE: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = E: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = ESE: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = SE: No (1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = SSE: Yes (1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = S: Yes (2.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = SSW: No (1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = SW: No (6.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = WSW
## |   |   |   |   |   |   |   |   |   wind_gust_speed <= 44: No (4.0)
## |   |   |   |   |   |   |   |   |   wind_gust_speed > 44: Yes (10.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = W
## |   |   |   |   |   |   |   |   |   wind_gust_dir = N: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = NNE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = NE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = ENE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = E: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = ESE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = SE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = SSE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = S: No (0.0)
```

```
## |   |   |   |   |   |   |   |   |   wind_gust_dir = SSW: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = SW: No (6.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = WSW
## |   |   |   |   |   |   |   |   |   |   sunshine <= 2.7: No (2.0)
## |   |   |   |   |   |   |   |   |   |   sunshine > 2.7: Yes (3.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = W
## |   |   |   |   |   |   |   |   |   |   wind_speed_9am <= 13: Yes (3.0)
## |   |   |   |   |   |   |   |   |   |   wind_speed_9am > 13: No (2.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = WNW: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = NW: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = NNW: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = WNW
## |   |   |   |   |   |   |   |   |   temp_3pm <= 12.3: Yes (5.0/1.0)
## |   |   |   |   |   |   |   |   |   temp_3pm > 12.3: No (6.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = NW: No (1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = NNW: No (0.0)
## |   |   |   |   |   |   |   wind_dir_3pm = W
## |   |   |   |   |   |   |   |   cloud_3pm <= 7
## |   |   |   |   |   |   |   |   |   wind_gust_dir = N: Yes (1.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = NNE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = NE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = ENE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = E: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = ESE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = SE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = SSE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = S: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = SSW: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = SW: Yes (4.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = WSW
## |   |   |   |   |   |   |   |   |   |   temp_3pm <= 15.6: Yes (3.0/1.0)
## |   |   |   |   |   |   |   |   |   |   temp_3pm > 15.6: No (2.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = W: No (13.0/2.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = WNW: Yes (3.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = NW: No (2.0/1.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = NNW: No (0.0)
```

```
## |   |   |   |   |   |   |   |   cloud_3pm > 7: No (6.0)
## |   |   |   |   |   |   |   wind_dir_3pm = WNW: Yes (27.0/10.0)
## |   |   |   |   |   |   |   wind_dir_3pm = NW
## |   |   |   |   |   |   |   |   evaporation <= 0.9: No (2.0)
## |   |   |   |   |   |   |   |   evaporation > 0.9
## |   |   |   |   |   |   |   |   |   temp_9am <= 8.1: No (2.0)
## |   |   |   |   |   |   |   |   |   temp_9am > 8.1: Yes (10.0/1.0)
## |   |   |   |   |   |   |   wind_dir_3pm = NNW
## |   |   |   |   |   |   |   |   rain_today = No: Yes (2.0)
## |   |   |   |   |   |   |   |   rain_today = Yes: No (8.0/2.0)
## |   |   |   |   |   |   humidity_3pm > 73
## |   |   |   |   |   |   |   wind_dir_9am = N
## |   |   |   |   |   |   |   |   humidity_9am <= 97: Yes (18.0)
## |   |   |   |   |   |   |   |   humidity_9am > 97: No (4.0/1.0)
## |   |   |   |   |   |   |   wind_dir_9am = NNE
## |   |   |   |   |   |   |   |   wind_speed_3pm <= 15: No (2.0)
## |   |   |   |   |   |   |   |   wind_speed_3pm > 15: Yes (4.0)
## |   |   |   |   |   |   |   wind_dir_9am = NE: Yes (6.0)
## |   |   |   |   |   |   |   wind_dir_9am = ENE
## |   |   |   |   |   |   |   |   rain_today = No: No (2.0)
## |   |   |   |   |   |   |   |   rain_today = Yes
## |   |   |   |   |   |   |   |   |   temp_3pm <= 21.3: Yes (6.0)
## |   |   |   |   |   |   |   |   |   temp_3pm > 21.3: No (2.0)
## |   |   |   |   |   |   |   wind_dir_9am = E: Yes (11.0/1.0)
## |   |   |   |   |   |   |   wind_dir_9am = ESE
## |   |   |   |   |   |   |   |   humidity_3pm <= 75: Yes (5.0)
## |   |   |   |   |   |   |   |   humidity_3pm > 75: No (12.0/3.0)
## |   |   |   |   |   |   |   wind_dir_9am = SE
## |   |   |   |   |   |   |   |   min_temp <= 14.5: Yes (8.0)
## |   |   |   |   |   |   |   |   min_temp > 14.5
## |   |   |   |   |   |   |   |   |   wind_speed_9am <= 20: Yes (3.0)
## |   |   |   |   |   |   |   |   |   wind_speed_9am > 20: No (8.0)
## |   |   |   |   |   |   |   wind_dir_9am = SSE
## |   |   |   |   |   |   |   |   evaporation <= 4.6: No (5.0/1.0)
## |   |   |   |   |   |   |   |   evaporation > 4.6: Yes (16.0/2.0)
## |   |   |   |   |   |   |   wind_dir_9am = S
```

```
## |   |   |   |   |   |   |   |   rain_today = No: No (3.0)
## |   |   |   |   |   |   |   |   rain_today = Yes
## |   |   |   |   |   |   |   |   |   pressure_9am <= 1017.9: No (16.0/6.0)
## |   |   |   |   |   |   |   |   |   pressure_9am > 1017.9: Yes (16.0/2.0)
## |   |   |   |   |   |   |   wind_dir_9am = SSW
## |   |   |   |   |   |   |   |   wind_dir_3pm = N: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = NNE: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = NE: No (2.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = ENE: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = E: No (1.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = ESE: Yes (1.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = SE: No (2.0/1.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = SSE: No (4.0/1.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = S
## |   |   |   |   |   |   |   |   |   wind_gust_dir = N: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = NNE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = NE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = ENE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = E: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = ESE: Yes (1.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = SE: Yes (1.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = SSE: No (1.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = S
## |   |   |   |   |   |   |   |   |   |   cloud_3pm <= 5: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   cloud_3pm > 5: No (2.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = SSW
## |   |   |   |   |   |   |   |   |   |   pressure_3pm <= 1021.6: Yes (3.0)
## |   |   |   |   |   |   |   |   |   |   pressure_3pm > 1021.6: No (2.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = SW: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = WSW: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = W: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = WNW: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = NW: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = NNW: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = SSW
## |   |   |   |   |   |   |   |   |   sunshine <= 3.7: No (2.0)
```

```
## |   |   |   |   |   |   |   |   |   sunshine > 3.7: Yes (5.0/1.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = SW: No (3.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = WSW: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = W: Yes (1.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = WNW: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = NW: Yes (2.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = NNW: No (0.0)
## |   |   |   |   |   |   |   wind_dir_9am = SW
## |   |   |   |   |   |   |   |   wind_gust_dir = N: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NNE: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NE: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = ENE: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = E: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = ESE: Yes (1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SE: No (2.0/1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SSE: Yes (2.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = S
## |   |   |   |   |   |   |   |   |   min_temp <= 18.2: Yes (10.0/1.0)
## |   |   |   |   |   |   |   |   |   min_temp > 18.2: No (3.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SSW
## |   |   |   |   |   |   |   |   |   wind_speed_3pm <= 15: Yes (2.0)
## |   |   |   |   |   |   |   |   |   wind_speed_3pm > 15: No (6.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SW
## |   |   |   |   |   |   |   |   |   sunshine <= 6.5: Yes (8.0)
## |   |   |   |   |   |   |   |   |   sunshine > 6.5: No (4.0/1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = WSW: Yes (4.0/1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = W: No (3.0/1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = WNW: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NW: No (1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NNW: Yes (0.0)
## |   |   |   |   |   |   |   wind_dir_9am = WSW
## |   |   |   |   |   |   |   |   cloud_3pm <= 6
## |   |   |   |   |   |   |   |   |   max_temp <= 15: No (8.0/2.0)
## |   |   |   |   |   |   |   |   |   max_temp > 15: Yes (12.0/2.0)
## |   |   |   |   |   |   |   |   cloud_3pm > 6: Yes (14.0/2.0)
## |   |   |   |   |   |   |   wind_dir_9am = W
```

```
## |   |   |   |   |   |   |   |   cloud_3pm <= 7: Yes (26.0/3.0)
## |   |   |   |   |   |   |   |   cloud_3pm > 7: No (2.0)
## |   |   |   |   |   |   |   wind_dir_9am = WNW: Yes (27.0/7.0)
## |   |   |   |   |   |   |   wind_dir_9am = NW: Yes (30.0/9.0)
## |   |   |   |   |   |   |   wind_dir_9am = NNW
## |   |   |   |   |   |   |   |   rain_today = No: Yes (5.0)
## |   |   |   |   |   |   |   |   rain_today = Yes
## |   |   |   |   |   |   |   |   |   cloud_3pm <= 6: No (5.0)
## |   |   |   |   |   |   |   |   |   cloud_3pm > 6
## |   |   |   |   |   |   |   |   |   |   humidity_9am <= 83: No (4.0)
## |   |   |   |   |   |   |   |   |   |   humidity_9am > 83: Yes (10.0/1.0)
## |   |   |   |   |   wind_speed_3pm > 24
## |   |   |   |   |   |   sunshine <= 5.6
## |   |   |   |   |   |   |   wind_dir_3pm = N: Yes (8.0/1.0)
## |   |   |   |   |   |   |   wind_dir_3pm = NNE
## |   |   |   |   |   |   |   |   wind_speed_3pm <= 28: No (3.0)
## |   |   |   |   |   |   |   |   wind_speed_3pm > 28: Yes (6.0/1.0)
## |   |   |   |   |   |   |   wind_dir_3pm = NE
## |   |   |   |   |   |   |   |   humidity_9am <= 88: No (3.0)
## |   |   |   |   |   |   |   |   humidity_9am > 88: Yes (3.0)
## |   |   |   |   |   |   |   wind_dir_3pm = ENE
## |   |   |   |   |   |   |   |   humidity_3pm <= 74: No (4.0/1.0)
## |   |   |   |   |   |   |   |   humidity_3pm > 74: Yes (7.0)
## |   |   |   |   |   |   |   wind_dir_3pm = E
## |   |   |   |   |   |   |   |   wind_speed_3pm <= 28: Yes (6.0)
## |   |   |   |   |   |   |   |   wind_speed_3pm > 28
## |   |   |   |   |   |   |   |   |   wind_speed_9am <= 24: No (2.0)
## |   |   |   |   |   |   |   |   |   wind_speed_9am > 24: Yes (3.0)
## |   |   |   |   |   |   |   wind_dir_3pm = ESE
## |   |   |   |   |   |   |   |   evaporation <= 2.7: Yes (4.0/1.0)
## |   |   |   |   |   |   |   |   evaporation > 2.7: No (12.0/5.0)
## |   |   |   |   |   |   |   wind_dir_3pm = SE
## |   |   |   |   |   |   |   |   rain_today = No: No (6.0/1.0)
## |   |   |   |   |   |   |   |   rain_today = Yes
## |   |   |   |   |   |   |   |   |   humidity_3pm <= 69
## |   |   |   |   |   |   |   |   |   |   pressure_3pm <= 1015.8: Yes (5.0/1.0)
```

```
## |   |   |   |   |   |   |   |   |   pressure_3pm > 1015.8: No (2.0)
## |   |   |   |   |   |   |   |   humidity_3pm > 69: Yes (15.0)
## |   |   |   |   |   |   wind_dir_3pm = SSE: No (18.0/7.0)
## |   |   |   |   |   |   wind_dir_3pm = S
## |   |   |   |   |   |   |   wind_dir_9am = N: No (1.0)
## |   |   |   |   |   |   |   wind_dir_9am = NNE: Yes (0.0)
## |   |   |   |   |   |   |   wind_dir_9am = NE: Yes (0.0)
## |   |   |   |   |   |   |   wind_dir_9am = ENE: Yes (0.0)
## |   |   |   |   |   |   |   wind_dir_9am = E: Yes (0.0)
## |   |   |   |   |   |   |   wind_dir_9am = ESE: Yes (0.0)
## |   |   |   |   |   |   |   wind_dir_9am = SE: Yes (0.0)
## |   |   |   |   |   |   |   wind_dir_9am = SSE: Yes (0.0)
## |   |   |   |   |   |   |   wind_dir_9am = S
## |   |   |   |   |   |   |   |   pressure_3pm <= 1015.2
## |   |   |   |   |   |   |   |   |   wind_gust_speed <= 50: Yes (3.0/1.0)
## |   |   |   |   |   |   |   |   |   wind_gust_speed > 50: No (3.0)
## |   |   |   |   |   |   |   |   pressure_3pm > 1015.2: Yes (7.0)
## |   |   |   |   |   |   |   wind_dir_9am = SSW
## |   |   |   |   |   |   |   |   pressure_9am <= 1023.2: Yes (4.0)
## |   |   |   |   |   |   |   |   pressure_9am > 1023.2: No (3.0/1.0)
## |   |   |   |   |   |   |   wind_dir_9am = SW: No (2.0)
## |   |   |   |   |   |   |   wind_dir_9am = WSW: No (2.0/1.0)
## |   |   |   |   |   |   |   wind_dir_9am = W: Yes (0.0)
## |   |   |   |   |   |   |   wind_dir_9am = WNW: Yes (0.0)
## |   |   |   |   |   |   |   wind_dir_9am = NW: Yes (0.0)
## |   |   |   |   |   |   |   wind_dir_9am = NNW: Yes (0.0)
## |   |   |   |   |   |   wind_dir_3pm = SSW
## |   |   |   |   |   |   |   wind_speed_9am <= 13: No (4.0)
## |   |   |   |   |   |   |   wind_speed_9am > 13: Yes (23.0/7.0)
## |   |   |   |   |   |   wind_dir_3pm = SW
## |   |   |   |   |   |   |   wind_speed_9am <= 22: Yes (9.0/1.0)
## |   |   |   |   |   |   |   wind_speed_9am > 22: No (3.0)
## |   |   |   |   |   |   wind_dir_3pm = WSW
## |   |   |   |   |   |   |   cloud_3pm <= 6: Yes (9.0/1.0)
## |   |   |   |   |   |   |   cloud_3pm > 6
## |   |   |   |   |   |   |   |   cloud_3pm <= 7: No (9.0/2.0)
```

```
## |   |   |   |   |   |   |   |   cloud_3pm > 7: Yes (4.0/1.0)
## |   |   |   |   |   |   wind_dir_3pm = W
## |   |   |   |   |   |   |   wind_speed_3pm <= 28: Yes (12.0)
## |   |   |   |   |   |   |   wind_speed_3pm > 28
## |   |   |   |   |   |   |   |   cloud_9am <= 6: No (2.0)
## |   |   |   |   |   |   |   |   cloud_9am > 6
## |   |   |   |   |   |   |   |   |   humidity_3pm <= 70
## |   |   |   |   |   |   |   |   |   |   humidity_3pm <= 69: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   humidity_3pm > 69: No (3.0)
## |   |   |   |   |   |   |   |   |   humidity_3pm > 70: Yes (7.0)
## |   |   |   |   |   |   wind_dir_3pm = WNW: Yes (18.0/6.0)
## |   |   |   |   |   |   wind_dir_3pm = NW: Yes (10.0/1.0)
## |   |   |   |   |   |   wind_dir_3pm = NNW: No (2.0)
## |   |   |   |   |   sunshine > 5.6
## |   |   |   |   |   |   humidity_9am <= 87
## |   |   |   |   |   |   |   wind_gust_dir = N
## |   |   |   |   |   |   |   |   sunshine <= 9.1
## |   |   |   |   |   |   |   |   |   wind_dir_9am = N: Yes (2.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = NNE: Yes (1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = NE: Yes (1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = ENE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = E: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = ESE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = SE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = SSE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = S: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = SSW: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = SW: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = WSW: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = W: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = WNW: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = NW: No (2.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = NNW: No (1.0)
## |   |   |   |   |   |   |   |   sunshine > 9.1: No (3.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NNE
## |   |   |   |   |   |   |   |   pressure_9am <= 1017.4: Yes (13.0/2.0)
```

```
## |   |   |   |   |   |   |   |   pressure_9am > 1017.4: No (5.0)
## |   |   |   |   |   |   |   wind_gust_dir = NE
## |   |   |   |   |   |   |   |   wind_dir_3pm = N: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = NNE: No (6.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = NE: No (21.0/1.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = ENE: Yes (3.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = E: No (1.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = ESE: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = SE: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = SSE: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = S: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = SSW: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = SW: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = WSW: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = W: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = WNW: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = NW: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = NNW: No (0.0)
## |   |   |   |   |   |   |   wind_gust_dir = ENE
## |   |   |   |   |   |   |   |   rainfall <= 4.4: No (7.0/1.0)
## |   |   |   |   |   |   |   |   rainfall > 4.4: Yes (4.0)
## |   |   |   |   |   |   |   wind_gust_dir = E: No (28.0/7.0)
## |   |   |   |   |   |   |   wind_gust_dir = ESE: No (18.0/6.0)
## |   |   |   |   |   |   |   wind_gust_dir = SE
## |   |   |   |   |   |   |   |   wind_dir_3pm = N: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = NNE: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = NE: No (1.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = ENE: Yes (1.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = E: Yes (1.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = ESE: No (8.0/3.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = SE
## |   |   |   |   |   |   |   |   |   wind_dir_9am = N: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = NNE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = NE: Yes (1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = ENE: No (2.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = E: No (0.0)
```

```
## |   |   |   |   |   |   |   |   |   wind_dir_9am = ESE
## |   |   |   |   |   |   |   |   |   |   temp_9am <= 21.2: No (4.0)
## |   |   |   |   |   |   |   |   |   |   temp_9am > 21.2: Yes (2.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = SE
## |   |   |   |   |   |   |   |   |   |   cloud_9am <= 5: No (5.0)
## |   |   |   |   |   |   |   |   |   |   cloud_9am > 5
## |   |   |   |   |   |   |   |   |   |   |   humidity_9am <= 72: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   |   humidity_9am > 72: No (2.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = SSE
## |   |   |   |   |   |   |   |   |   |   pressure_9am <= 1015.8: Yes (5.0)
## |   |   |   |   |   |   |   |   |   |   pressure_9am > 1015.8
## |   |   |   |   |   |   |   |   |   |   |   wind_speed_9am <= 22: Yes (4.0/1
## |   |   |   |   |   |   |   |   |   |   |   wind_speed_9am > 22: No (3.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = S
## |   |   |   |   |   |   |   |   |   |   cloud_3pm <= 6: Yes (3.0)
## |   |   |   |   |   |   |   |   |   |   cloud_3pm > 6: No (2.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = SSW: No (2.0/1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = SW: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = WSW: Yes (1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = W: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = WNW: No (1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = NW: No (1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = NNW: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = SSE
## |   |   |   |   |   |   |   |   |   pressure_3pm <= 1018.5: No (8.0/2.0)
## |   |   |   |   |   |   |   |   |   pressure_3pm > 1018.5: Yes (4.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = S: Yes (3.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = SSW: Yes (1.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = SW: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = WSW: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = W: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = WNW: No (1.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = NW: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = NNW: No (0.0)
## |   |   |   |   |   |   |   wind_gust_dir = SSE
## |   |   |   |   |   |   |   |   cloud_3pm <= 5
```

```
## |   |   |   |   |   |   |   |   |   pressure_9am <= 1021.6: No (20.0/1.0)
## |   |   |   |   |   |   |   |   |   pressure_9am > 1021.6: Yes (7.0/1.0)
## |   |   |   |   |   |   |   |   cloud_3pm > 5
## |   |   |   |   |   |   |   |   |   wind_speed_3pm <= 28: No (4.0/1.0)
## |   |   |   |   |   |   |   |   |   wind_speed_3pm > 28: Yes (10.0)
## |   |   |   |   |   |   |   wind_gust_dir = S: No (53.0/14.0)
## |   |   |   |   |   |   |   wind_gust_dir = SSW
## |   |   |   |   |   |   |   |   evaporation <= 6.9
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = N: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = NNE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = NE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = ENE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = E: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = ESE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = SE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = SSE: Yes (4.0/1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = S
## |   |   |   |   |   |   |   |   |   |   temp_9am <= 20.5: No (11.0/1.0)
## |   |   |   |   |   |   |   |   |   |   temp_9am > 20.5: Yes (5.0/2.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = SSW
## |   |   |   |   |   |   |   |   |   |   wind_speed_9am <= 20
## |   |   |   |   |   |   |   |   |   |   |   cloud_9am <= 4: No (2.0)
## |   |   |   |   |   |   |   |   |   |   |   cloud_9am > 4: Yes (5.0/1.0)
## |   |   |   |   |   |   |   |   |   |   wind_speed_9am > 20: No (9.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = SW: No (4.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = WSW: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = W: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = WNW: Yes (1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = NW: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = NNW: No (0.0)
## |   |   |   |   |   |   |   |   evaporation > 6.9: Yes (4.0)
## |   |   |   |   |   |   |   wind_gust_dir = SW
## |   |   |   |   |   |   |   |   sunshine <= 8.3: Yes (4.0)
## |   |   |   |   |   |   |   |   sunshine > 8.3
## |   |   |   |   |   |   |   |   |   max_temp <= 23.8: No (8.0)
## |   |   |   |   |   |   |   |   |   max_temp > 23.8: Yes (2.0)
```

```
## |   |   |   |   |   |   |   wind_gust_dir = WSW
## |   |   |   |   |   |   |   |   max_temp <= 17.5: No (8.0)
## |   |   |   |   |   |   |   |   max_temp > 17.5
## |   |   |   |   |   |   |   |   |   max_temp <= 25.1: Yes (6.0)
## |   |   |   |   |   |   |   |   |   max_temp > 25.1: No (2.0)
## |   |   |   |   |   |   |   wind_gust_dir = W: No (29.0/8.0)
## |   |   |   |   |   |   |   wind_gust_dir = WNW
## |   |   |   |   |   |   |   |   wind_dir_3pm = N: Yes (1.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = NNE: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = NE: No (1.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = ENE: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = E: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = ESE: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = SE: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = SSE: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = S: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = SSW: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = SW: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = WSW: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = W
## |   |   |   |   |   |   |   |   |   pressure_3pm <= 1006: Yes (3.0)
## |   |   |   |   |   |   |   |   |   pressure_3pm > 1006: No (4.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = WNW
## |   |   |   |   |   |   |   |   |   cloud_9am <= 5: Yes (5.0/1.0)
## |   |   |   |   |   |   |   |   |   cloud_9am > 5: No (3.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = NW: No (4.0/1.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = NNW: No (0.0)
## |   |   |   |   |   |   |   wind_gust_dir = NW
## |   |   |   |   |   |   |   |   wind_gust_speed <= 52
## |   |   |   |   |   |   |   |   |   wind_speed_9am <= 17: Yes (3.0/1.0)
## |   |   |   |   |   |   |   |   |   wind_speed_9am > 17: No (7.0)
## |   |   |   |   |   |   |   |   wind_gust_speed > 52: Yes (2.0)
## |   |   |   |   |   |   |   wind_gust_dir = NNW
## |   |   |   |   |   |   |   |   cloud_9am <= 6: No (7.0/1.0)
## |   |   |   |   |   |   |   |   cloud_9am > 6: Yes (3.0)
## |   |   |   |   |   |   humidity_9am > 87
```

```
## |   |   |   |   |   |   |   wind_gust_speed <= 43: No (20.0/4.0)
## |   |   |   |   |   |   |   wind_gust_speed > 43
## |   |   |   |   |   |   |   |   wind_dir_3pm = N
## |   |   |   |   |   |   |   |   |   rainfall <= 5.3: Yes (5.0/1.0)
## |   |   |   |   |   |   |   |   |   rainfall > 5.3: No (2.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = NNE
## |   |   |   |   |   |   |   |   |   wind_speed_3pm <= 30: No (2.0)
## |   |   |   |   |   |   |   |   |   wind_speed_3pm > 30: Yes (3.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = NE
## |   |   |   |   |   |   |   |   |   cloud_3pm <= 4: Yes (3.0)
## |   |   |   |   |   |   |   |   |   cloud_3pm > 4: No (5.0/1.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = ENE
## |   |   |   |   |   |   |   |   |   rainfall <= 8.7: No (2.0)
## |   |   |   |   |   |   |   |   |   rainfall > 8.7: Yes (2.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = E: Yes (4.0/1.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = ESE
## |   |   |   |   |   |   |   |   |   humidity_3pm <= 75: No (4.0)
## |   |   |   |   |   |   |   |   |   humidity_3pm > 75: Yes (3.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = SE
## |   |   |   |   |   |   |   |   |   pressure_9am <= 1017.8: No (8.0/2.0)
## |   |   |   |   |   |   |   |   |   pressure_9am > 1017.8: Yes (7.0/1.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = SSE: Yes (13.0/4.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = S
## |   |   |   |   |   |   |   |   |   pressure_9am <= 1014.5: No (5.0)
## |   |   |   |   |   |   |   |   |   pressure_9am > 1014.5: Yes (13.0/3.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = SSW
## |   |   |   |   |   |   |   |   |   rain_today = No: No (3.0/1.0)
## |   |   |   |   |   |   |   |   |   rain_today = Yes: Yes (10.0/1.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = SW
## |   |   |   |   |   |   |   |   |   wind_speed_3pm <= 26: Yes (3.0)
## |   |   |   |   |   |   |   |   |   wind_speed_3pm > 26: No (5.0/1.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = WSW
## |   |   |   |   |   |   |   |   |   evaporation <= 3.5: Yes (3.0/1.0)
## |   |   |   |   |   |   |   |   |   evaporation > 3.5
## |   |   |   |   |   |   |   |   |   |   wind_gust_speed <= 52: No (8.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_speed > 52: Yes (3.0/1.0)
```

```
## |   |   |   |   |   |   |   |   wind_dir_3pm = W
## |   |   |   |   |   |   |   |   |   evaporation <= 4.2: No (3.0)
## |   |   |   |   |   |   |   |   |   evaporation > 4.2: Yes (24.0/7.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = WNW
## |   |   |   |   |   |   |   |   |   rain_today = No: Yes (5.0/1.0)
## |   |   |   |   |   |   |   |   |   rain_today = Yes: No (11.0/3.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = NW: Yes (9.0/1.0)
## |   |   |   |   |   |   |   |   wind_dir_3pm = NNW
## |   |   |   |   |   |   |   |   |   cloud_9am <= 7: Yes (9.0)
## |   |   |   |   |   |   |   |   |   cloud_9am > 7
## |   |   |   |   |   |   |   |   |   |   min_temp <= 5.8: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   min_temp > 5.8: No (3.0)
## |   |   |   wind_gust_speed > 54
## |   |   |   |   temp_9am <= 4.3: No (28.0/8.0)
## |   |   |   |   temp_9am > 4.3
## |   |   |   |   |   pressure_3pm <= 1006.9
## |   |   |   |   |   |   max_temp <= 19.3: Yes (286.0/21.0)
## |   |   |   |   |   |   max_temp > 19.3
## |   |   |   |   |   |   |   sunshine <= 8.3: Yes (112.0/10.0)
## |   |   |   |   |   |   |   sunshine > 8.3
## |   |   |   |   |   |   |   |   wind_speed_3pm <= 39: Yes (66.0/22.0)
## |   |   |   |   |   |   |   |   wind_speed_3pm > 39: No (11.0/1.0)
## |   |   |   |   |   pressure_3pm > 1006.9
## |   |   |   |   |   |   cloud_9am <= 4
## |   |   |   |   |   |   |   rain_today = No: No (20.0/3.0)
## |   |   |   |   |   |   |   rain_today = Yes
## |   |   |   |   |   |   |   |   wind_gust_dir = N
## |   |   |   |   |   |   |   |   |   humidity_9am <= 82: No (4.0/1.0)
## |   |   |   |   |   |   |   |   |   humidity_9am > 82: Yes (4.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NNE: No (1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NE: Yes (2.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = ENE: Yes (1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = E: Yes (1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = ESE: Yes (0.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SE: No (4.0/1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SSE: Yes (2.0)
```

```
## |   |   |   |   |   |   |   |   wind_gust_dir = S
## |   |   |   |   |   |   |   |   |   cloud_9am <= 3: No (6.0/1.0)
## |   |   |   |   |   |   |   |   |   cloud_9am > 3: Yes (3.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SSW
## |   |   |   |   |   |   |   |   |   wind_speed_3pm <= 26: No (4.0)
## |   |   |   |   |   |   |   |   |   wind_speed_3pm > 26: Yes (6.0/1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = SW: No (6.0/1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = WSW: Yes (13.0/1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = W: Yes (24.0/7.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = WNW
## |   |   |   |   |   |   |   |   |   pressure_3pm <= 1008.4: No (4.0)
## |   |   |   |   |   |   |   |   |   pressure_3pm > 1008.4: Yes (9.0/1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NW: Yes (7.0/1.0)
## |   |   |   |   |   |   |   |   wind_gust_dir = NNW: No (1.0)
## |   |   |   |   |   |   |   cloud_9am > 4
## |   |   |   |   |   |   |   |   wind_gust_speed <= 63
## |   |   |   |   |   |   |   |   wind_speed_9am <= 20: Yes (258.0/58.0)
## |   |   |   |   |   |   |   |   wind_speed_9am > 20
## |   |   |   |   |   |   |   |   |   wind_gust_dir = N
## |   |   |   |   |   |   |   |   |   |   pressure_9am <= 1015.5
## |   |   |   |   |   |   |   |   |   |   |   temp_9am <= 19.5: Yes (7.0)
## |   |   |   |   |   |   |   |   |   |   |   temp_9am > 19.5: No (2.0)
## |   |   |   |   |   |   |   |   |   |   pressure_9am > 1015.5: No (3.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = NNE: Yes (1.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = NE: No (2.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = ENE
## |   |   |   |   |   |   |   |   |   |   wind_speed_3pm <= 33: Yes (6.0)
## |   |   |   |   |   |   |   |   |   |   wind_speed_3pm > 33: No (2.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = E
## |   |   |   |   |   |   |   |   |   |   sunshine <= 8.2: Yes (7.0)
## |   |   |   |   |   |   |   |   |   |   sunshine > 8.2: No (7.0/1.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = ESE
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = N: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = NNE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = NE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = ENE: Yes (0.0)
```

```
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = E: No (2.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = ESE
## |   |   |   |   |   |   |   |   |   |   |   wind_speed_9am <= 30: No (4.0/1.0
## |   |   |   |   |   |   |   |   |   |   |   wind_speed_9am > 30: Yes (3.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = SE: Yes (3.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = SSE: No (1.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = S: Yes (1.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = SSW: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = SW: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = WSW: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = W: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = WNW: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = NW: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = NNW: Yes (0.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = SE: Yes (31.0/7.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = SSE
## |   |   |   |   |   |   |   |   |   |   rain_today = No: No (2.0)
## |   |   |   |   |   |   |   |   |   |   rain_today = Yes: Yes (30.0/9.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = S
## |   |   |   |   |   |   |   |   |   |   temp_9am <= 21.7: Yes (53.0/19.0)
## |   |   |   |   |   |   |   |   |   |   temp_9am > 21.7: No (5.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = SSW
## |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = N: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = NNE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = NE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = ENE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = E: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = ESE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = SE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = SSE: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = S
## |   |   |   |   |   |   |   |   |   |   |   humidity_3pm <= 73: No (9.0)
## |   |   |   |   |   |   |   |   |   |   |   humidity_3pm > 73: Yes (5.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = SSW
## |   |   |   |   |   |   |   |   |   |   |   humidity_3pm <= 75: Yes (11.0/3.0
## |   |   |   |   |   |   |   |   |   |   |   humidity_3pm > 75
```

```
## |   |   |   |   |   |   |   |   |   |   |   |   |   temp_9am <= 12.2: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   |   |   |   temp_9am > 12.2: No (6.0/1.0
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = SW: No (2.0/1.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = WSW: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = W: Yes (1.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = WNW: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = NW: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = NNW: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = SW: Yes (25.0/6.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = WSW
## |   |   |   |   |   |   |   |   |   |   |   rain_today = No: Yes (4.0)
## |   |   |   |   |   |   |   |   |   |   |   rain_today = Yes
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = N: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = NNE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = NE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = ENE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = E: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = ESE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = SE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = SSE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = S: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = SSW: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = SW: No (2.0/1.0)
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = WSW
## |   |   |   |   |   |   |   |   |   |   |   |   |   cloud_3pm <= 7: Yes (11.0/1.0
## |   |   |   |   |   |   |   |   |   |   |   |   |   cloud_3pm > 7: No (4.0)
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = W
## |   |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_speed <= 61
## |   |   |   |   |   |   |   |   |   |   |   |   |   |   cloud_9am <= 6
## |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   humidity_3pm <= 71: Y
## |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   humidity_3pm > 71: No
## |   |   |   |   |   |   |   |   |   |   |   |   |   |   cloud_9am > 6: Yes (5.0)
## |   |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_speed > 61: No (3.0
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = WNW: No (1.0)
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = NW: Yes (1.0)
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = NNW: Yes (0.0)
```

```
## |   |   |   |   |   |   |   |   |   wind_gust_dir = W
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = N: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = NNE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = NE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = ENE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = E: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = ESE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = SE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = SSE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = S: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = SSW: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = SW: No (1.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = WSW: Yes (6.0/1.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = W
## |   |   |   |   |   |   |   |   |   |   |   temp_3pm <= 15.2: Yes (19.0/5.0)
## |   |   |   |   |   |   |   |   |   |   |   temp_3pm > 15.2: No (4.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = WNW
## |   |   |   |   |   |   |   |   |   |   |   temp_3pm <= 14.4: Yes (8.0)
## |   |   |   |   |   |   |   |   |   |   |   temp_3pm > 14.4: No (3.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = NW
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = N: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = NNE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = NE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = ENE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = E: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = ESE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = SE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = SSE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = S: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = SSW: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = SW: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = WSW: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = W: No (2.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = WNW: Yes (3.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = NW: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = NNW: Yes (0.0)
```

```
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = NNW: Yes (1.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = WNW
## |   |   |   |   |   |   |   |   |   |   min_temp <= 7.3: No (5.0)
## |   |   |   |   |   |   |   |   |   |   min_temp > 7.3: Yes (19.0/3.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = NW: No (9.0/3.0)
## |   |   |   |   |   |   |   |   |   wind_gust_dir = NNW
## |   |   |   |   |   |   |   |   |   |   wind_speed_9am <= 24
## |   |   |   |   |   |   |   |   |   |   |   rain_today = No: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   |   rain_today = Yes: No (2.0)
## |   |   |   |   |   |   |   |   |   |   wind_speed_9am > 24: Yes (4.0)
## |   |   |   |   |   |   |   wind_gust_speed > 63: Yes (376.0/72.0)
## |   humidity_3pm > 81
## |   |   temp_9am <= 6.4
## |   |   |   pressure_3pm <= 1014: Yes (135.0/17.0)
## |   |   |   pressure_3pm > 1014
## |   |   |   |   humidity_3pm <= 94
## |   |   |   |   |   wind_dir_9am = N
## |   |   |   |   |   |   pressure_3pm <= 1025.1
## |   |   |   |   |   |   |   wind_gust_speed <= 15: No (9.0)
## |   |   |   |   |   |   |   wind_gust_speed > 15
## |   |   |   |   |   |   |   |   cloud_9am <= 6: No (34.0/15.0)
## |   |   |   |   |   |   |   |   cloud_9am > 6: Yes (12.0)
## |   |   |   |   |   |   pressure_3pm > 1025.1: No (16.0)
## |   |   |   |   |   wind_dir_9am = NNE
## |   |   |   |   |   |   humidity_9am <= 92: Yes (3.0/1.0)
## |   |   |   |   |   |   humidity_9am > 92: No (9.0)
## |   |   |   |   |   wind_dir_9am = NE
## |   |   |   |   |   |   wind_gust_speed <= 22: No (6.0)
## |   |   |   |   |   |   wind_gust_speed > 22: Yes (8.0/1.0)
## |   |   |   |   |   wind_dir_9am = ENE
## |   |   |   |   |   |   humidity_9am <= 94: Yes (4.0)
## |   |   |   |   |   |   humidity_9am > 94: No (15.0/2.0)
## |   |   |   |   |   wind_dir_9am = E
## |   |   |   |   |   |   rain_today = No: No (21.0/1.0)
## |   |   |   |   |   |   rain_today = Yes
## |   |   |   |   |   |   |   wind_speed_9am <= 9: Yes (2.0)
```

```
## |   |   |   |   |   |   |   wind_speed_9am > 9: No (3.0)
## |   |   |   |   |   wind_dir_9am = ESE: No (25.0/3.0)
## |   |   |   |   |   wind_dir_9am = SE: No (8.0/1.0)
## |   |   |   |   |   wind_dir_9am = SSE: No (14.0/3.0)
## |   |   |   |   |   wind_dir_9am = S: No (11.0/1.0)
## |   |   |   |   |   wind_dir_9am = SSW: No (21.0/3.0)
## |   |   |   |   |   wind_dir_9am = SW: No (27.0/7.0)
## |   |   |   |   |   wind_dir_9am = WSW
## |   |   |   |   |   |   pressure_3pm <= 1018: No (28.0/5.0)
## |   |   |   |   |   |   pressure_3pm > 1018: Yes (4.0)
## |   |   |   |   |   wind_dir_9am = W: No (38.0/11.0)
## |   |   |   |   |   wind_dir_9am = WNW
## |   |   |   |   |   |   wind_speed_3pm <= 24
## |   |   |   |   |   |   |   temp_3pm <= 1.9: Yes (6.0/2.0)
## |   |   |   |   |   |   |   temp_3pm > 1.9: No (5.0)
## |   |   |   |   |   |   wind_speed_3pm > 24: Yes (2.0)
## |   |   |   |   |   wind_dir_9am = NW
## |   |   |   |   |   |   pressure_3pm <= 1016: No (5.0/1.0)
## |   |   |   |   |   |   pressure_3pm > 1016: Yes (2.0)
## |   |   |   |   |   wind_dir_9am = NNW
## |   |   |   |   |   |   wind_gust_speed <= 28: No (10.0)
## |   |   |   |   |   |   wind_gust_speed > 28
## |   |   |   |   |   |   |   max_temp <= 12.2: No (2.0)
## |   |   |   |   |   |   |   max_temp > 12.2: Yes (6.0)
## |   |   |   |   humidity_3pm > 94
## |   |   |   |   |   temp_3pm <= -1.2: Yes (78.0/13.0)
## |   |   |   |   |   temp_3pm > -1.2
## |   |   |   |   |   |   wind_dir_9am = N
## |   |   |   |   |   |   |   wind_gust_dir = N
## |   |   |   |   |   |   |   |   pressure_3pm <= 1017.8: Yes (2.0)
## |   |   |   |   |   |   |   |   pressure_3pm > 1017.8: No (3.0)
## |   |   |   |   |   |   |   wind_gust_dir = NNE: No (4.0/1.0)
## |   |   |   |   |   |   |   wind_gust_dir = NE: Yes (1.0)
## |   |   |   |   |   |   |   wind_gust_dir = ENE: Yes (0.0)
## |   |   |   |   |   |   |   wind_gust_dir = E: Yes (0.0)
## |   |   |   |   |   |   |   wind_gust_dir = ESE: Yes (0.0)
```

```
## |   |   |   |   |   |   |   wind_gust_dir = SE: Yes (0.0)
## |   |   |   |   |   |   |   wind_gust_dir = SSE: Yes (1.0)
## |   |   |   |   |   |   |   wind_gust_dir = S: No (4.0)
## |   |   |   |   |   |   |   wind_gust_dir = SSW: Yes (2.0)
## |   |   |   |   |   |   |   wind_gust_dir = SW: Yes (0.0)
## |   |   |   |   |   |   |   wind_gust_dir = WSW: Yes (0.0)
## |   |   |   |   |   |   |   wind_gust_dir = W
## |   |   |   |   |   |   |   |   temp_3pm <= -0.5: No (4.0)
## |   |   |   |   |   |   |   |   temp_3pm > -0.5
## |   |   |   |   |   |   |   |   |   humidity_9am <= 98: Yes (6.0)
## |   |   |   |   |   |   |   |   |   humidity_9am > 98: No (2.0)
## |   |   |   |   |   |   |   wind_gust_dir = WNW: Yes (4.0/1.0)
## |   |   |   |   |   |   |   wind_gust_dir = NW: Yes (3.0/1.0)
## |   |   |   |   |   |   |   wind_gust_dir = NNW: Yes (1.0)
## |   |   |   |   |   |   wind_dir_9am = NNE
## |   |   |   |   |   |   |   pressure_9am <= 1023.3: Yes (8.0/1.0)
## |   |   |   |   |   |   |   pressure_9am > 1023.3: No (4.0)
## |   |   |   |   |   |   wind_dir_9am = NE
## |   |   |   |   |   |   |   wind_gust_speed <= 46: No (6.0)
## |   |   |   |   |   |   |   wind_gust_speed > 46: Yes (3.0)
## |   |   |   |   |   |   wind_dir_9am = ENE
## |   |   |   |   |   |   |   humidity_3pm <= 98
## |   |   |   |   |   |   |   |   humidity_3pm <= 95: No (2.0)
## |   |   |   |   |   |   |   |   humidity_3pm > 95
## |   |   |   |   |   |   |   |   |   temp_9am <= 3.8
## |   |   |   |   |   |   |   |   |   |   temp_3pm <= 5.3: Yes (3.0/1.0)
## |   |   |   |   |   |   |   |   |   |   temp_3pm > 5.3: No (2.0)
## |   |   |   |   |   |   |   |   |   temp_9am > 3.8: Yes (8.0)
## |   |   |   |   |   |   |   humidity_3pm > 98: No (4.0)
## |   |   |   |   |   |   wind_dir_9am = E
## |   |   |   |   |   |   |   evaporation <= 3.2: Yes (2.0)
## |   |   |   |   |   |   |   evaporation > 3.2
## |   |   |   |   |   |   |   |   wind_gust_speed <= 57
## |   |   |   |   |   |   |   |   |   humidity_3pm <= 98
## |   |   |   |   |   |   |   |   |   |   humidity_3pm <= 96: No (4.0)
## |   |   |   |   |   |   |   |   |   |   humidity_3pm > 96
```

```
## |   |   |   |   |   |   |   |   |   |   rain_today = No
## |   |   |   |   |   |   |   |   |   |   |   humidity_9am <= 97: Yes (3.0/1.0
## |   |   |   |   |   |   |   |   |   |   |   humidity_9am > 97: No (2.0)
## |   |   |   |   |   |   |   |   |   |   rain_today = Yes: Yes (5.0/1.0)
## |   |   |   |   |   |   |   |   |   humidity_3pm > 98: No (5.0)
## |   |   |   |   |   |   |   |   wind_gust_speed > 57: Yes (5.0)
## |   |   |   |   |   |   wind_dir_9am = ESE
## |   |   |   |   |   |   |   temp_3pm <= 1.7: Yes (7.0/1.0)
## |   |   |   |   |   |   |   temp_3pm > 1.7: No (8.0/3.0)
## |   |   |   |   |   |   wind_dir_9am = SE: No (8.0)
## |   |   |   |   |   |   wind_dir_9am = SSE: No (3.0)
## |   |   |   |   |   |   wind_dir_9am = S
## |   |   |   |   |   |   |   cloud_3pm <= 6: No (2.0)
## |   |   |   |   |   |   |   cloud_3pm > 6: Yes (3.0)
## |   |   |   |   |   |   wind_dir_9am = SSW
## |   |   |   |   |   |   |   rainfall <= 9: No (4.0)
## |   |   |   |   |   |   |   rainfall > 9: Yes (3.0)
## |   |   |   |   |   |   wind_dir_9am = SW: No (15.0/5.0)
## |   |   |   |   |   |   wind_dir_9am = WSW
## |   |   |   |   |   |   |   wind_dir_3pm = N: No (0.0)
## |   |   |   |   |   |   |   wind_dir_3pm = NNE: No (0.0)
## |   |   |   |   |   |   |   wind_dir_3pm = NE: No (0.0)
## |   |   |   |   |   |   |   wind_dir_3pm = ENE: No (0.0)
## |   |   |   |   |   |   |   wind_dir_3pm = E: No (0.0)
## |   |   |   |   |   |   |   wind_dir_3pm = ESE: No (0.0)
## |   |   |   |   |   |   |   wind_dir_3pm = SE: No (0.0)
## |   |   |   |   |   |   |   wind_dir_3pm = SSE: Yes (2.0)
## |   |   |   |   |   |   |   wind_dir_3pm = S: No (0.0)
## |   |   |   |   |   |   |   wind_dir_3pm = SSW: Yes (1.0)
## |   |   |   |   |   |   |   wind_dir_3pm = SW: No (4.0)
## |   |   |   |   |   |   |   wind_dir_3pm = WSW
## |   |   |   |   |   |   |   |   min_temp <= 1.7: No (16.0/3.0)
## |   |   |   |   |   |   |   |   min_temp > 1.7: Yes (3.0)
## |   |   |   |   |   |   |   wind_dir_3pm = W
## |   |   |   |   |   |   |   |   humidity_9am <= 94: No (2.0)
## |   |   |   |   |   |   |   |   humidity_9am > 94: Yes (9.0/2.0)
```

```
## |   |   |   |   |   |   |   wind_dir_3pm = WNW: No (2.0/1.0)
## |   |   |   |   |   |   |   wind_dir_3pm = NW: No (0.0)
## |   |   |   |   |   |   |   wind_dir_3pm = NNW: No (0.0)
## |   |   |   |   |   |   wind_dir_9am = W
## |   |   |   |   |   |   |   wind_gust_dir = N: Yes (0.0)
## |   |   |   |   |   |   |   wind_gust_dir = NNE: No (1.0)
## |   |   |   |   |   |   |   wind_gust_dir = NE: Yes (0.0)
## |   |   |   |   |   |   |   wind_gust_dir = ENE: Yes (0.0)
## |   |   |   |   |   |   |   wind_gust_dir = E: Yes (0.0)
## |   |   |   |   |   |   |   wind_gust_dir = ESE: Yes (0.0)
## |   |   |   |   |   |   |   wind_gust_dir = SE: Yes (0.0)
## |   |   |   |   |   |   |   wind_gust_dir = SSE: No (1.0)
## |   |   |   |   |   |   |   wind_gust_dir = S: Yes (0.0)
## |   |   |   |   |   |   |   wind_gust_dir = SSW: Yes (0.0)
## |   |   |   |   |   |   |   wind_gust_dir = SW: No (1.0)
## |   |   |   |   |   |   |   wind_gust_dir = WSW
## |   |   |   |   |   |   |   |   rainfall <= 0.1: No (4.0/1.0)
## |   |   |   |   |   |   |   |   rainfall > 0.1
## |   |   |   |   |   |   |   |   |   max_temp <= 0.9: No (3.0/1.0)
## |   |   |   |   |   |   |   |   |   max_temp > 0.9: Yes (12.0/1.0)
## |   |   |   |   |   |   |   wind_gust_dir = W
## |   |   |   |   |   |   |   |   wind_gust_speed <= 70
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = N: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = NNE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = NE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = ENE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = E: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = ESE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = SE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = SSE: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = S: Yes (1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = SSW: Yes (1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = SW: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = WSW: No (1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = W
## |   |   |   |   |   |   |   |   |   |   humidity_9am <= 97: Yes (5.0/1.0)
```

```
## |   |   |   |   |   |   |   |   |   humidity_9am > 97: No (7.0/1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = WNW: No (3.0/1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = NW: No (0.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = NNW: No (0.0)
## |   |   |   |   |   |   |   |   wind_gust_speed > 70: Yes (7.0)
## |   |   |   |   |   |   |   wind_gust_dir = WNW
## |   |   |   |   |   |   |   |   humidity_3pm <= 97: Yes (2.0)
## |   |   |   |   |   |   |   |   humidity_3pm > 97: No (3.0/1.0)
## |   |   |   |   |   |   |   wind_gust_dir = NW
## |   |   |   |   |   |   |   |   temp_3pm <= 4.7: Yes (3.0)
## |   |   |   |   |   |   |   |   temp_3pm > 4.7: No (2.0)
## |   |   |   |   |   |   |   wind_gust_dir = NNW: Yes (0.0)
## |   |   |   |   |   |   wind_dir_9am = WNW: Yes (31.0/7.0)
## |   |   |   |   |   |   wind_dir_9am = NW
## |   |   |   |   |   |   |   wind_speed_3pm <= 24: Yes (17.0)
## |   |   |   |   |   |   |   wind_speed_3pm > 24
## |   |   |   |   |   |   |   |   temp_9am <= 3.6: No (2.0)
## |   |   |   |   |   |   |   |   temp_9am > 3.6: Yes (2.0)
## |   |   |   |   |   |   wind_dir_9am = NNW
## |   |   |   |   |   |   |   wind_gust_speed <= 41: No (2.0)
## |   |   |   |   |   |   |   wind_gust_speed > 41: Yes (8.0)
## |   |   temp_9am > 6.4
## |   |   |   rainfall <= 2.3
## |   |   |   |   humidity_3pm <= 88
## |   |   |   |   |   pressure_3pm <= 1015.1
## |   |   |   |   |   |   cloud_3pm <= 5
## |   |   |   |   |   |   |   temp_3pm <= 17.7: Yes (187.0/45.0)
## |   |   |   |   |   |   |   temp_3pm > 17.7
## |   |   |   |   |   |   |   |   rain_today = No
## |   |   |   |   |   |   |   |   |   cloud_3pm <= 2
## |   |   |   |   |   |   |   |   |   |   rainfall <= 0.1: No (15.0)
## |   |   |   |   |   |   |   |   |   |   rainfall > 0.1: Yes (2.0)
## |   |   |   |   |   |   |   |   |   cloud_3pm > 2
## |   |   |   |   |   |   |   |   |   |   sunshine <= 8.3
## |   |   |   |   |   |   |   |   |   |   |   rainfall <= 0.7: Yes (15.0)
## |   |   |   |   |   |   |   |   |   |   |   rainfall > 0.7
```

```
## |   |   |   |   |   |   |   |   |   |   |   evaporation <= 3.5: Yes (3.0)
## |   |   |   |   |   |   |   |   |   |   |   evaporation > 3.5: No (3.0)
## |   |   |   |   |   |   |   |   |   |   sunshine > 8.3
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = N
## |   |   |   |   |   |   |   |   |   |   |   |   humidity_3pm <= 84: No (9.0)
## |   |   |   |   |   |   |   |   |   |   |   |   humidity_3pm > 84: Yes (5.0/1.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = NNE
## |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_speed <= 33: No (12.0/1
## |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_speed > 33: Yes (18.0/0
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = NE
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = N: Yes (3.0/1.0)
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = NNE: No (8.0)
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = NE
## |   |   |   |   |   |   |   |   |   |   |   |   |   wind_speed_9am <= 15: No (5.0
## |   |   |   |   |   |   |   |   |   |   |   |   |   wind_speed_9am > 15
## |   |   |   |   |   |   |   |   |   |   |   |   |   |   pressure_3pm <= 1013.2: Y
## |   |   |   |   |   |   |   |   |   |   |   |   |   |   pressure_3pm > 1013.2: No
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = ENE: No (3.0)
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = E: No (0.0)
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = ESE: No (0.0)
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = SE: No (1.0)
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = SSE: Yes (1.0)
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = S: No (0.0)
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = SSW: No (0.0)
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = SW: No (2.0/1.0)
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = WSW: No (0.0)
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = W: No (0.0)
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = WNW: No (0.0)
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = NW: No (0.0)
## |   |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = NNW: No (0.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = ENE
## |   |   |   |   |   |   |   |   |   |   |   |   temp_9am <= 23.7: No (4.0)
## |   |   |   |   |   |   |   |   |   |   |   |   temp_9am > 23.7: Yes (4.0/1.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = E
## |   |   |   |   |   |   |   |   |   |   |   |   cloud_9am <= 5: Yes (3.0/1.0)
## |   |   |   |   |   |   |   |   |   |   |   |   cloud_9am > 5: No (2.0)
```

```
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = ESE: Yes (1.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = SE
## |   |   |   |   |   |   |   |   |   |   |   humidity_9am <= 96: Yes (12.0)
## |   |   |   |   |   |   |   |   |   |   |   humidity_9am > 96: No (2.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = SSE: Yes (15.0/6.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = S
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = N: No (2.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = NNE: No (3.0/1.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = NE: No (2.0/1.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = ENE: No (1.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = E: Yes (1.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = ESE: Yes (1.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = SE: Yes (3.0/1.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = SSE: No (2.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = S: Yes (13.0/4.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = SSW: No (4.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = SW: No (0.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = WSW: No (0.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = W: No (0.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = WNW: Yes (1.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = NW: No (0.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = NNW: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = SSW: No (22.0/9.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = SW: Yes (4.0/1.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = WSW
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_speed <= 39: No (2.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_speed > 39: Yes (3.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = W
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_speed <= 37: Yes (3.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_speed > 37: No (10.0/2
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = WNW: No (9.0/4.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = NW
## |   |   |   |   |   |   |   |   |   |   |   temp_3pm <= 19.1: No (3.0)
## |   |   |   |   |   |   |   |   |   |   |   temp_3pm > 19.1: Yes (4.0/1.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = NNW: No (3.0/1.0)
## |   |   |   |   |   |   |   |   rain_today = Yes: Yes (33.0/8.0)
```

```
## |   |   |   |   |   |   cloud_3pm > 5: Yes (695.0/141.0)
## |   |   |   |   |   pressure_3pm > 1015.1
## |   |   |   |   |   |   humidity_9am <= 96
## |   |   |   |   |   |   |   wind_speed_9am <= 31
## |   |   |   |   |   |   |   |   rain_today = No
## |   |   |   |   |   |   |   |   |   cloud_3pm <= 7
## |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = N: Yes (20.0/8.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = NNE
## |   |   |   |   |   |   |   |   |   |   |   wind_speed_9am <= 7: No (7.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_speed_9am > 7: Yes (24.0/9.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = NE: No (22.0/6.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = ENE
## |   |   |   |   |   |   |   |   |   |   |   rainfall <= 0.1
## |   |   |   |   |   |   |   |   |   |   |   |   sunshine <= 6.3: Yes (3.0)
## |   |   |   |   |   |   |   |   |   |   |   |   sunshine > 6.3: No (6.0/1.0)
## |   |   |   |   |   |   |   |   |   |   |   rainfall > 0.1: No (7.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = E
## |   |   |   |   |   |   |   |   |   |   |   pressure_3pm <= 1019.8
## |   |   |   |   |   |   |   |   |   |   |   |   evaporation <= 6.1: Yes (11.0)
## |   |   |   |   |   |   |   |   |   |   |   |   evaporation > 6.1: No (2.0)
## |   |   |   |   |   |   |   |   |   |   |   pressure_3pm > 1019.8: No (6.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = ESE
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = N: Yes (1.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = NNE: No (0.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = NE: Yes (1.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = ENE: Yes (1.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = E: No (1.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = ESE: No (6.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = SE: Yes (1.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = SSE: No (2.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = S: Yes (1.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = SSW: No (0.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = SW: Yes (1.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = WSW: No (0.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = W: No (1.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = WNW: Yes (2.0)
```

```
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = NW: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = NNW: Yes (1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = SE
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = N
## |   |   |   |   |   |   |   |   |   |   |   rainfall <= 0.2
## |   |   |   |   |   |   |   |   |   |   |   |   evaporation <= 3.8: Yes (2.0
## |   |   |   |   |   |   |   |   |   |   |   |   evaporation > 3.8
## |   |   |   |   |   |   |   |   |   |   |   |   |   humidity_9am <= 72
## |   |   |   |   |   |   |   |   |   |   |   |   |   |   temp_3pm <= 16.4: No
## |   |   |   |   |   |   |   |   |   |   |   |   |   |   temp_3pm > 16.4: Yes
## |   |   |   |   |   |   |   |   |   |   |   |   |   humidity_9am > 72: No (7
## |   |   |   |   |   |   |   |   |   |   |   rainfall > 0.2: Yes (3.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = NNE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = NE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = ENE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = E: Yes (3.0/1.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = ESE: No (2.0/1.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = SE
## |   |   |   |   |   |   |   |   |   |   |   humidity_3pm <= 86: No (5.0)
## |   |   |   |   |   |   |   |   |   |   |   humidity_3pm > 86: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = SSE
## |   |   |   |   |   |   |   |   |   |   |   humidity_3pm <= 85: No (3.0)
## |   |   |   |   |   |   |   |   |   |   |   humidity_3pm > 85: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = S: Yes (3.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = SSW: Yes (8.0/1.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = SW: Yes (5.0/1.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = WSW: Yes (1.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = W: No (3.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = WNW: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = NW: No (2.0/1.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = NNW: No (1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = SSE: No (30.0/11.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = S
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = N: Yes (3.0/1.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = NNE: No (1.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = NE: No (0.0)
```

```
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = ENE: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = E: No (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = ESE: No (1.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = SE: No (3.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = SSE
## |   |   |   |   |   |   |   |   |   |   |   pressure_3pm <= 1021.6: No (3.0)
## |   |   |   |   |   |   |   |   |   |   |   pressure_3pm > 1021.6: Yes (3.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = S: No (17.0/4.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = SSW
## |   |   |   |   |   |   |   |   |   |   |   evaporation <= 3.5: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   |   evaporation > 3.5
## |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = N: No (0.0)
## |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = NNE: No (0.0
## |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = NE: No (0.0)
## |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = ENE: No (0.0
## |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = E: No (0.0)
## |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = ESE: No (0.0
## |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = SE: No (0.0)
## |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = SSE: No (0.0
## |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = S
## |   |   |   |   |   |   |   |   |   |   |   |   |   rainfall <= 0.1: Yes (2.0
## |   |   |   |   |   |   |   |   |   |   |   |   |   rainfall > 0.1: No (4.0/1
## |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = SSW: No (3.0
## |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = SW: No (0.0)
## |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = WSW: No (0.0
## |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = W: No (0.0)
## |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = WNW: No (0.0
## |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = NW: No (0.0)
## |   |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = NNW: No (0.0
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = SW
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_speed <= 33: No (4.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_speed > 33: Yes (4.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = WSW
## |   |   |   |   |   |   |   |   |   |   |   cloud_3pm <= 6: No (3.0/1.0)
## |   |   |   |   |   |   |   |   |   |   |   cloud_3pm > 6: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_9am = W
```

```
## |   |   |   |   |   |   |   |   |   |   |   |   cloud_3pm <= 6: Yes (3.0)
## |   |   |   |   |   |   |   |   |   |   |   |   cloud_3pm > 6: No (2.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = WNW
## |   |   |   |   |   |   |   |   |   |   |   |   temp_3pm <= 13.2: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   |   |   temp_3pm > 13.2: No (2.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = NW: No (2.0/1.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_dir_9am = NNW: Yes (1.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = SSW
## |   |   |   |   |   |   |   |   |   |   |   temp_3pm <= 14: No (13.0)
## |   |   |   |   |   |   |   |   |   |   |   temp_3pm > 14: Yes (36.0/14.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = SW
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = N: No (2.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = NNE: No (0.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = NE: No (0.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = ENE: Yes (1.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = E: No (0.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = ESE: No (0.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = SE: Yes (1.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = SSE: Yes (2.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = S: Yes (3.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = SSW
## |   |   |   |   |   |   |   |   |   |   |   |   humidity_9am <= 80: Yes (3.0)
## |   |   |   |   |   |   |   |   |   |   |   |   humidity_9am > 80: No (2.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = SW
## |   |   |   |   |   |   |   |   |   |   |   |   temp_9am <= 11: Yes (3.0)
## |   |   |   |   |   |   |   |   |   |   |   |   temp_9am > 11: No (5.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = WSW
## |   |   |   |   |   |   |   |   |   |   |   |   pressure_9am <= 1016.8: Yes (3.0
## |   |   |   |   |   |   |   |   |   |   |   |   pressure_9am > 1016.8: No (8.0/1
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = W
## |   |   |   |   |   |   |   |   |   |   |   |   sunshine <= 4.5: No (3.0)
## |   |   |   |   |   |   |   |   |   |   |   |   sunshine > 4.5: Yes (3.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = WNW: No (0.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = NW: No (0.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_gust_dir = NNW: No (1.0)
## |   |   |   |   |   |   |   |   |   |   wind_dir_3pm = WSW: Yes (32.0/15.0)
```

```
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = W
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = N: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = NNE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = NE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = ENE: No (1.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = E: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = ESE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = SE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = SSE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = S: Yes (1.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = SSW: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = SW: Yes (1.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = WSW
## |   |   |   |   |   |   |   |   |   |   |   wind_speed_9am <= 9: Yes (3.0)
## |   |   |   |   |   |   |   |   |   |   |   wind_speed_9am > 9: No (4.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = W
## |   |   |   |   |   |   |   |   |   |   |   humidity_3pm <= 82: No (3.0)
## |   |   |   |   |   |   |   |   |   |   |   humidity_3pm > 82: Yes (6.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = WNW: Yes (4.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = NW: No (2.0/1.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = NNW: No (1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = WNW: Yes (30.0/10.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = NW
## |   |   |   |   |   |   |   |   |   |   max_temp <= 17.3: No (5.0)
## |   |   |   |   |   |   |   |   |   |   max_temp > 17.3: Yes (12.0/2.0)
## |   |   |   |   |   |   |   |   |   wind_dir_3pm = NNW
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = N: No (3.0/1.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = NNE: Yes (1.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = NE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = ENE: Yes (1.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = E: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = ESE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = SE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = SSE: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = S: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = SSW: Yes (0.0)
```

```
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = SW: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = WSW: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = W: Yes (0.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = WNW: Yes (1.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = NW: Yes (3.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_dir = NNW
## |   |   |   |   |   |   |   |   |   |   |   rainfall <= 0.1
## |   |   |   |   |   |   |   |   |   |   |   |   evaporation <= 4.5: No (4.0/1
## |   |   |   |   |   |   |   |   |   |   |   |   evaporation > 4.5: Yes (4.0)
## |   |   |   |   |   |   |   |   |   |   |   rainfall > 0.1: No (5.0)
## |   |   |   |   |   |   |   |   |   cloud_3pm > 7
## |   |   |   |   |   |   |   |   |   |   rainfall <= 0.1
## |   |   |   |   |   |   |   |   |   |   |   humidity_9am <= 81: Yes (99.0/28.0)
## |   |   |   |   |   |   |   |   |   |   |   humidity_9am > 81: No (65.0/25.0)
## |   |   |   |   |   |   |   |   |   |   rainfall > 0.1: Yes (90.0/27.0)
## |   |   |   |   |   |   |   |   rain_today = Yes
## |   |   |   |   |   |   |   |   |   wind_dir_9am = N
## |   |   |   |   |   |   |   |   |   |   evaporation <= 1.4: No (2.0)
## |   |   |   |   |   |   |   |   |   |   evaporation > 1.4: Yes (11.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = NNE: Yes (1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = NE: Yes (3.0/1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = ENE
## |   |   |   |   |   |   |   |   |   |   wind_speed_3pm <= 17: Yes (4.0)
## |   |   |   |   |   |   |   |   |   |   wind_speed_3pm > 17: No (2.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = E
## |   |   |   |   |   |   |   |   |   |   rainfall <= 1.8: Yes (3.0)
## |   |   |   |   |   |   |   |   |   |   rainfall > 1.8: No (4.0/1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = ESE
## |   |   |   |   |   |   |   |   |   |   pressure_9am <= 1024.7: No (2.0)
## |   |   |   |   |   |   |   |   |   |   pressure_9am > 1024.7: Yes (2.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = SE: Yes (8.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = SSE: No (6.0/2.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = S: No (9.0/1.0)
## |   |   |   |   |   |   |   |   |   wind_dir_9am = SSW
## |   |   |   |   |   |   |   |   |   |   cloud_9am <= 6
## |   |   |   |   |   |   |   |   |   |   |   sunshine <= 8.2: No (3.0)
```

```
## |   |   |   |   |   |   |   |   |   |   sunshine > 8.2: Yes (4.0/1.0)
## |   |   |   |   |   |   |   |   |   cloud_9am > 6: Yes (5.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = SW: Yes (14.0/6.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = WSW
## |   |   |   |   |   |   |   |   |   humidity_9am <= 84
## |   |   |   |   |   |   |   |   |   |   humidity_9am <= 74: No (3.0/1.0)
## |   |   |   |   |   |   |   |   |   |   humidity_9am > 74: Yes (5.0)
## |   |   |   |   |   |   |   |   |   humidity_9am > 84: No (4.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = W: Yes (18.0/2.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = WNW
## |   |   |   |   |   |   |   |   |   evaporation <= 4.6: No (4.0/1.0)
## |   |   |   |   |   |   |   |   |   evaporation > 4.6: Yes (4.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = NW: Yes (13.0/4.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = NNW
## |   |   |   |   |   |   |   |   |   cloud_3pm <= 7
## |   |   |   |   |   |   |   |   |   |   rainfall <= 1.7: No (3.0)
## |   |   |   |   |   |   |   |   |   |   rainfall > 1.7: Yes (2.0)
## |   |   |   |   |   |   |   |   |   cloud_3pm > 7: Yes (6.0)
## |   |   |   |   |   |   |   wind_speed_9am > 31: Yes (31.0/5.0)
## |   |   |   |   |   |   humidity_9am > 96
## |   |   |   |   |   |   |   wind_dir_3pm = N
## |   |   |   |   |   |   |   |   temp_3pm <= 16.1: No (12.0)
## |   |   |   |   |   |   |   |   temp_3pm > 16.1: Yes (2.0)
## |   |   |   |   |   |   |   wind_dir_3pm = NNE: No (11.0/3.0)
## |   |   |   |   |   |   |   wind_dir_3pm = NE: No (7.0)
## |   |   |   |   |   |   |   wind_dir_3pm = ENE: Yes (4.0/1.0)
## |   |   |   |   |   |   |   wind_dir_3pm = E: No (8.0)
## |   |   |   |   |   |   |   wind_dir_3pm = ESE: No (2.0/1.0)
## |   |   |   |   |   |   |   wind_dir_3pm = SE: No (25.0/10.0)
## |   |   |   |   |   |   |   wind_dir_3pm = SSE
## |   |   |   |   |   |   |   |   pressure_3pm <= 1018.6: Yes (2.0)
## |   |   |   |   |   |   |   |   pressure_3pm > 1018.6: No (6.0)
## |   |   |   |   |   |   |   wind_dir_3pm = S: No (18.0/3.0)
## |   |   |   |   |   |   |   wind_dir_3pm = SSW: No (21.0/4.0)
## |   |   |   |   |   |   |   wind_dir_3pm = SW
## |   |   |   |   |   |   |   |   cloud_3pm <= 6
```

```
## |   |   |   |   |   |   |   |   humidity_3pm <= 85: Yes (2.0)
## |   |   |   |   |   |   |   |   humidity_3pm > 85: No (3.0)
## |   |   |   |   |   |   |   |   cloud_3pm > 6: No (5.0)
## |   |   |   |   |   |   |   wind_dir_3pm = WSW
## |   |   |   |   |   |   |   |   pressure_3pm <= 1024: Yes (6.0/1.0)
## |   |   |   |   |   |   |   |   pressure_3pm > 1024: No (8.0)
## |   |   |   |   |   |   |   wind_dir_3pm = W: Yes (14.0/5.0)
## |   |   |   |   |   |   |   wind_dir_3pm = WNW: Yes (7.0/1.0)
## |   |   |   |   |   |   |   wind_dir_3pm = NW
## |   |   |   |   |   |   |   |   wind_dir_9am = N
## |   |   |   |   |   |   |   |   |   pressure_3pm <= 1019.9: Yes (2.0)
## |   |   |   |   |   |   |   |   |   pressure_3pm > 1019.9: No (2.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = NNE: No (1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = NE: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = ENE: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = E: Yes (1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = ESE: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = SE: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = SSE: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = S: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = SSW: No (1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = SW: No (0.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = WSW: Yes (1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = W: No (1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = WNW: Yes (1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = NW: No (4.0/1.0)
## |   |   |   |   |   |   |   |   wind_dir_9am = NNW: Yes (2.0)
## |   |   |   |   |   |   |   wind_dir_3pm = NNW
## |   |   |   |   |   |   |   |   pressure_3pm <= 1016.4
## |   |   |   |   |   |   |   |   |   rain_today = No
## |   |   |   |   |   |   |   |   |   |   wind_gust_speed <= 31: No (2.0)
## |   |   |   |   |   |   |   |   |   |   wind_gust_speed > 31: Yes (4.0)
## |   |   |   |   |   |   |   |   |   rain_today = Yes: Yes (2.0)
## |   |   |   |   |   |   |   |   pressure_3pm > 1016.4: No (13.0/1.0)
## |   |   |   |   humidity_3pm > 88
## |   |   |   |   |   temp_3pm <= 27.8: Yes (2128.0/410.0)
```

```
## |   |   |   |   |   temp_3pm > 27.8: No (25.0/2.0)
## |   |   |   rainfall > 2.3: Yes (4073.0/504.0)
##
## Number of Leaves  :  3780
##
## Size of the tree :   5039
```

References

Hornik, Kurt. 2020. *RWeka: R/Weka Interface*. https://CRAN.R-project.org/package=RWeka.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.67 Weka Decision Tree Performance

Here we plot the performance of the decision tree, showing a risk chart. The areas under the recall and risk curves are also reported.

```
predicted <- predict(model, ds[te, vars], type="prob")[,2]
riskchart(predicted, actual_te, risk_te)
```

An error matrix shows, clockwise from the top left, the percentages of true negatives, false positives, true positives, and false negatives.

```
predicted <- predict(model, ds[te, vars], type="class")
sum(actual_te != predicted)/length(predicted) # Overall error rate
```

```
## [1] 0.1561875
```

```
round(100*table(actual_te, predicted, dnn=c("Actual", "Predicted"))/length(predicte
```

```
##       Predicted
## Actual No Yes
##    No  75   5
##    Yes 11  10
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.68 Weka Decision Tree Plot Using Party

% <<dtrees:plot_party} % plot(as.party(model)) % @

% We can also display a textual version using

% <<dtrees:print_party, out.lines=12} % print(as.party(model)) % @

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.69 The Original C5.0 Implementation

The (Kuhn and Quinlan 2021) package interfaces the original C code of the C5.0 implementation by Ross Quinlan, the developer of the decision tree induction algorithm.

```
library(C50)
model <- C5.0(form, ds[tr, vars])
```

```
model
```

```
##
## Call:
## C5.0.formula(formula = form, data = ds[tr, vars])
##
## Classification Tree
## Number of samples: 123722
## Number of predictors: 20
##
## Tree size: 845
##
## Non-standard options: attempt to group attributes
```

```
C5imp(model)
```

```
##                 Overall
## humidity_3pm     100.00
## sunshine          90.80
## wind_gust_speed   89.91
## rain_today        88.29
## pressure_3pm      36.63
## pressure_9am      25.09
## cloud_3pm         19.31
## rainfall          18.42
## temp_9am          14.99
## wind_dir_3pm      12.89
## temp_3pm          10.27
## wind_speed_9am    10.27
## wind_speed_3pm     9.98
## humidity_9am       9.74
## cloud_9am          8.49
## wind_dir_9am       7.55
## wind_gust_dir      7.17
## min_temp           6.07
## evaporation        4.22
## max_temp           2.14
```

% DONT EVAL YET - SEEMS TO BE TAKING LONG TIME

```
plot(model)
```

I am not aware of any converter from a C5.0 tree to an rpart tree and so fancyRpartPlot() will not be useful here.

References

Kuhn, Max, and Ross Quinlan. 2021. *C50: C5.0 Decision Trees and Rule-Based Models*. https://topepo.github.io/C5.0/.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.70 C5.0 Summary

```
summary(model)
```

```
##
## Call:
## C5.0.formula(formula = form, data = ds[tr, vars])
##
##
## C5.0 [Release 2.07 GPL Edition]      Thu Jun 24 16:06:34 2021
## -------------------------------
##
## Class specified by attribute `outcome'
##
## Read 123722 cases (21 attributes) from undefined.data
##
## Decision tree:
##
## humidity_3pm <= 66:
## :...rain_today = Yes:
## :   :...sunshine > 9:
## :   :   :...pressure_9am > 1014: No (1111/54)
## :   :   :   pressure_9am <= 1014:
## :   :   :   :...cloud_3pm > 6:
## :   :   :       :...wind_speed_3pm > 13:
## :   :   :       :   :...wind_gust_speed <= 70: No (60/16)
## :   :   :       :   :   wind_gust_speed > 70: Yes (4)
## :   :   :       :   wind_speed_3pm <= 13:
## :   :   :       :   :...evaporation > 6.8: Yes (6)
## :   :   :       :       evaporation <= 6.8:
## :   :   :       :       :...wind_dir_3pm in [N-E]: Yes (4/1)
## :   :   :       :           wind_dir_3pm in [ESE-NNW]: No (5)
## :   :   :       cloud_3pm <= 6:
## :   :   :       :...sunshine > 11:
## :   :   :           :...wind_speed_3pm <= 20: No (123/3)
## :   :   :           :   wind_speed_3pm > 20:
## :   :   :           :   :...wind_dir_3pm in [ESE-NNW]: No (119/10)
## :   :   :           :       wind_dir_3pm in [N-E]:
## :   :   :           :       :...temp_9am <= 24.8: Yes (5)
```

```
## :   :   :           :           temp_9am > 24.8:
## :   :   :           :           :...cloud_3pm <= 4: No (7/1)
## :   :   :           :               cloud_3pm > 4: Yes (3)
## :   :   :           sunshine <= 11:
## :   :   :           :...wind_dir_9am in [SSE-NNW]: No (343/59)
## :   :   :               wind_dir_9am in [N-SE]:
## :   :   :               :...wind_dir_9am in [N-NE]: No (64/13)
## :   :   :                   wind_dir_9am in [ENE-SE]:
## :   :   :                   :...cloud_9am <= 4:
## :   :   :                       :...temp_9am <= 28.5: No (17/6)
## :   :   :                       :   temp_9am > 28.5: Yes (13/1)
## :   :   :                       cloud_9am > 4:
## :   :   :                       :...evaporation > 8.8: Yes (7/2)
## :   :   :                           evaporation <= 8.8:
## :   :   :                           :...wind_gust_speed <= 87: No (35/4)
## :   :   :                               wind_gust_speed > 87: Yes (2)
## :   :   sunshine <= 9:
## :   :   :...wind_gust_speed <= 46:
## :   :       :...humidity_3pm <= 50:
## :   :       :   :...sunshine > 6.4: No (1908/172)
## :   :       :   :   sunshine <= 6.4:
## :   :       :   :   :...wind_gust_speed <= 37:
## :   :       :   :       :...pressure_3pm > 1012.5: No (98/6)
## :   :       :   :       :   pressure_3pm <= 1012.5:
## :   :       :   :       :   :...pressure_9am <= 1012.4: No (20/2)
## :   :       :   :       :       pressure_9am > 1012.4:
## :   :       :   :       :       :...min_temp > 18.1: No (4)
## :   :       :   :       :           min_temp <= 18.1:
## :   :       :   :       :           :...wind_dir_9am in [N-WNW]: Yes (8)
## :   :       :   :       :               wind_dir_9am in [NW-NNW]: No (2)
## :   :       :   :       wind_gust_speed > 37:
## :   :       :   :       :...wind_dir_9am in [N-E]:
## :   :       :   :           :...cloud_3pm > 4: Yes (39/10)
## :   :       :   :           :   cloud_3pm <= 4:
## :   :       :   :           :   :...wind_gust_speed <= 44: No (12/2)
## :   :       :   :           :       wind_gust_speed > 44: Yes (2)
```

```
## :   :       :   :           wind_dir_9am in [ESE-NNW]:
## :   :       :   :           :...humidity_9am <= 78:
## :   :       :   :               :...wind_speed_3pm <= 28: No (58/6)
## :   :       :   :               :   wind_speed_3pm > 28:
## :   :       :   :               :   :...cloud_9am <= 5: Yes (3)
## :   :       :   :               :       cloud_9am > 5: No (2)
## :   :       :   :               humidity_9am > 78:
## :   :       :   :               :...humidity_3pm <= 41: No (9)
## :   :       :   :                   humidity_3pm > 41:
## :   :       :   :                   :...sunshine <= 4.2: Yes (8/1)
## :   :       :   :                       sunshine > 4.2: No (16/6)
## :   :       :   humidity_3pm > 50:
## :   :       :   :...wind_gust_speed <= 31:
## :   :       :       :...pressure_3pm > 1016:
## :   :       :       :   :...min_temp <= 18.9: No (991/110)
## :   :       :       :   :   min_temp > 18.9:
## :   :       :       :   :   :...pressure_9am <= 1021.1: No (16/1)
## :   :       :       :   :       pressure_9am > 1021.1:
## :   :       :       :   :       :...cloud_9am <= 6: No (4/1)
## :   :       :       :   :           cloud_9am > 6: Yes (11/1)
## :   :       :       :   pressure_3pm <= 1016:
## :   :       :       :   :...wind_dir_9am in [N-NNE]:
## :   :       :       :       :...sunshine <= 5.2:
## :   :       :       :       :   :...sunshine <= 3.4: No (12/5)
## :   :       :       :       :   :   sunshine > 3.4: Yes (11)
## :   :       :       :       :   sunshine > 5.2:
## :   :       :       :       :   :...cloud_9am <= 7: No (108/30)
## :   :       :       :       :       cloud_9am > 7: Yes (10/3)
## :   :       :       :       wind_dir_9am in [NE-NNW]:
## :   :       :       :       :...humidity_3pm <= 61:
## :   :       :       :           :...cloud_3pm <= 6: No (296/41)
## :   :       :       :           :   cloud_3pm > 6:
## :   :       :       :           :   :...sunshine > 8: No (20/1)
## :   :       :       :           :       sunshine <= 8:
## :   :       :       :           :       :...min_temp <= 12.7: Yes (7)
## :   :       :       :           :           min_temp > 12.7: No (32/11)
```

```
## :   :       :       :           humidity_3pm > 61:
## :   :       :       :           :...cloud_3pm <= 4:
## :   :       :       :               :...sunshine <= 5.8: Yes (6)
## :   :       :       :               :   sunshine > 5.8: No (30/12)
## :   :       :       :               cloud_3pm > 4:
## :   :       :       :               :...wind_gust_dir in [N-NW]: No (193/49)
## :   :       :       :                   wind_gust_dir = NNW:
## :   :       :       :                   :...sunshine <= 6.9: No (2)
## :   :       :       :                       sunshine > 6.9: Yes (8/1)
## :   :       :       wind_gust_speed > 31:
## :   :       :       :...pressure_3pm > 1014.3:
## :   :       :           :...rainfall > 6.5:
## :   :       :           :   :...min_temp <= 3: No (25/2)
## :   :       :           :   :   min_temp > 3:
## :   :       :           :   :   :...humidity_9am <= 82:
## :   :       :           :   :       :...wind_speed_9am > 31:
## :   :       :           :   :       :   :...temp_9am <= 11.7: Yes (9)
## :   :       :           :   :       :   :   temp_9am > 11.7:
## :   :       :           :   :       :   :   :...sunshine <= 5.9: Yes (5)
## :   :       :           :   :       :   :       sunshine > 5.9: No (11/3)
## :   :       :           :   :       :   wind_speed_9am <= 31:
## :   :       :           :   :       :   :...wind_dir_3pm in [WNW-NNW]:
## :   :       :           :   :       :       :...cloud_9am <= 2: No (6)
## :   :       :           :   :       :       :   cloud_9am > 2: Yes (23/7)
## :   :       :           :   :       :       wind_dir_3pm in [N-W]:
## :   :       :           :   :       :       :...max_temp > 14: No (420/103)
## :   :       :           :   :       :           max_temp <= 14:
## :   :       :           :   :       :           :...temp_3pm <= 17.4: No (32/11)
## :   :       :           :   :       :               temp_3pm > 17.4: Yes (5)
## :   :       :           :   :       humidity_9am > 82:
## :   :       :           :   :       :...cloud_9am <= 7: No (303/125)
## :   :       :           :   :           cloud_9am > 7:
## :   :       :           :   :           :...humidity_3pm > 64: Yes (14/1)
## :   :       :           :   :               humidity_3pm <= 64:
## :   :       :           :   :               :...cloud_3pm <= 4: No (11/2)
## :   :       :           :   :                   cloud_3pm > 4: [S1]
```

```
## :   :       :           :   rainfall <= 6.5:
## :   :       :           :   :...wind_speed_9am > 15: No (649/100)
## :   :       :           :       wind_speed_9am <= 15:
## :   :       :           :       :...cloud_3pm <= 4: No (141/19)
## :   :       :           :           cloud_3pm > 4:
## :   :       :           :           :...pressure_3pm > 1021.7:
## :   :       :           :               :...humidity_9am <= 68: No (42/1)
## :   :       :           :               :   humidity_9am > 68:
## :   :       :           :               :   :...max_temp > 21.3:
## :   :       :           :               :       :...cloud_3pm > 6: Yes (8)
## :   :       :           :               :       :   cloud_3pm <= 6: [S2]
## :   :       :           :               :       max_temp <= 21.3:
## :   :       :           :               :       :...rainfall <= 4.7: No (182/27)
## :   :       :           :               :           rainfall > 4.7: [S3]
## :   :       :           :               pressure_3pm <= 1021.7:
## :   :       :           :               :...humidity_3pm > 59:
## :   :       :           :                   :...evaporation <= 4.3: Yes (74/32)
## :   :       :           :                   :   evaporation > 4.3: [S4]
## :   :       :           :                   humidity_3pm <= 59:
## :   :       :           :                   :...humidity_3pm > 51: No (259/53)
## :   :       :           :                       humidity_3pm <= 51:
## :   :       :           :                       :...cloud_9am <= 1: No (17/1)
## :   :       :           :                           cloud_9am > 1:
## :   :       :           :                           :...wind_gust_speed > 41: [S5
## :   :       :           :                               wind_gust_speed <= 41: [S
## :   :       :           pressure_3pm <= 1014.3:
## :   :       :           :...pressure_3pm <= 1000:
## :   :       :               :...rainfall > 27.6: No (4)
## :   :       :               :   rainfall <= 27.6:
## :   :       :               :   :...humidity_3pm <= 54: Yes (17/1)
## :   :       :               :       humidity_3pm > 54:
## :   :       :               :       :...min_temp <= 6.2: No (3)
## :   :       :               :           min_temp > 6.2:
## :   :       :               :           :...cloud_9am > 5: Yes (9)
## :   :       :               :               cloud_9am <= 5:
## :   :       :               :               :...min_temp <= 13.3: Yes (7/1)
```

```
## :   :       :               :                   min_temp > 13.3: No (3)
## :   :       :               pressure_3pm > 1000:
## :   :       :               :...humidity_9am <= 86:
## :   :       :                   :...wind_dir_9am in [N-NNE]:
## :   :       :                   :   :...wind_gust_speed <= 39:
## :   :       :                   :   :   :...wind_gust_speed <= 35: No (18/3)
## :   :       :                   :   :   :   wind_gust_speed > 35:
## :   :       :                   :   :   :   :...cloud_9am <= 3: Yes (5)
## :   :       :                   :   :   :       cloud_9am > 3: No (32/13)
## :   :       :                   :   :   wind_gust_speed > 39:
## :   :       :                   :   :   :...pressure_3pm <= 1009.3: Yes (16)
## :   :       :                   :   :       pressure_3pm > 1009.3:
## :   :       :                   :   :       :...rainfall > 6.8: No (4)
## :   :       :                   :   :           rainfall <= 6.8: [S7]
## :   :       :                   :   wind_dir_9am in [NE-NNW]:
## :   :       :                   :   :...wind_speed_9am > 26:
## :   :       :                   :       :...wind_gust_speed <= 39: Yes (25/6)
## :   :       :                   :       :   wind_gust_speed > 39: [S8]
## :   :       :                   :       wind_speed_9am <= 26:
## :   :       :                   :       :...wind_gust_speed <= 44: No (659/206)
## :   :       :                   :           wind_gust_speed > 44:
## :   :       :                   :           :...pressure_3pm > 1014: Yes (6)
## :   :       :                   :               pressure_3pm <= 1014: [S9]
## :   :       :                   humidity_9am > 86:
## :   :       :                   :...humidity_3pm > 60: Yes (168/62)
## :   :       :                       humidity_3pm <= 60:
## :   :       :                       :...humidity_3pm <= 52: Yes (72/29)
## :   :       :                           humidity_3pm > 52:
## :   :       :                           :...wind_speed_3pm <= 11: No (18/1)
## :   :       :                               wind_speed_3pm > 11:
## :   :       :                               :...wind_dir_3pm in [N-ENE]: [S10]
## :   :       :                                   wind_dir_3pm in [E-NNW]:
## :   :       :                                   :...sunshine > 8.1: No (72/17)
## :   :       :                                       sunshine <= 8.1: [S11]
## :   :       wind_gust_speed > 46:
## :   :       :...humidity_3pm <= 48:
```

```
## :   :           :...wind_dir_3pm in [N-ESE]:
## :   :           :   :...humidity_9am <= 49: No (12)
## :   :           :   :   humidity_9am > 49:
## :   :           :   :   :...wind_gust_speed > 72: Yes (10/1)
## :   :           :   :       wind_gust_speed <= 72:
## :   :           :   :       :...pressure_3pm <= 1002.6: Yes (9)
## :   :           :   :           pressure_3pm > 1002.6:
## :   :           :   :           :...wind_dir_9am in [SSW-NNW]: No (17)
## :   :           :   :               wind_dir_9am in [N-S]:
## :   :           :   :               :...wind_speed_3pm <= 22: Yes (49/20)
## :   :           :   :                   wind_speed_3pm > 22: No (45/12)
## :   :           :   wind_dir_3pm in [SE-NNW]:
## :   :           :   :...sunshine <= 6.7:
## :   :           :       :...cloud_9am <= 3: No (20/2)
## :   :           :       :   cloud_9am > 3:
## :   :           :       :   :...humidity_3pm <= 43:
## :   :           :       :       :...pressure_9am <= 996.6: Yes (6)
## :   :           :       :       :   pressure_9am > 996.6: No (73/23)
## :   :           :       :       humidity_3pm > 43:
## :   :           :       :       :...wind_speed_9am <= 37: Yes (70/24)
## :   :           :       :           wind_speed_9am > 37: No (4)
## :   :           :       sunshine > 6.7:
## :   :           :       :...wind_dir_3pm = NNW:
## :   :           :           :...cloud_3pm <= 1: Yes (5)
## :   :           :           :   cloud_3pm > 1: No (44/17)
## :   :           :           wind_dir_3pm in [SE-NW]:
## :   :           :           :...pressure_9am > 1017.8: No (90/4)
## :   :           :               pressure_9am <= 1017.8:
## :   :           :               :...humidity_3pm <= 45: No (618/104)
## :   :           :                   humidity_3pm > 45:
## :   :           :                   :...wind_speed_9am > 15: No (99/25)
## :   :           :                       wind_speed_9am <= 15:
## :   :           :                       :...humidity_9am > 80: No (13/2)
## :   :           :                           humidity_9am <= 80:
## :   :           :                           :...humidity_9am > 71: Yes (10)
## :   :           :                               humidity_9am <= 71: [S12]
```

```
## :   :           humidity_3pm > 48:
## :   :           :...pressure_3pm <= 1012.5:
## :   :               :...evaporation <= 0.9:
## :   :               :   :...humidity_9am <= 85: No (24/3)
## :   :               :   :   humidity_9am > 85:
## :   :               :   :   :...humidity_9am <= 93: Yes (7)
## :   :               :   :       humidity_9am > 93: No (4/1)
## :   :               :   evaporation > 0.9:
## :   :               :   :...wind_gust_speed > 59:
## :   :               :       :...wind_speed_9am > 28:
## :   :               :       :   :...humidity_9am <= 77: Yes (117/56)
## :   :               :       :   :   humidity_9am > 77:
## :   :               :       :   :   :...humidity_9am > 90: Yes (16)
## :   :               :       :   :       humidity_9am <= 90:
## :   :               :       :   :       :...wind_gust_speed > 65: Yes (44/6)
## :   :               :       :   :           wind_gust_speed <= 65:
## :   :               :       :   :           :...cloud_9am <= 7: No (6)
## :   :               :       :   :               cloud_9am > 7: Yes (2)
## :   :               :       :   wind_speed_9am <= 28:
## :   :               :       :   :...wind_dir_3pm in [N-NNE]: Yes (25)
## :   :               :       :       wind_dir_3pm in [NE-NNW]:
## :   :               :       :       :...cloud_3pm <= 3:
## :   :               :       :           :...cloud_9am <= 2: No (5)
## :   :               :       :           :   cloud_9am > 2: [S13]
## :   :               :       :           cloud_3pm > 3:
## :   :               :       :           :...humidity_3pm > 52: Yes (245/42)
## :   :               :       :               humidity_3pm <= 52:
## :   :               :       :               :...humidity_3pm > 49: Yes (55/18)
## :   :               :       :                   humidity_3pm <= 49:
## :   :               :       :                   :...max_temp <= 15.2: Yes (3)
## :   :               :       :                       max_temp > 15.2: No (8/1)
## :   :               :       wind_gust_speed <= 59:
## :   :               :       :...wind_speed_9am > 26:
## :   :               :           :...humidity_9am <= 76: No (55/11)
## :   :               :           :   humidity_9am > 76:
## :   :               :           :   :...wind_gust_dir in [N-SE]: Yes (12/2)
```

```
## :   :                :           :       wind_gust_dir in [SSE-NNW]:
## :   :                :           :       :...temp_3pm > 17: [S14]
## :   :                :           :           temp_3pm <= 17:
## :   :                :           :           :...wind_gust_speed <= 56: Yes (9)
## :   :                :           :               wind_gust_speed > 56: [S15]
## :   :                :           wind_speed_9am <= 26:
## :   :                :           :...wind_gust_speed <= 52:
## :   :                :               :...wind_speed_9am <= 19: Yes (220/84)
## :   :                :               :   wind_speed_9am > 19: No (110/51)
## :   :                :               wind_gust_speed > 52:
## :   :                :               :...wind_speed_3pm > 26:
## :   :                :                   :...temp_9am <= 17.9: Yes (90/30)
## :   :                :                   :   temp_9am > 17.9: [S16]
## :   :                :                   wind_speed_3pm <= 26:
## :   :                :                   :...cloud_9am > 7: Yes (26/3)
## :   :                :                       cloud_9am <= 7: [S17]
## :   :                pressure_3pm > 1012.5:
## :   :                :...humidity_9am <= 68:
## :   :                    :...cloud_3pm <= 2: No (29/2)
## :   :                    :   cloud_3pm > 2:
## :   :                    :   :...wind_gust_dir in [N-W]: No (407/106)
## :   :                    :       wind_gust_dir in [WNW-NNW]:
## :   :                    :       :...pressure_3pm <= 1016.6: Yes (14/3)
## :   :                    :           pressure_3pm > 1016.6: No (7/1)
## :   :                    humidity_9am > 68:
## :   :                    :...cloud_9am <= 3: No (84/22)
## :   :                        cloud_9am > 3:
## :   :                        :...temp_9am <= 6.3:
## :   :                            :...wind_dir_3pm in [N-ESE]: Yes (3)
## :   :                            :   wind_dir_3pm in [SE-NNW]:
## :   :                            :   :...rainfall <= 8.6: No (42/2)
## :   :                            :       rainfall > 8.6:
## :   :                            :       :...rainfall <= 12.2: Yes (6/1)
## :   :                            :           rainfall > 12.2: No (5)
## :   :                            temp_9am > 6.3:
## :   :                            :...rainfall > 20.2:
```

```
## :   :                                :...evaporation <= 3: Yes (16)
## :   :                                :   evaporation > 3:
## :   :                                :   :...evaporation <= 4.1: No (7/1)
## :   :                                :       evaporation > 4.1:
## :   :                                :       :...wind_speed_9am <= 26: Yes (57/12
## :   :                                :           wind_speed_9am > 26: [S18]
## :   :                                rainfall <= 20.2:
## :   :                                :...wind_dir_3pm in [NW-NNW]: Yes (54/17)
## :   :                                    wind_dir_3pm in [N-WNW]:
## :   :                                    :...wind_dir_9am in [N-ENE]: [S19]
## :   :                                        wind_dir_9am in [E-NNW]:
## :   :                                        :...cloud_3pm > 6: [S20]
## :   :                                            cloud_3pm <= 6:
## :   :                                            :...sunshine > 7:
## :   :                                                :...cloud_3pm <= 4: [S21]
## :   :                                                :   cloud_3pm > 4: [S22]
## :   :                                                sunshine <= 7: [S23]
## :   rain_today = No:
## :   :...sunshine <= 7.7:
## :       :...pressure_3pm <= 1013.7:
## :       :   :...wind_gust_speed <= 54:
## :       :   :   :...humidity_3pm <= 27: No (285/24)
## :       :   :   :   humidity_3pm > 27:
## :       :   :   :   :...wind_dir_9am in [N-ENE]:
## :       :   :   :       :...wind_speed_9am <= 0: No (83/18)
## :       :   :   :       :   wind_speed_9am > 0:
## :       :   :   :       :   :...sunshine <= 0.8: Yes (99/31)
## :       :   :   :       :       sunshine > 0.8:
## :       :   :   :       :       :...wind_gust_speed <= 35: No (223/66)
## :       :   :   :       :           wind_gust_speed > 35:
## :       :   :   :       :           :...wind_dir_3pm in [NW-NNW]: Yes (148/61)
## :       :   :   :       :               wind_dir_3pm in [N-WNW]:
## :       :   :   :       :               :...wind_dir_3pm in [N-NNE]: Yes (150/67
## :       :   :   :       :                   wind_dir_3pm in [NE-WNW]:
## :       :   :   :       :                   :...wind_dir_9am = ENE: [S24]
## :       :   :   :       :                       wind_dir_9am in [N-NE]:
```

```
## :       :   :   :       :                       :...sunshine > 6.5: No (47/9)
## :       :   :   :       :                           sunshine <= 6.5: [S25]
## :       :   :   :       wind_dir_9am in [E-NNW]:
## :       :   :   :       :...cloud_3pm <= 3: No (173/21)
## :       :   :   :           cloud_3pm > 3:
## :       :   :   :           :...wind_speed_9am > 7: No (1296/342)
## :       :   :   :               wind_speed_9am <= 7:
## :       :   :   :               :...wind_gust_speed <= 41:
## :       :   :   :                   :...wind_speed_9am <= 2: Yes (41/20)
## :       :   :   :                   :   wind_speed_9am > 2:
## :       :   :   :                   :   :...wind_speed_3pm > 13:
## :       :   :   :                   :       :...wind_gust_speed <= 33: Yes (42/18
## :       :   :   :                   :       :   wind_gust_speed > 33: No (64/19)
## :       :   :   :                   :       wind_speed_3pm <= 13:
## :       :   :   :                   :       :...wind_speed_3pm > 9: No (49/4)
## :       :   :   :                   :           wind_speed_3pm <= 9: [S26]
## :       :   :   :                   wind_gust_speed > 41:
## :       :   :   :                   :...wind_speed_3pm <= 15: No (17/6)
## :       :   :   :                       wind_speed_3pm > 15:
## :       :   :   :                       :...wind_dir_3pm in [WNW-NNW]: Yes (17/1
## :       :   :   :                           wind_dir_3pm in [N-W]:
## :       :   :   :                           :...cloud_9am <= 4: No (3)
## :       :   :   :                               cloud_9am > 4:
## :       :   :   :                               :...cloud_9am <= 5: Yes (7)
## :       :   :   :                                   cloud_9am > 5: [S27]
## :       :   :   wind_gust_speed > 54:
## :       :   :   :...humidity_3pm > 49:
## :       :   :       :...wind_speed_9am <= 28: Yes (345/104)
## :       :   :       :   wind_speed_9am > 28:
## :       :   :       :   :...max_temp <= 13.5: Yes (16/1)
## :       :   :       :       max_temp > 13.5:
## :       :   :       :       :...wind_gust_speed <= 61:
## :       :   :       :           :...temp_3pm <= 25.2: No (30/3)
## :       :   :       :           :   temp_3pm > 25.2: Yes (2)
## :       :   :       :           wind_gust_speed > 61:
## :       :   :       :           :...evaporation > 4.4:
```

```
## :       :   :       :               :...evaporation <= 12.9: Yes (37/9)
## :       :   :       :               :   evaporation > 12.9: No (3)
## :       :   :       :               evaporation <= 4.4:
## :       :   :       :               :...cloud_9am <= 4: Yes (4)
## :       :   :       :                   cloud_9am > 4:
## :       :   :       :                   :...humidity_9am <= 80: No (23/3)
## :       :   :       :                       humidity_9am > 80: Yes (2)
## :       :   :       humidity_3pm <= 49:
## :       :   :       :...pressure_3pm > 1006.1:
## :       :   :           :...cloud_3pm <= 7: No (392/113)
## :       :   :           :   cloud_3pm > 7: Yes (35/14)
## :       :   :           pressure_3pm <= 1006.1:
## :       :   :           :...humidity_3pm <= 22:
## :       :   :               :...wind_gust_dir in [N-ESE]:
## :       :   :               :   :...min_temp <= 11.7: No (6)
## :       :   :               :   :   min_temp > 11.7: Yes (18/5)
## :       :   :               :   wind_gust_dir in [SE-NNW]:
## :       :   :               :   :...pressure_9am <= 1006.5: No (19)
## :       :   :               :       pressure_9am > 1006.5:
## :       :   :               :       :...wind_speed_9am <= 28: No (22/3)
## :       :   :               :           wind_speed_9am > 28: Yes (10/1)
## :       :   :               humidity_3pm > 22:
## :       :   :               :...min_temp > 22.7: Yes (21/2)
## :       :   :                   min_temp <= 22.7:
## :       :   :                   :...sunshine <= 3.9:
## :       :   :                       :...wind_speed_9am <= 46: Yes (69/16)
## :       :   :                       :   wind_speed_9am > 46: No (4)
## :       :   :                       sunshine > 3.9:
## :       :   :                       :...wind_dir_3pm in [WNW-NNW]:
## :       :   :                           :...wind_gust_dir = NNW: No (19/5)
## :       :   :                           :   wind_gust_dir in [N-NW]:
## :       :   :                           :   :...humidity_9am > 52: Yes (61/17)
## :       :   :                           :       humidity_9am <= 52: [S28]
## :       :   :                           wind_dir_3pm in [N-W]:
## :       :   :                           :...cloud_9am > 5:
## :       :   :                               :...wind_dir_9am in [NW-NNW]: [S29]
```

```
## :       :   :                               :   wind_dir_9am in [N-WNW]:
## :       :   :                               :   :...max_temp <= 31.2: No (21)
## :       :   :                               :       max_temp > 31.2: [S30]
## :       :   :                               cloud_9am <= 5:
## :       :   :                               :...temp_3pm <= 13.4: Yes (4)
## :       :   :                                   temp_3pm > 13.4:
## :       :   :                                   :...wind_dir_9am in [N-ENE]: [S3
## :       :   :                                       wind_dir_9am in [E-NNW]: [S3
## :       :   pressure_3pm > 1013.7:
## :       :   :...pressure_3pm > 1021.8:
## :       :       :...temp_9am <= 18.9: No (2348/165)
## :       :       :   temp_9am > 18.9:
## :       :       :   :...pressure_9am > 1029.8: Yes (7)
## :       :       :       pressure_9am <= 1029.8:
## :       :       :       :...wind_gust_dir in [N-WNW]: No (104/24)
## :       :       :           wind_gust_dir in [NW-NNW]: Yes (3)
## :       :       pressure_3pm <= 1021.8:
## :       :       :...pressure_9am > 1026:
## :       :           :...sunshine > 7.3: No (7)
## :       :           :   sunshine <= 7.3:
## :       :           :   :...wind_dir_3pm in [N-ESE]: Yes (15/2)
## :       :           :       wind_dir_3pm in [SE-NNW]: No (8/2)
## :       :           pressure_9am <= 1026:
## :       :           :...sunshine > 4.7:
## :       :               :...humidity_3pm <= 25: No (107/1)
## :       :               :   humidity_3pm > 25:
## :       :               :   :...wind_gust_speed <= 46:
## :       :               :       :...wind_dir_3pm in [N-WSW]: No (1580/153)
## :       :               :       :   wind_dir_3pm in [W-NNW]:
## :       :               :       :   :...humidity_3pm <= 48: No (226/22)
## :       :               :       :       humidity_3pm > 48:
## :       :               :       :       :...temp_9am <= 7.2: No (29)
## :       :               :       :           temp_9am > 7.2:
## :       :               :       :           :...cloud_3pm > 5: [S33]
## :       :               :       :               cloud_3pm <= 5:
## :       :               :       :               :...max_temp > 22.2: No (20)
```

```
## :       :                :       :                   max_temp <= 22.2: [S34]
## :       :                :       wind_gust_speed > 46:
## :       :                :       :...wind_gust_dir in [N-WSW]: No (355/71)
## :       :                :           wind_gust_dir in [W-NNW]:
## :       :                :           :...temp_9am > 21.4: Yes (9/1)
## :       :                :               temp_9am <= 21.4:
## :       :                :               :...humidity_9am > 82: [S35]
## :       :                :                   humidity_9am <= 82:
## :       :                :                   :...pressure_3pm > 1015.4: No (61/8)
## :       :                :                       pressure_3pm <= 1015.4:
## :       :                :                       :...max_temp > 20.1: No (19/3)
## :       :                :                           max_temp <= 20.1: [S36]
## :       :                sunshine <= 4.7:
## :       :                :...wind_dir_9am in [ENE-NNW]: No (1329/243)
## :       :                    wind_dir_9am in [N-NE]:
## :       :                    :...evaporation > 7.1:
## :       :                        :...humidity_3pm > 53:
## :       :                        :   :...wind_dir_9am = N: Yes (6)
## :       :                        :   :   wind_dir_9am in [NNE-NE]: No (5)
## :       :                        :   humidity_3pm <= 53:
## :       :                        :   :...temp_9am <= 24.4: No (51)
## :       :                        :       temp_9am > 24.4:
## :       :                        :       :...wind_dir_3pm in [N-SW]: No (5)
## :       :                        :           wind_dir_3pm in [WSW-NNW]: Yes (3)
## :       :                        evaporation <= 7.1:
## :       :                        :...wind_speed_9am <= 0: No (60/8)
## :       :                            wind_speed_9am > 0:
## :       :                            :...cloud_3pm > 7:
## :       :                                :...wind_speed_3pm <= 17: No (43/16)
## :       :                                :   wind_speed_3pm > 17: Yes (30/9)
## :       :                                cloud_3pm <= 7:
## :       :                                :...wind_dir_3pm in [W-NNW]:
## :       :                                    :...temp_9am > 13.6: No (52/13)
## :       :                                    :   temp_9am <= 13.6: [S37]
## :       :                                    wind_dir_3pm in [N-WSW]:
## :       :                                    :...cloud_9am <= 6:
```

```
## :       :                                       :...humidity_3pm > 49: No (69/6)
## :       :                                       :   humidity_3pm <= 49: [S38]
## :       :                                       cloud_9am > 6:
## :       :                                       :...cloud_9am > 7: No (26/4)
## :       :                                           cloud_9am <= 7: [S39]
## :       sunshine > 7.7:
## :       :...wind_gust_speed <= 50:
## :           :...humidity_3pm <= 49: No (39314/1464)
## :           :   humidity_3pm > 49:
## :           :   :...pressure_9am > 1012.2: No (19848/1907)
## :           :       pressure_9am <= 1012.2:
## :           :       :...sunshine > 11.6: No (222/13)
## :           :           sunshine <= 11.6:
## :           :           :...cloud_3pm > 7:
## :           :               :...wind_speed_9am > 7: No (88/28)
## :           :               :   wind_speed_9am <= 7:
## :           :               :   :...max_temp <= 25.3: Yes (10)
## :           :               :       max_temp > 25.3:
## :           :               :       :...humidity_3pm <= 61: No (5)
## :           :               :           humidity_3pm > 61: Yes (3)
## :           :               cloud_3pm <= 7:
## :           :               :...wind_gust_speed <= 37: No (1022/168)
## :           :                   wind_gust_speed > 37:
## :           :                   :...rainfall > 0.3:
## :           :                       :...wind_speed_3pm <= 19:
## :           :                       :   :...pressure_3pm <= 1010.8: Yes (40/11)
## :           :                       :   :   pressure_3pm > 1010.8: No (9/1)
## :           :                       :   wind_speed_3pm > 19:
## :           :                       :   :...cloud_3pm > 6: Yes (9/3)
## :           :                       :       cloud_3pm <= 6:
## :           :                       :       :...wind_speed_3pm > 30:
## :           :                       :           :...cloud_3pm <= 5: Yes (5)
## :           :                       :           :   cloud_3pm > 5: No (2)
## :           :                       :           wind_speed_3pm <= 30:
## :           :                       :           :...cloud_3pm > 4: No (31/1)
## :           :                       :               cloud_3pm <= 4: [S40]
```

```
## :           :                       rainfall <= 0.3:
## :           :                       :...wind_gust_dir in [WSW-NNW]:
## :           :                           :...cloud_9am > 5: No (104/47)
## :           :                           :   cloud_9am <= 5:
## :           :                           :   :...evaporation <= 4.5:
## :           :                           :       :...min_temp <= 8.9: No (6/1)
## :           :                           :       :   min_temp > 8.9: Yes (14/1)
## :           :                           :       evaporation > 4.5:
## :           :                           :       :...wind_gust_speed <= 46: [S41]
## :           :                           :           wind_gust_speed > 46: [S42]
## :           :                           wind_gust_dir in [N-SW]:
## :           :                           :...wind_gust_speed <= 46: No (389/65)
## :           :                               wind_gust_speed > 46:
## :           :                               :...wind_dir_9am in [N-NNE]: [S43]
## :           :                                   wind_dir_9am in [NE-NNW]:
## :           :                                   :...max_temp > 32: [S44]
## :           :                                       max_temp <= 32: [S45]
## :           wind_gust_speed > 50:
## :           :...sunshine > 10.6: No (1989/95)
## :               sunshine <= 10.6:
## :               :...humidity_3pm <= 50:
## :                   :...wind_dir_3pm in [ENE-NNW]:
## :                   :   :...wind_dir_3pm in [ENE-WNW]:
## :                   :   :   :...wind_speed_3pm <= 13:
## :                   :   :   :   :...humidity_9am > 65:
## :                   :   :   :   :   :...temp_9am <= 15.2: No (23/1)
## :                   :   :   :   :   :   temp_9am > 15.2:
## :                   :   :   :   :   :   :...wind_gust_speed <= 61: No (28/13)
## :                   :   :   :   :   :       wind_gust_speed > 61: Yes (16/1)
## :                   :   :   :   :   humidity_9am <= 65:
## :                   :   :   :   :   :...temp_3pm <= 29.1: No (90/7)
## :                   :   :   :   :       temp_3pm > 29.1:
## :                   :   :   :   :       :...humidity_3pm <= 24: [S46]
## :                   :   :   :   :           humidity_3pm > 24:
## :                   :   :   :   :           :...evaporation > 12.2: Yes (6)
## :                   :   :   :   :               evaporation <= 12.2:
```

```
## :                    :   :   :   :               :...cloud_9am > 5: No (5)
## :                    :   :   :   :                   cloud_9am <= 5: [S47]
## :                    :   :   :   wind_speed_3pm > 13:
## :                    :   :   :   :...pressure_3pm > 1009.7: No (2075/97)
## :                    :   :   :       pressure_3pm <= 1009.7:
## :                    :   :   :       :...humidity_3pm <= 33:
## :                    :   :   :           :...humidity_9am <= 53: No (631/34)
## :                    :   :   :           :   humidity_9am > 53:
## :                    :   :   :           :   :...wind_gust_speed <= 63: No (114/1
## :                    :   :   :           :       wind_gust_speed > 63: [S48]
## :                    :   :   :           humidity_3pm > 33:
## :                    :   :   :           :...wind_dir_3pm = ENE: Yes (13/5)
## :                    :   :   :               wind_dir_3pm in [E-WNW]:
## :                    :   :   :               :...pressure_3pm <= 999.3: Yes (21/8
## :                    :   :   :                   pressure_3pm > 999.3:
## :                    :   :   :                   :...humidity_3pm <= 40: [S49]
## :                    :   :   :                       humidity_3pm > 40:
## :                    :   :   :                       :...humidity_9am <= 34: [S50
## :                    :   :   :                           humidity_9am > 34:
## :                    :   :   :                           :...humidity_9am <= 61:
## :                    :   :   :                               humidity_9am > 61: [
## :                    :   :   wind_dir_3pm in [NW-NNW]:
## :                    :   :   :...pressure_3pm > 1016: No (69/2)
## :                    :   :       pressure_3pm <= 1016:
## :                    :   :       :...humidity_3pm <= 21:
## :                    :   :           :...humidity_3pm <= 11:
## :                    :   :           :   :...wind_gust_speed <= 74: No (167/10)
## :                    :   :           :   :   wind_gust_speed > 74:
## :                    :   :           :   :   :...cloud_3pm <= 1: Yes (3)
## :                    :   :           :   :       cloud_3pm > 1: No (23/4)
## :                    :   :           :   humidity_3pm > 11:
## :                    :   :           :   :...wind_speed_3pm <= 43: No (305/57)
## :                    :   :           :       wind_speed_3pm > 43:
## :                    :   :           :       :...max_temp <= 32.6: Yes (9/1)
## :                    :   :           :           max_temp > 32.6: No (12/2)
## :                    :   :           humidity_3pm > 21:
```

```
## :                    :   :           :...cloud_3pm > 6: Yes (84/30)
## :                    :   :               cloud_3pm <= 6:
## :                    :   :               :...wind_gust_speed <= 70:
## :                    :   :                   :...wind_dir_3pm = NW: No (252/44)
## :                    :   :                   :   wind_dir_3pm = NNW:
## :                    :   :                   :   :...evaporation > 6.7: No (16/1)
## :                    :   :                   :       evaporation <= 6.7: [S53]
## :                    :   :                   wind_gust_speed > 70:
## :                    :   :                   :...temp_3pm > 32.6: Yes (15/1)
## :                    :   :                       temp_3pm <= 32.6:
## :                    :   :                       :...evaporation > 6.5: No (9)
## :                    :   :                           evaporation <= 6.5: [S54]
## :                    :   wind_dir_3pm in [N-NE]:
## :                    :   :...pressure_3pm > 1017.1: No (106/4)
## :                    :       pressure_3pm <= 1017.1:
## :                    :       :...humidity_3pm <= 14: No (109/7)
## :                    :           humidity_3pm > 14:
## :                    :           :...cloud_3pm > 7:
## :                    :               :...wind_gust_speed > 54:
## :                    :               :   :...sunshine <= 8.6: Yes (29/3)
## :                    :               :   :   sunshine > 8.6: No (2)
## :                    :               :   wind_gust_speed <= 54:
## :                    :               :   :...cloud_9am > 5: No (7)
## :                    :               :       cloud_9am <= 5:
## :                    :               :       :...humidity_3pm <= 25: No (2)
## :                    :               :           humidity_3pm > 25: Yes (4)
## :                    :               cloud_3pm <= 7:
## :                    :               :...wind_gust_dir in [SSW-NNW]:
## :                    :                   :...wind_gust_speed > 74: Yes (34/8)
## :                    :                   :   wind_gust_speed <= 74:
## :                    :                   :   :...wind_dir_3pm = N:
## :                    :                   :       :...temp_9am <= 28.5: No (114/41
## :                    :                   :       :   temp_9am > 28.5: Yes (8)
## :                    :                   :       wind_dir_3pm in [NNE-NE]:
## :                    :                   :       :...temp_9am <= 17: No (8/1)
## :                    :                   :           temp_9am > 17:
```

```
## :                    :                    :           :...sunshine <= 8.4: No (3)
## :                    :                    :               sunshine > 8.4: [S55]
## :                    :                    wind_gust_dir in [N-S]:
## :                    :                    :...rainfall > 0.1:
## :                    :                        :...pressure_9am <= 1013: Yes (14/2)
## :                    :                        :   pressure_9am > 1013: No (17/2)
## :                    :                        rainfall <= 0.1:
## :                    :                        :...cloud_9am > 6: No (31/7)
## :                    :                            cloud_9am <= 6:
## :                    :                            :...wind_dir_3pm = NE: [S56]
## :                    :                                wind_dir_3pm in [N-NNE]:
## :                    :                                :...sunshine <= 8.3: No (21/1
## :                    :                                    sunshine > 8.3: [S57]
## :                    humidity_3pm > 50:
## :                    :...min_temp > 25.1: Yes (52/17)
## :                        min_temp <= 25.1:
## :                        :...wind_gust_dir in [N-SW]:
## :                            :...wind_speed_9am <= 2:
## :                            :   :...wind_gust_speed <= 52: No (6)
## :                            :   :   wind_gust_speed > 52:
## :                            :   :   :...humidity_9am <= 71: No (6/1)
## :                            :   :       humidity_9am > 71: Yes (17/2)
## :                            :   wind_speed_9am > 2:
## :                            :   :...wind_gust_speed > 65: No (172/64)
## :                            :       wind_gust_speed <= 65:
## :                            :       :...pressure_3pm <= 1011.6:
## :                            :           :...cloud_3pm > 5:
## :                            :           :   :...wind_speed_3pm <= 28: Yes (24/4)
## :                            :           :   :   wind_speed_3pm > 28:
## :                            :           :   :   :...pressure_3pm > 1007.6: No (21
## :                            :           :   :       pressure_3pm <= 1007.6: [S58]
## :                            :           :   cloud_3pm <= 5:
## :                            :           :   :...max_temp <= 20.6: [S59]
## :                            :           :       max_temp > 20.6:
## :                            :           :       :...max_temp <= 23.6: No (44/1)
## :                            :           :           max_temp > 23.6:
```

```
## :                            :           :           :...humidity_9am > 65: [S60]
## :                            :           :               humidity_9am <= 65: [S61
## :                            :           pressure_3pm > 1011.6:
## :                            :           :...wind_gust_dir in [N-SSE]: No (511/67
## :                            :               wind_gust_dir in [S-SW]: [S62]
## :                            wind_gust_dir in [WSW-NNW]:
## :                            :...pressure_3pm > 1012.2:
## :                                :...sunshine > 8.5: No (35/2)
## :                                :   sunshine <= 8.5:
## :                                :   :...wind_gust_dir in [WSW-W]:
## :                                :       :...temp_9am <= 17.4:
## :                                :       :   :...max_temp <= 4.2: Yes (10/3)
## :                                :       :   :   max_temp > 4.2: No (116/19)
## :                                :       :   temp_9am > 17.4:
## :                                :       :   :...pressure_9am <= 1014.7: No (4)
## :                                :       :       pressure_9am > 1014.7: Yes (11/1
## :                                :       wind_gust_dir in [WNW-NNW]:
## :                                :       :...sunshine <= 8.3: No (11/1)
## :                                :           sunshine > 8.3:
## :                                :           :...wind_speed_3pm > 35: No (14/1)
## :                                :               wind_speed_3pm <= 35:
## :                                :               :...wind_gust_speed > 57: [S63]
## :                                :                   wind_gust_speed <= 57:
## :                                :                   :...rainfall <= 0.1: [S64]
## :                                :                       rainfall > 0.1: [S65]
## :                                pressure_3pm <= 1012.2:
## :                                :...pressure_9am > 1014.4: Yes (40/5)
## :                                    pressure_9am <= 1014.4:
## :                                    :...cloud_9am <= 1:
## :                                        :...pressure_3pm <= 999.1: Yes (2)
## :                                        :   pressure_3pm > 999.1: No (15/1)
## :                                        cloud_9am > 1:
## :                                        :...wind_gust_dir = NNW: No (30/10)
## :                                            wind_gust_dir in [WSW-NW]:
## :                                            :...pressure_3pm <= 1002.2: Yes (41/
## :                                                pressure_3pm > 1002.2:
```

```
## :                                                :...humidity_3pm <= 51: No (27/7
## :                                                    humidity_3pm > 51: [S66]
## humidity_3pm > 66:
## :...humidity_3pm <= 81:
##     :...wind_gust_speed > 41:
##     :   :...rainfall > 0.2:
##     :   :   :...temp_9am <= 4.4:
##     :   :   :   :...humidity_3pm <= 72: No (13)
##     :   :   :   :   humidity_3pm > 72:
##     :   :   :   :   :...wind_dir_9am in [W-NNW]: Yes (10/3)
##     :   :   :   :       wind_dir_9am in [N-WSW]:
##     :   :   :   :       :...wind_gust_dir in [N-ENE]: Yes (2)
##     :   :   :   :           wind_gust_dir in [E-NNW]: No (19/2)
##     :   :   :   temp_9am > 4.4:
##     :   :   :   :...wind_gust_speed <= 54:
##     :   :   :       :...sunshine > 9.2: No (48/8)
##     :   :   :       :   sunshine <= 9.2:
##     :   :   :       :   :...wind_speed_3pm <= 24: Yes (1305/457)
##     :   :   :       :       wind_speed_3pm > 24:
##     :   :   :       :       :...wind_gust_speed <= 43: No (75/21)
##     :   :   :       :           wind_gust_speed > 43:
##     :   :   :       :           :...sunshine <= 5.3:
##     :   :   :       :               :...cloud_9am <= 7: Yes (124/42)
##     :   :   :       :               :   cloud_9am > 7:
##     :   :   :       :               :   :...rain_today = No: [S67]
##     :   :   :       :               :       rain_today = Yes:
##     :   :   :       :               :       :...wind_speed_9am <= 28: Yes (55/8)
##     :   :   :       :               :           wind_speed_9am > 28: [S68]
##     :   :   :       :               sunshine > 5.3:
##     :   :   :       :               :...humidity_9am <= 86:
##     :   :   :       :                   :...evaporation > 5.5: [S69]
##     :   :   :       :                   :   evaporation <= 5.5:
##     :   :   :       :                   :   :...temp_3pm > 24.3: Yes (77/36)
##     :   :   :       :                   :       temp_3pm <= 24.3:
##     :   :   :       :                   :       :...rainfall <= 31.4: No (233/69
##     :   :   :       :                   :           rainfall > 31.4: [S70]
```

```
##     :   :   :       :                   humidity_9am > 86:
##     :   :   :       :                   :...wind_speed_9am > 17:
##     :   :   :       :                       :...wind_gust_speed > 44: No (101/45
##     :   :   :       :                       :   wind_gust_speed <= 44: [S71]
##     :   :   :       :                       wind_speed_9am <= 17:
##     :   :   :       :                       :...wind_speed_3pm > 26: Yes (55/12)
##     :   :   :       :                           wind_speed_3pm <= 26: [S72]
##     :   :   :       wind_gust_speed > 54:
##     :   :   :       :...pressure_3pm <= 1006.9:
##     :   :   :           :...max_temp <= 19.3: Yes (286/21)
##     :   :   :           :   max_temp > 19.3:
##     :   :   :           :   :...sunshine <= 8.3:
##     :   :   :           :       :...temp_3pm > 24.1: Yes (61/1)
##     :   :   :           :       :   temp_3pm <= 24.1:
##     :   :   :           :       :   :...temp_3pm <= 20.9: Yes (39/2)
##     :   :   :           :       :       temp_3pm > 20.9:
##     :   :   :           :       :       :...humidity_3pm <= 70: Yes (4)
##     :   :   :           :       :           humidity_3pm > 70: No (8/1)
##     :   :   :           :       sunshine > 8.3:
##     :   :   :           :       :...wind_speed_3pm > 39: No (11/1)
##     :   :   :           :           wind_speed_3pm <= 39:
##     :   :   :           :           :...pressure_3pm <= 997.6: Yes (11)
##     :   :   :           :               pressure_3pm > 997.6:
##     :   :   :           :               :...pressure_9am <= 1002.3: No (8)
##     :   :   :           :                   pressure_9am > 1002.3: Yes (47/14)
##     :   :   :           pressure_3pm > 1006.9:
##     :   :   :           :...cloud_9am <= 4:
##     :   :   :               :...rain_today = No: No (20/3)
##     :   :   :               :   rain_today = Yes:
##     :   :   :               :   :...humidity_3pm > 76: Yes (24/3)
##     :   :   :               :       humidity_3pm <= 76:
##     :   :   :               :       :...wind_speed_9am > 26: No (33/11)
##     :   :   :               :           wind_speed_9am <= 26:
##     :   :   :               :           :...wind_dir_3pm in [N-S]: No (11/4)
##     :   :   :               :               wind_dir_3pm in [SSW-NNW]:
##     :   :   :               :               :...pressure_3pm <= 1007.6: No (2)
```

```
##     :   :   :               :                   pressure_3pm > 1007.6: Yes (32/3
##     :   :   :               cloud_9am > 4:
##     :   :   :               :...wind_gust_speed <= 63: Yes (620/190)
##     :   :   :                   wind_gust_speed > 63:
##     :   :   :                   :...humidity_3pm > 70: Yes (264/36)
##     :   :   :                       humidity_3pm <= 70:
##     :   :   :                       :...cloud_3pm <= 4:
##     :   :   :                           :...wind_dir_3pm in [N-SE]: Yes (2)
##     :   :   :                           :   wind_dir_3pm in [SSE-NNW]: No (5)
##     :   :   :                           cloud_3pm > 4:
##     :   :   :                           :...cloud_3pm <= 7:
##     :   :   :                               :...rain_today = Yes: Yes (72/15)
##     :   :   :                               :   rain_today = No:
##     :   :   :                               :   :...temp_9am <= 13.8: Yes (5)
##     :   :   :                               :       temp_9am > 13.8: [S73]
##     :   :   :                               cloud_3pm > 7:
##     :   :   :                               :...sunshine <= 8.4: Yes (3)
##     :   :   :                                   sunshine > 8.4:
##     :   :   :                                   :...cloud_9am <= 7: No (5)
##     :   :   :                                       cloud_9am > 7: [S74]
##     :   :   rainfall <= 0.2:
##     :   :   :...sunshine <= 7.5:
##     :   :       :...pressure_3pm <= 1016.1:
##     :   :       :   :...min_temp <= 16.1: Yes (313/76)
##     :   :       :   :   min_temp > 16.1:
##     :   :       :   :   :...wind_speed_3pm <= 28:
##     :   :       :   :       :...humidity_9am <= 85: Yes (149/48)
##     :   :       :   :       :   humidity_9am > 85: No (9/1)
##     :   :       :   :       wind_speed_3pm > 28:
##     :   :       :   :       :...humidity_9am <= 69:
##     :   :       :   :           :...max_temp <= 27.7:
##     :   :       :   :           :   :...wind_speed_9am <= 13: No (2)
##     :   :       :   :           :   :   wind_speed_9am > 13: Yes (15/1)
##     :   :       :   :           :   max_temp > 27.7:
##     :   :       :   :           :   :...temp_3pm <= 29: No (10/1)
##     :   :       :   :           :       temp_3pm > 29: Yes (2)
```

```
##     :   :       :   :           humidity_9am > 69:
##     :   :       :   :           :...wind_speed_9am > 39: Yes (3)
##     :   :       :   :               wind_speed_9am <= 39:
##     :   :       :   :               :...max_temp <= 34: No (39/4)
##     :   :       :   :                   max_temp > 34: Yes (2)
##     :   :       :   pressure_3pm > 1016.1:
##     :   :       :   :...temp_9am <= 8.7: No (21/2)
##     :   :       :       temp_9am > 8.7:
##     :   :       :       :...humidity_3pm > 74: Yes (92/38)
##     :   :       :           humidity_3pm <= 74:
##     :   :       :           :...wind_gust_dir in [W-NNW]:
##     :   :       :               :...wind_speed_9am <= 30: Yes (15/2)
##     :   :       :               :   wind_speed_9am > 30: No (2)
##     :   :       :               wind_gust_dir in [N-WSW]:
##     :   :       :               :...cloud_9am > 5: No (74/15)
##     :   :       :                   cloud_9am <= 5:
##     :   :       :                   :...rainfall <= 0.1: No (34/16)
##     :   :       :                       rainfall > 0.1: Yes (5)
##     :   :       sunshine > 7.5:
##     :   :       :...wind_gust_speed <= 54:
##     :   :           :...cloud_3pm <= 5:
##     :   :           :   :...wind_dir_3pm in [N-SW]: No (902/186)
##     :   :           :   :   wind_dir_3pm in [WSW-NNW]:
##     :   :           :   :   :...max_temp <= 13.7: No (29/3)
##     :   :           :   :       max_temp > 13.7:
##     :   :           :   :       :...wind_speed_3pm <= 19: Yes (58/14)
##     :   :           :   :           wind_speed_3pm > 19: No (114/48)
##     :   :           :   cloud_3pm > 5:
##     :   :           :   :...wind_speed_9am <= 7:
##     :   :           :       :...humidity_3pm <= 68: No (5)
##     :   :           :       :   humidity_3pm > 68:
##     :   :           :       :   :...min_temp <= 0.3: No (3)
##     :   :           :       :       min_temp > 0.3: Yes (27/3)
##     :   :           :       wind_speed_9am > 7:
##     :   :           :       :...pressure_9am > 1017.6: No (158/42)
##     :   :           :           pressure_9am <= 1017.6:
```

```
##     :   :           :           :...wind_speed_9am <= 9:
##     :   :           :               :...max_temp <= 22.8: Yes (2)
##     :   :           :               :   max_temp > 22.8: No (14)
##     :   :           :               wind_speed_9am > 9:
##     :   :           :               :...rainfall <= 0: Yes (150/72)
##     :   :           :                   rainfall > 0:
##     :   :           :                   :...wind_dir_3pm in [N-NW]: No (23/5)
##     :   :           :                       wind_dir_3pm = NNW: Yes (2)
##     :   :           wind_gust_speed > 54:
##     :   :           :...cloud_3pm > 6: Yes (186/69)
##     :   :               cloud_3pm <= 6:
##     :   :               :...humidity_9am > 86:
##     :   :                   :...temp_3pm > 26.3: Yes (3)
##     :   :                   :   temp_3pm <= 26.3:
##     :   :                   :   :...wind_speed_9am > 7: No (51/5)
##     :   :                   :       wind_speed_9am <= 7:
##     :   :                   :       :...temp_3pm <= 23.3: Yes (4)
##     :   :                   :           temp_3pm > 23.3: No (2)
##     :   :                   humidity_9am <= 86:
##     :   :                   :...sunshine > 10:
##     :   :                       :...wind_speed_3pm > 26: No (16)
##     :   :                       :   wind_speed_3pm <= 26:
##     :   :                       :   :...cloud_3pm <= 3: No (3)
##     :   :                       :       cloud_3pm > 3: Yes (3)
##     :   :                       sunshine <= 10:
##     :   :                       :...wind_dir_3pm in [SSW-NNW]:
##     :   :                           :...wind_speed_9am <= 20:
##     :   :                           :   :...sunshine > 8.6: No (3/1)
##     :   :                           :   :   sunshine <= 8.6:
##     :   :                           :   :   :...wind_speed_9am <= 2: No (4/1)
##     :   :                           :   :       wind_speed_9am > 2: Yes (46/5)
##     :   :                           :   wind_speed_9am > 20:
##     :   :                           :   :...cloud_9am <= 2: Yes (4)
##     :   :                           :       cloud_9am > 2: [S75]
##     :   :                           wind_dir_3pm in [N-S]:
##     :   :                           :...rainfall > 0.1:
```

```
##     :   :                               :...humidity_9am <= 80: No (26/12)
##     :   :                               :   humidity_9am > 80: Yes (9)
##     :   :                               rainfall <= 0.1:
##     :   :                               :...max_temp <= 21:
##     :   :                                   :...humidity_3pm > 75: Yes (12)
##     :   :                                   :   humidity_3pm <= 75: [S76]
##     :   :                                   max_temp > 21:
##     :   :                                   :...wind_gust_dir in [N-NE]: [S77]
##     :   :                                       wind_gust_dir in [ENE-NNW]:
##     :   :                                       :...cloud_9am <= 3: [S78]
##     :   :                                           cloud_9am > 3: [S79]
##     :   wind_gust_speed <= 41:
##     :   :...rain_today = No:
##     :       :...sunshine <= 0.7:
##     :       :   :...pressure_3pm > 1017.2: No (257/71)
##     :       :   :   pressure_3pm <= 1017.2:
##     :       :   :   :...wind_dir_3pm in [N-SSE]:
##     :       :   :       :...cloud_3pm <= 5: Yes (10/1)
##     :       :   :       :   cloud_3pm > 5:
##     :       :   :       :   :...wind_gust_dir = NNW: No (4)
##     :       :   :       :       wind_gust_dir in [N-NW]:
##     :       :   :       :       :...cloud_3pm <= 7:
##     :       :   :       :           :...wind_dir_3pm in [N-NE]: Yes (20/3)
##     :       :   :       :           :   wind_dir_3pm in [ENE-SSE]: No (32/15)
##     :       :   :       :           cloud_3pm > 7:
##     :       :   :       :           :...pressure_9am <= 1019.3: Yes (54/9)
##     :       :   :       :               pressure_9am > 1019.3: No (5)
##     :       :   :       wind_dir_3pm in [S-NNW]:
##     :       :   :       :...humidity_9am > 94: No (9)
##     :       :   :           humidity_9am <= 94:
##     :       :   :           :...humidity_9am <= 82:
##     :       :   :               :...wind_speed_3pm <= 11: Yes (9/2)
##     :       :   :               :   wind_speed_3pm > 11: No (31/7)
##     :       :   :               humidity_9am > 82:
##     :       :   :               :...cloud_9am <= 6: No (2)
##     :       :   :                   cloud_9am > 6:
```

```
##     :       :   :                   :...wind_speed_9am > 9: Yes (15)
##     :       :   :                       wind_speed_9am <= 9:
##     :       :   :                       :...max_temp <= 19.6: Yes (6)
##     :       :   :                           max_temp > 19.6:
##     :       :   :                           :...max_temp <= 24.7: No (7)
##     :       :   :                               max_temp > 24.7: Yes (3)
##     :       :   sunshine > 0.7:
##     :       :   :...pressure_3pm > 1022.9: No (1421/180)
##     :       :       pressure_3pm <= 1022.9:
##     :       :       :...cloud_3pm <= 5: No (3535/714)
##     :       :           cloud_3pm > 5:
##     :       :           :...pressure_3pm <= 1008.5:
##     :       :               :...humidity_3pm > 72: Yes (132/44)
##     :       :               :   humidity_3pm <= 72:
##     :       :               :   :...rainfall <= 0.5: No (117/44)
##     :       :               :       rainfall > 0.5:
##     :       :               :       :...cloud_9am <= 5: No (3)
##     :       :               :           cloud_9am > 5:
##     :       :               :           :...sunshine <= 5: Yes (9)
##     :       :               :               sunshine > 5: No (5/1)
##     :       :               pressure_3pm > 1008.5:
##     :       :               :...wind_gust_dir in [N-NE]:
##     :       :                   :...rainfall <= 0:
##     :       :                   :   :...wind_gust_speed <= 33:
##     :       :                   :   :   :...wind_speed_9am <= 2: No (16)
##     :       :                   :   :   :   wind_speed_9am > 2:
##     :       :                   :   :   :   :...wind_speed_9am <= 15: No (93/27)
##     :       :                   :   :   :       wind_speed_9am > 15:
##     :       :                   :   :   :       :...cloud_3pm <= 7: Yes (6)
##     :       :                   :   :   :           cloud_3pm > 7: No (6/2)
##     :       :                   :   :   wind_gust_speed > 33:
##     :       :                   :   :   :...sunshine <= 2.3: No (8)
##     :       :                   :   :       sunshine > 2.3: [S80]
##     :       :                   :   rainfall > 0:
##     :       :                   :   :...wind_gust_speed > 39: No (4)
##     :       :                   :       wind_gust_speed <= 39:
```

```
##     :       :                   :       :...cloud_9am > 7: Yes (24/5)
##     :       :                   :           cloud_9am <= 7:
##     :       :                   :           :...min_temp <= 2.8: Yes (6)
##     :       :                   :               min_temp > 2.8:
##     :       :                   :               :...min_temp <= 7.1: No (10/1)
##     :       :                   :                   min_temp > 7.1: [S81]
##     :       :                   wind_gust_dir in [ENE-NNW]:
##     :       :                   :...wind_speed_3pm > 15: No (758/199)
##     :       :                       wind_speed_3pm <= 15:
##     :       :                       :...wind_gust_speed <= 28: No (297/75)
##     :       :                           wind_gust_speed > 28:
##     :       :                           :...wind_gust_dir in [ENE-SSE]:
##     :       :                               :...max_temp > 28.6: Yes (10)
##     :       :                               :   max_temp <= 28.6:
##     :       :                               :   :...wind_speed_9am > 7: [S82]
##     :       :                               :       wind_speed_9am <= 7: [S83]
##     :       :                               wind_gust_dir in [S-NNW]:
##     :       :                               :...humidity_3pm <= 76: No (235/75)
##     :       :                                   humidity_3pm > 76: [S84]
##     :       rain_today = Yes:
##     :       :...cloud_9am <= 1: No (85/15)
##     :           cloud_9am > 1:
##     :           :...pressure_3pm <= 1013.8:
##     :               :...humidity_3pm > 75:
##     :               :   :...cloud_3pm > 6: Yes (251/68)
##     :               :   :   cloud_3pm <= 6:
##     :               :   :   :...wind_dir_3pm in [ESE-NNW]: Yes (121/45)
##     :               :   :       wind_dir_3pm in [N-E]:
##     :               :   :       :...wind_dir_9am in [NW-NNW]: No (13)
##     :               :   :           wind_dir_9am in [N-WNW]:
##     :               :   :           :...sunshine <= 6: Yes (7/1)
##     :               :   :               sunshine > 6:
##     :               :   :               :...evaporation <= 4.5: Yes (3)
##     :               :   :                   evaporation > 4.5: No (47/19)
##     :               :   humidity_3pm <= 75:
##     :               :   :...sunshine > 9.4:
```

```
##     :                :       :...wind_speed_3pm <= 30: No (48/9)
##     :                :       :   wind_speed_3pm > 30: Yes (3)
##     :                :       sunshine <= 9.4:
##     :                :       :...wind_gust_speed > 30:
##     :                :           :...cloud_3pm <= 5:
##     :                :           :   :...temp_3pm > 18.1: No (187/76)
##     :                :           :   :   temp_3pm <= 18.1:
##     :                :           :   :   :...cloud_3pm <= 2: No (4)
##     :                :           :   :       cloud_3pm > 2: Yes (64/15)
##     :                :           :   cloud_3pm > 5:
##     :                :           :   :...temp_9am > 29.1: No (21/5)
##     :                :           :       temp_9am <= 29.1:
##     :                :           :       :...evaporation <= 1.3: No (20/6)
##     :                :           :           evaporation > 1.3: Yes (286/107)
##     :                :           wind_gust_speed <= 30:
##     :                :           :...min_temp <= 5.6: No (13)
##     :                :               min_temp > 5.6:
##     :                :               :...humidity_9am <= 74:
##     :                :                   :...sunshine <= 3.2: Yes (2)
##     :                :                   :   sunshine > 3.2: No (25/2)
##     :                :                   humidity_9am > 74:
##     :                :                   :...wind_speed_9am > 9: No (66/22)
##     :                :                       wind_speed_9am <= 9:
##     :                :                       :...humidity_3pm <= 67: No (9/1)
##     :                :                           humidity_3pm > 67:
##     :                :                           :...rainfall <= 1.5: No (7)
##     :                :                               rainfall > 1.5: Yes (106/47)
##     :                pressure_3pm > 1013.8:
##     :                :...temp_3pm <= 12.4:
##     :                    :...wind_gust_speed <= 33:
##     :                    :   :...cloud_3pm <= 6: No (82/5)
##     :                    :   :   cloud_3pm > 6:
##     :                    :   :   :...wind_speed_3pm <= 6: Yes (5/1)
##     :                    :   :       wind_speed_3pm > 6: No (93/13)
##     :                    :   wind_gust_speed > 33:
##     :                    :   :...humidity_3pm <= 78: No (126/37)
```

```
##     :                   :       humidity_3pm > 78:
##     :                   :       :...cloud_9am <= 6: Yes (12/2)
##     :                   :           cloud_9am > 6: No (9/3)
##     :                   temp_3pm > 12.4:
##     :                   :...cloud_3pm <= 2:
##     :                       :...wind_gust_dir in [WSW-NNW]: No (32/1)
##     :                       :   wind_gust_dir in [N-SW]:
##     :                       :   :...pressure_9am <= 1019.1: No (13)
##     :                       :       pressure_9am > 1019.1:
##     :                       :       :...rainfall <= 8.8: No (17/4)
##     :                       :           rainfall > 8.8: Yes (8)
##     :                       cloud_3pm > 2:
##     :                       :...sunshine <= 0.5: Yes (126/48)
##     :                           sunshine > 0.5:
##     :                           :...wind_gust_speed <= 28: No (521/154)
##     :                               wind_gust_speed > 28:
##     :                               :...humidity_3pm > 77:
##     :                                   :...wind_dir_3pm in [N-SSW]: Yes (164/56
##     :                                   :   wind_dir_3pm in [SW-NNW]:
##     :                                   :   :...temp_9am <= 9: Yes (5)
##     :                                   :       temp_9am > 9: No (65/23)
##     :                                   humidity_3pm <= 77:
##     :                                   :...evaporation <= 1.3: No (106/25)
##     :                                       evaporation > 1.3:
##     :                                       :...rainfall <= 5.1:
##     :                                           :...cloud_9am > 4: [S85]
##     :                                           :   cloud_9am <= 4: [S86]
##     :                                           rainfall > 5.1: [S87]
##     humidity_3pm > 81:
##     :...temp_9am <= 6.4:
##         :...pressure_3pm <= 1014:
##         :   :...temp_9am > 5.6: Yes (52)
##         :   :   temp_9am <= 5.6:
##         :   :   :...wind_gust_speed > 48: Yes (33/1)
##         :   :       wind_gust_speed <= 48:
##         :   :       :...wind_speed_9am <= 15:
```

```
##         :   :           :...temp_9am <= 5.5: Yes (39/7)
##         :   :           :   temp_9am > 5.5: No (3)
##         :   :           wind_speed_9am > 15:
##         :   :           :...humidity_3pm <= 98: No (6)
##         :   :               humidity_3pm > 98: Yes (2)
##         :   pressure_3pm > 1014:
##         :   :...humidity_3pm <= 94:
##         :       :...wind_gust_speed <= 19: No (66/7)
##         :       :   wind_gust_speed > 19:
##         :       :   :...pressure_3pm <= 1015.3:
##         :       :       :...cloud_9am <= 6:
##         :       :       :   :...humidity_9am <= 94:
##         :       :       :   :   :...rainfall <= 2.7: No (52/8)
##         :       :       :   :   :   rainfall > 2.7: Yes (13/1)
##         :       :       :   :   humidity_9am > 94:
##         :       :       :   :   :...wind_dir_9am in [N-W]: No (110/10)
##         :       :       :   :       wind_dir_9am in [WNW-NNW]: Yes (8/2)
##         :       :       :   cloud_9am > 6:
##         :       :       :   :...wind_dir_9am in [N-SE]: Yes (5)
##         :       :       :       wind_dir_9am in [SSE-NNW]:
##         :       :       :       :...temp_3pm <= 6.7: Yes (2)
##         :       :       :           temp_3pm > 6.7: No (7)
##         :       :       pressure_3pm > 1015.3:
##         :       :       :...pressure_3pm > 1024.1:
##         :       :           :...temp_3pm <= 7: Yes (3)
##         :       :           :   temp_3pm > 7: No (22/1)
##         :       :           pressure_3pm <= 1024.1:
##         :       :           :...cloud_3pm > 7:
##         :       :               :...wind_speed_3pm <= 15: Yes (9/1)
##         :       :               :   wind_speed_3pm > 15: No (20/7)
##         :       :               cloud_3pm <= 7:
##         :       :               :...pressure_9am > 1018.9: Yes (27/4)
##         :       :                   pressure_9am <= 1018.9:
##         :       :                   :...sunshine <= 7.7: Yes (5)
##         :       :                       sunshine > 7.7: No (7/1)
##         :       humidity_3pm > 94:
```

```
##         :       :...wind_gust_dir in [NW-NNW]: Yes (55/7)
##         :           wind_gust_dir in [N-WNW]:
##         :           :...temp_3pm <= -1.2: Yes (72/13)
##         :               temp_3pm > -1.2:
##         :               :...rainfall > 3.3: Yes (103/36)
##         :                   rainfall <= 3.3:
##         :                   :...temp_9am <= -1.2: No (27/3)
##         :                       temp_9am > -1.2:
##         :                       :...wind_gust_speed <= 35: No (27/6)
##         :                           wind_gust_speed > 35:
##         :                           :...rain_today = Yes: No (21/8)
##         :                               rain_today = No:
##         :                               :...rainfall > 0.3: Yes (15/3)
##         :                                   rainfall <= 0.3:
##         :                                   :...rainfall <= 0.1: [S88]
##         :                                       rainfall > 0.1: [S89]
##         temp_9am > 6.4:
##         :...temp_3pm > 30.3:
##             :...rainfall <= 9.6: No (23)
##             :   rainfall > 9.6: Yes (5/1)
##             temp_3pm <= 30.3:
##             :...rainfall > 2.3:
##                 :...humidity_3pm > 88: Yes (2339/183)
##                 :   humidity_3pm <= 88:
##                 :   :...min_temp > 22.2: Yes (147/4)
##                 :       min_temp <= 22.2:
##                 :       :...wind_gust_speed <= 43:
##                 :           :...humidity_3pm <= 82:
##                 :           :   :...wind_dir_3pm in [N-NW]: Yes (120/49)
##                 :           :   :   wind_dir_3pm = NNW: No (4)
##                 :           :   humidity_3pm > 82:
##                 :           :   :...wind_speed_3pm <= 9: Yes (202/33)
##                 :           :       wind_speed_3pm > 9:
##                 :           :       :...wind_gust_speed <= 30:
##                 :           :           :...wind_gust_dir in [N-SE]: Yes (36/8)
##                 :           :           :   wind_gust_dir in [SSE-NNW]:
```

```
##                  :           :           :   :...humidity_9am > 95: Yes (30/9)
##                  :           :           :       humidity_9am <= 95: [S90]
##                  :           :           wind_gust_speed > 30:
##                  :           :           :...wind_speed_9am <= 13: Yes (194/35)
##                  :           :               wind_speed_9am > 13:
##                  :           :               :...humidity_9am <= 98: Yes (154/41)
##                  :           :                   humidity_9am > 98:
##                  :           :                   :...min_temp <= 18: No (15/2)
##                  :           :                       min_temp > 18: Yes (2)
##                  :           wind_gust_speed > 43:
##                  :           :...pressure_3pm <= 1003.8: Yes (111/2)
##                  :               pressure_3pm > 1003.8:
##                  :               :...temp_9am <= 21.2: Yes (600/72)
##                  :                   temp_9am > 21.2:
##                  :                   :...min_temp <= 19.1: Yes (23/2)
##                  :                       min_temp > 19.1:
##                  :                       :...min_temp <= 19.6: No (8/2)
##                  :                           min_temp > 19.6:
##                  :                           :...wind_speed_3pm <= 26: [S91]
##                  :                               wind_speed_3pm > 26: [S92]
##                 rainfall <= 2.3:
##                 :...humidity_3pm > 88:
##                     :...humidity_9am <= 96: Yes (1718/283)
##                     :   humidity_9am > 96:
##                     :   :...wind_dir_3pm in [NW-NNW]: Yes (44/4)
##                     :       wind_dir_3pm in [N-WNW]:
##                     :       :...pressure_3pm <= 1015.4: Yes (192/46)
##                     :           pressure_3pm > 1015.4:
##                     :           :...min_temp <= 2.9: Yes (10)
##                     :               min_temp > 2.9:
##                     :               :...humidity_3pm <= 93:
##                     :                   :...wind_gust_speed <= 39: No (57/17)
##                     :                   :   wind_gust_speed > 39: Yes (8/1)
##                     :                   humidity_3pm > 93:
##                     :                   :...rain_today = Yes: Yes (34/9)
##                     :                       rain_today = No:
```

```
##                        :                        :...wind_speed_3pm <= 4: Yes (6)
##                        :                            wind_speed_3pm > 4: [S93]
##                        humidity_3pm <= 88:
##                        :...pressure_3pm <= 1015.1:
##                            :...cloud_3pm > 5: Yes (695/141)
##                            :   cloud_3pm <= 5:
##                            :   :...temp_3pm <= 17.7: Yes (187/45)
##                            :       temp_3pm > 17.7:
##                            :       :...rain_today = Yes: Yes (33/8)
##                            :           rain_today = No:
##                            :           :...cloud_3pm <= 2:
##                            :               :...rainfall <= 0.1: No (13)
##                            :               :   rainfall > 0.1: Yes (2)
##                            :               cloud_3pm > 2:
##                            :               :...sunshine <= 8.3:
##                            :                   :...rainfall <= 0.7: Yes (15)
##                            :                   :   rainfall > 0.7:
##                            :                   :   :...evaporation <= 3.5: Yes (3)
##                            :                   :       evaporation > 3.5: No (3)
##                            :                   sunshine > 8.3:
##                            :                   :...wind_gust_speed > 33: Yes (164/79
##                            :                       wind_gust_speed <= 33:
##                            :                       :...cloud_3pm <= 4: Yes (4/1)
##                            :                           cloud_3pm > 4:
##                            :                           :...temp_3pm > 20.9: No (32/3
##                            :                               temp_3pm <= 20.9: [S94]
##                            pressure_3pm > 1015.1:
##                            :...humidity_9am > 96:
##                                :...pressure_3pm > 1021.8:
##                                :   :...wind_dir_9am in [N-WSW]: No (67/6)
##                                :   :   wind_dir_9am in [W-NNW]:
##                                :   :   :...wind_gust_speed <= 31: No (18/2)
##                                :   :       wind_gust_speed > 31: Yes (8/1)
##                                :   pressure_3pm <= 1021.8:
##                                :   :...wind_dir_3pm in [N-SW]: No (65/22)
##                                :       wind_dir_3pm in [WSW-NNW]:
```

```
##                              :       :...max_temp <= 17.8: Yes (33/7)
##                              :           max_temp > 17.8:
##                              :           :...cloud_9am <= 7: No (7)
##                              :               cloud_9am > 7: Yes (2)
##                              humidity_9am <= 96:
##                              :...wind_speed_9am > 31: Yes (31/5)
##                                  wind_speed_9am <= 31:
##                                  :...rain_today = Yes:
##                                      :...humidity_3pm > 84: Yes (82/22)
##                                      :   humidity_3pm <= 84:
##                                      :   :...wind_dir_9am in [N-WSW]:
##                                      :       :...pressure_9am <= 1017.6: Yes (5)
##                                      :       :   pressure_9am > 1017.6: No (38/14
##                                      :       wind_dir_9am in [W-NNW]: [S95]
##                                      rain_today = No:
##                                      :...cloud_9am <= 2:
##                                          :...cloud_3pm <= 5: No (9)
##                                          :   cloud_3pm > 5: [S96]
##                                          cloud_9am > 2:
##                                          :...wind_gust_speed > 56:
##                                              :...wind_speed_3pm <= 43: Yes (50/9)
##                                              :   wind_speed_3pm > 43: No (4)
##                                              wind_gust_speed <= 56:
##                                              :...wind_speed_3pm <= 13:
##                                                  :...cloud_3pm <= 5: [S97]
##                                                  :   cloud_3pm > 5:
##                                                  :   :...wind_gust_speed > 22: [S
##                                                  :       wind_gust_speed <= 22: [
##                                                  wind_speed_3pm > 13:
##                                                  :...wind_dir_9am in [NW-NNW]: [S
##                                                      wind_dir_9am in [N-WNW]: [S1
##
## SubTree [S1]
##
## wind_dir_3pm in [N-WNW]: Yes (104/36)
## wind_dir_3pm in [NW-NNW]: No (6/2)
```

```
##
## SubTree [S2]
##
## evaporation <= 3.9: Yes (4)
## evaporation > 3.9: No (13/1)
##
## SubTree [S3]
##
## wind_speed_9am <= 2: Yes (4)
## wind_speed_9am > 2:
## :...cloud_9am <= 6: No (11/2)
##     cloud_9am > 6: Yes (6/2)
##
## SubTree [S4]
##
## humidity_3pm <= 65: No (126/36)
## humidity_3pm > 65: Yes (24/10)
##
## SubTree [S5]
##
## wind_dir_3pm in [N-SW]: No (2)
## wind_dir_3pm in [WSW-NNW]: Yes (7)
##
## SubTree [S6]
##
## wind_speed_3pm > 15: No (155/57)
## wind_speed_3pm <= 15:
## :...min_temp <= 4.1: No (3)
##     min_temp > 4.1: Yes (9/2)
##
## SubTree [S7]
##
## pressure_9am <= 1011.2: No (2)
## pressure_9am > 1011.2: Yes (9)
##
## SubTree [S8]
```

```
##
## wind_dir_9am in [NE-ESE]: Yes (2)
## wind_dir_9am in [SE-NNW]: No (9)
##
## SubTree [S9]
##
## wind_dir_9am in [NE-ESE]: Yes (8/1)
## wind_dir_9am in [SE-NNW]: No (60/19)
##
## SubTree [S10]
##
## humidity_9am <= 91: No (4/1)
## humidity_9am > 91: Yes (12/2)
##
## SubTree [S11]
##
## humidity_9am > 98: No (5)
## humidity_9am <= 98:
## :...wind_dir_9am in [NW-NNW]: Yes (8/1)
##     wind_dir_9am in [N-WNW]:
##     :...max_temp <= 18.7: No (7/1)
##         max_temp > 18.7:
##         :...humidity_3pm <= 59: Yes (28/10)
##             humidity_3pm > 59: No (7/1)
##
## SubTree [S12]
##
## wind_dir_9am in [N-SW]: No (7/1)
## wind_dir_9am in [WSW-NNW]: Yes (6/1)
##
## SubTree [S13]
##
## wind_dir_3pm in [NE-SSW]: No (3)
## wind_dir_3pm in [SW-NNW]:
## :...temp_9am <= 8.3: No (5/1)
##     temp_9am > 8.3: Yes (19/3)
```

```
##
## SubTree [S14]
##
## wind_dir_3pm in [N-W]: No (12)
## wind_dir_3pm in [WNW-NNW]: Yes (2)
##
## SubTree [S15]
##
## humidity_9am <= 90: No (6)
## humidity_9am > 90: Yes (3)
##
## SubTree [S16]
##
## wind_gust_dir in [SSE-NNW]: No (15/1)
## wind_gust_dir in [N-SE]:
## :...cloud_3pm <= 6: No (16/7)
##     cloud_3pm > 6: Yes (6)
##
## SubTree [S17]
##
## wind_gust_dir in [S-NNW]: Yes (105/25)
## wind_gust_dir in [N-SSE]:
## :...evaporation > 9.5: Yes (5)
##     evaporation <= 9.5:
##     :...humidity_9am <= 67: Yes (5)
##         humidity_9am > 67: No (25/8)
##
## SubTree [S18]
##
## wind_dir_9am in [NW-NNW]: Yes (5)
## wind_dir_9am in [N-WNW]:
## :...temp_9am <= 17.8: No (18/4)
##     temp_9am > 17.8: Yes (3)
##
## SubTree [S19]
##
```

```
## wind_gust_dir in [N-NNE]: No (9/2)
## wind_gust_dir in [NE-NNW]:
## :...wind_gust_dir in [NE-ENE]: Yes (12)
##     wind_gust_dir in [E-NNW]:
##     :...wind_speed_9am <= 26: Yes (40/12)
##         wind_speed_9am > 26: No (4)
##
## SubTree [S20]
##
## wind_speed_3pm <= 37: No (183/55)
## wind_speed_3pm > 37: Yes (12/2)
##
## SubTree [S21]
##
## evaporation <= 4.3: No (17)
## evaporation > 4.3:
## :...pressure_9am > 1015.1: No (42/7)
##     pressure_9am <= 1015.1:
##     :...rainfall <= 3.6: No (4)
##         rainfall > 3.6: Yes (12/2)
##
## SubTree [S22]
##
## cloud_9am > 6: No (31/5)
## cloud_9am <= 6:
## :...pressure_3pm > 1020.2: No (78/24)
##     pressure_3pm <= 1020.2:
##     :...pressure_3pm > 1019.3: Yes (11/1)
##         pressure_3pm <= 1019.3:
##         :...cloud_9am > 5: Yes (9/2)
##             cloud_9am <= 5:
##             :...wind_speed_3pm <= 28: Yes (125/52)
##                 wind_speed_3pm > 28: No (55/19)
##
## SubTree [S23]
##
```

```
## humidity_3pm <= 50: No (12/4)
## humidity_3pm > 50:
## :...rainfall > 14.1: Yes (11)
##     rainfall <= 14.1:
##     :...humidity_3pm <= 52: Yes (14/1)
##         humidity_3pm > 52:
##         :...cloud_3pm <= 4:
##             :...wind_speed_9am <= 30: Yes (32/7)
##             :   wind_speed_9am > 30: No (4)
##             cloud_3pm > 4:
##             :...wind_gust_dir in [SW-NNW]: No (40/16)
##                 wind_gust_dir in [N-SSW]:
##                 :...rainfall > 6.6: Yes (12)
##                     rainfall <= 6.6:
##                     :...pressure_3pm <= 1018.6: No (15/5)
##                         pressure_3pm > 1018.6: Yes (7)
##
## SubTree [S24]
##
## wind_speed_9am <= 7: No (11)
## wind_speed_9am > 7:
## :...pressure_3pm <= 1012.7: No (45/12)
##     pressure_3pm > 1012.7: Yes (3)
##
## SubTree [S25]
##
## humidity_3pm <= 47: No (46/15)
## humidity_3pm > 47:
## :...humidity_9am <= 29: No (5)
##     humidity_9am > 29: Yes (78/31)
##
## SubTree [S26]
##
## wind_speed_3pm <= 2: Yes (3)
## wind_speed_3pm > 2:
## :...cloud_9am <= 1: Yes (4/1)
```

```
##     cloud_9am > 1: No (50/14)
##
## SubTree [S27]
##
## cloud_3pm <= 5: No (4/1)
## cloud_3pm > 5:
## :...humidity_9am <= 75: Yes (15/3)
##     humidity_9am > 75: No (3)
##
## SubTree [S28]
##
## humidity_3pm <= 37: No (18/2)
## humidity_3pm > 37: Yes (16/6)
##
## SubTree [S29]
##
## wind_gust_speed <= 59: No (4)
## wind_gust_speed > 59: Yes (6/1)
##
## SubTree [S30]
##
## max_temp <= 33.8: Yes (4)
## max_temp > 33.8: No (4)
##
## SubTree [S31]
##
## wind_gust_speed <= 74: Yes (19/5)
## wind_gust_speed > 74: No (9/2)
##
## SubTree [S32]
##
## wind_speed_9am > 19: No (10)
## wind_speed_9am <= 19:
## :...wind_speed_9am <= 15: No (4)
##     wind_speed_9am > 15: Yes (5/1)
##
```

```
## SubTree [S33]
##
## wind_dir_3pm = W: No (45/15)
## wind_dir_3pm in [WNW-NNW]: Yes (54/23)
##
## SubTree [S34]
##
## temp_9am <= 14.7: No (55/12)
## temp_9am > 14.7: Yes (7)
##
## SubTree [S35]
##
## wind_gust_dir = NNW: No (2)
## wind_gust_dir in [W-NW]:
## :...sunshine <= 7.5: Yes (12/1)
##     sunshine > 7.5: No (4/1)
##
## SubTree [S36]
##
## wind_speed_3pm <= 37: Yes (16/2)
## wind_speed_3pm > 37: No (2)
##
## SubTree [S37]
##
## wind_gust_speed <= 22: No (5)
## wind_gust_speed > 22: Yes (81/33)
##
## SubTree [S38]
##
## wind_gust_speed <= 43: No (22/2)
## wind_gust_speed > 43: Yes (18/6)
##
## SubTree [S39]
##
## rainfall > 0.1: Yes (26/9)
## rainfall <= 0.1:
```

```
## :...wind_speed_3pm <= 35: No (114/31)
##     wind_speed_3pm > 35:
##     :...humidity_9am <= 49: No (2)
##         humidity_9am > 49: Yes (7)
##
## SubTree [S40]
##
## wind_gust_dir in [N-ENE]: Yes (3)
## wind_gust_dir in [E-NNW]: No (14/1)
##
## SubTree [S41]
##
## pressure_3pm > 1002: No (236/43)
## pressure_3pm <= 1002:
## :...wind_speed_3pm <= 20: Yes (9)
##     wind_speed_3pm > 20: No (8/1)
##
## SubTree [S42]
##
## wind_dir_3pm in [N-SW]: No (11)
## wind_dir_3pm in [WSW-NNW]:
## :...wind_gust_dir in [WSW-NW]: Yes (28/10)
##     wind_gust_dir = NNW: No (4)
##
## SubTree [S43]
##
## pressure_9am <= 1010.4: Yes (11/1)
## pressure_9am > 1010.4: No (4)
##
## SubTree [S44]
##
## humidity_3pm > 62: Yes (6)
## humidity_3pm <= 62:
## :...pressure_9am <= 1011.1: Yes (17/6)
##     pressure_9am > 1011.1: No (9)
##
```

```
## SubTree [S45]
##
## cloud_9am > 4: No (50/3)
## cloud_9am <= 4:
## :...evaporation > 7.9: No (10)
##     evaporation <= 7.9:
##     :...wind_dir_3pm in [N-S]: Yes (7/1)
##         wind_dir_3pm in [SSW-NNW]: No (3)
##
## SubTree [S46]
##
## wind_gust_dir in [N-WNW]: No (97/10)
## wind_gust_dir in [NW-NNW]: Yes (4/1)
##
## SubTree [S47]
##
## wind_gust_dir in [N-SSE]: No (20/8)
## wind_gust_dir in [S-NNW]:
## :...wind_dir_9am in [N-E]: Yes (10)
##     wind_dir_9am in [ESE-NNW]:
##     :...pressure_3pm <= 1009.2: Yes (3)
##         pressure_3pm > 1009.2: No (4)
##
## SubTree [S48]
##
## wind_speed_3pm > 30: No (42/5)
## wind_speed_3pm <= 30:
## :...evaporation > 5.1: Yes (12/2)
##     evaporation <= 5.1:
##     :...wind_dir_3pm in [ENE-SW]: Yes (2)
##         wind_dir_3pm in [WSW-WNW]: No (19/4)
##
## SubTree [S49]
##
## cloud_3pm <= 7: No (174/16)
## cloud_3pm > 7:
```

```
## :...pressure_9am <= 1009.4: No (7/1)
##     pressure_9am > 1009.4: Yes (4)
##
## SubTree [S50]
##
## temp_9am <= 28.5: Yes (9/1)
## temp_9am > 28.5: No (3)
##
## SubTree [S51]
##
## wind_dir_3pm in [E-ESE]: Yes (2)
## wind_dir_3pm in [SE-WNW]: No (88/9)
##
## SubTree [S52]
##
## pressure_3pm <= 1002: Yes (10/2)
## pressure_3pm > 1002: No (84/21)
##
## SubTree [S53]
##
## wind_gust_dir in [N-S]:
## :...temp_9am <= 25.7: No (35/5)
## :   temp_9am > 25.7: Yes (4)
## wind_gust_dir in [SSW-NNW]:
## :...rainfall > 0.1: No (18/6)
##     rainfall <= 0.1:
##     :...humidity_9am <= 79: Yes (81/34)
##         humidity_9am > 79: No (6)
##
## SubTree [S54]
##
## wind_dir_3pm = NNW:
## :...min_temp > 19.4: No (3)
## :   min_temp <= 19.4:
## :   :...sunshine <= 8.1: No (2)
## :       sunshine > 8.1: Yes (24/3)
```

```
## wind_dir_3pm = NW:
## :...wind_speed_9am <= 15: No (17/3)
##     wind_speed_9am > 15:
##     :...humidity_9am > 54: Yes (18/3)
##         humidity_9am <= 54:
##         :...wind_gust_dir in [N-WNW]: No (13/2)
##             wind_gust_dir in [NW-NNW]: Yes (20/9)
##
## SubTree [S55]
##
## wind_gust_speed <= 52: No (6/1)
## wind_gust_speed > 52: Yes (41/8)
##
## SubTree [S56]
##
## evaporation <= 11.3: No (60/3)
## evaporation > 11.3:
## :...wind_speed_3pm <= 22: Yes (3)
##     wind_speed_3pm > 22: No (3)
##
## SubTree [S57]
##
## wind_gust_speed <= 63:
## :...evaporation <= 5.9: No (135/28)
## :   evaporation > 5.9:
## :   :...wind_gust_speed <= 57: No (18/5)
## :       wind_gust_speed > 57:
## :       :...wind_speed_3pm <= 33: Yes (10)
## :           wind_speed_3pm > 33: No (6/2)
## wind_gust_speed > 63:
## :...sunshine > 9.9: No (9)
##     sunshine <= 9.9:
##     :...wind_speed_3pm <= 41: Yes (29/6)
##         wind_speed_3pm > 41:
##         :...pressure_3pm > 1007.4: No (11)
##             pressure_3pm <= 1007.4:
```

```
##             :...evaporation <= 4.3: No (2)
##                 evaporation > 4.3: Yes (7)
##
## SubTree [S58]
##
## wind_dir_3pm in [N-ENE]: No (4/1)
## wind_dir_3pm in [E-NNW]: Yes (6)
##
## SubTree [S59]
##
## wind_speed_3pm <= 22: Yes (10/1)
## wind_speed_3pm > 22:
## :...wind_gust_speed > 57: No (6)
##     wind_gust_speed <= 57:
##     :...sunshine <= 9.2: Yes (13/4)
##         sunshine > 9.2: No (3)
##
## SubTree [S60]
##
## pressure_9am <= 1014.5: No (45/15)
## pressure_9am > 1014.5: Yes (8/1)
##
## SubTree [S61]
##
## wind_speed_3pm > 26: No (69/5)
## wind_speed_3pm <= 26:
## :...humidity_3pm <= 58: No (13/3)
##     humidity_3pm > 58: Yes (4)
##
## SubTree [S62]
##
## wind_dir_9am in [N-NNE]: Yes (8/2)
## wind_dir_9am in [NE-NNW]:
## :...wind_gust_dir in [SSW-SW]: No (130/16)
##     wind_gust_dir = S:
##     :...wind_dir_3pm in [SSW-NNW]:
```

```
##         :...wind_gust_speed <= 54: No (4)
##         :   wind_gust_speed > 54:
##         :   :...wind_dir_3pm in [SSW-SW]: Yes (13/3)
##         :       wind_dir_3pm in [WSW-NNW]: No (2)
##         wind_dir_3pm in [N-S]:
##         :...min_temp > 20.5: Yes (8/2)
##             min_temp <= 20.5:
##             :...wind_dir_9am in [NE-S]: No (55/1)
##                 wind_dir_9am in [SSW-NNW]:
##                 :...min_temp <= 7.5: Yes (4)
##                     min_temp > 7.5: No (47/9)
##
## SubTree [S63]
##
## temp_3pm <= 13.4: No (20/6)
## temp_3pm > 13.4: Yes (41/8)
##
## SubTree [S64]
##
## wind_dir_3pm in [N-E]: Yes (12/4)
## wind_dir_3pm in [ESE-NNW]: No (43/9)
##
## SubTree [S65]
##
## temp_9am <= 12.6: Yes (7)
## temp_9am > 12.6: No (7/2)
##
## SubTree [S66]
##
## temp_3pm > 26.1: Yes (11)
## temp_3pm <= 26.1:
## :...rainfall > 0.5: Yes (22/6)
##     rainfall <= 0.5:
##     :...wind_gust_speed > 65: No (35/9)
##         wind_gust_speed <= 65:
##         :...rainfall > 0.3: No (9/1)
```

```
##             rainfall <= 0.3:
##             :...wind_dir_3pm in [N-E]: No (8/1)
##                 wind_dir_3pm in [ESE-NNW]:
##                 :...pressure_9am <= 1011.5:
##                     :...rainfall <= 0.1: No (51/22)
##                     :   rainfall > 0.1: Yes (13/3)
##                     pressure_9am > 1011.5:
##                     :...wind_speed_3pm <= 33: Yes (25/1)
##                         wind_speed_3pm > 33:
##                         :...wind_dir_3pm in [ESE-WNW]: Yes (2)
##                             wind_dir_3pm in [NW-NNW]: No (3)
##
## SubTree [S67]
##
## wind_gust_dir in [N-SSE]: No (9/1)
## wind_gust_dir in [S-NNW]: Yes (8/1)
##
## SubTree [S68]
##
## wind_speed_3pm <= 28: Yes (4)
## wind_speed_3pm > 28: No (5)
##
## SubTree [S69]
##
## wind_dir_3pm in [N-WSW]: Yes (28/7)
## wind_dir_3pm in [W-NNW]: No (4)
##
## SubTree [S70]
##
## min_temp <= 15.6: Yes (8)
## min_temp > 15.6: No (3)
##
## SubTree [S71]
##
## humidity_3pm <= 79: Yes (16/2)
## humidity_3pm > 79: No (2)
```

```
##
## SubTree [S72]
##
## wind_dir_3pm in [N-SE]: No (6)
## wind_dir_3pm in [SSE-NNW]:
## :...sunshine <= 8.6: Yes (21/5)
##     sunshine > 8.6: No (2)
##
## SubTree [S73]
##
## temp_9am <= 22.8: No (6)
## temp_9am > 22.8: Yes (2)
##
## SubTree [S74]
##
## humidity_9am > 93: No (3)
## humidity_9am <= 93:
## :...pressure_3pm <= 1008.6: No (2)
##     pressure_3pm > 1008.6: Yes (7)
##
## SubTree [S75]
##
## wind_dir_9am in [NW-NNW]: Yes (22/7)
## wind_dir_9am in [N-WNW]:
## :...min_temp > 19.2: Yes (7/1)
##     min_temp <= 19.2:
##     :...wind_dir_9am = N: Yes (3)
##         wind_dir_9am in [NNE-WNW]: No (33/4)
##
## SubTree [S76]
##
## humidity_3pm <= 72: Yes (25/9)
## humidity_3pm > 72: No (5)
##
## SubTree [S77]
##
```

```
## wind_dir_9am in [N-NW]: No (55/4)
## wind_dir_9am = NNW: Yes (2)
##
## SubTree [S78]
##
## humidity_3pm <= 76: Yes (8)
## humidity_3pm > 76: No (2)
##
## SubTree [S79]
##
## wind_gust_dir in [ENE-SSE]:
## :...wind_speed_9am <= 9: No (6)
## :   wind_speed_9am > 9: Yes (44/17)
## wind_gust_dir in [S-NNW]:
## :...min_temp <= 21.1: No (64/10)
##     min_temp > 21.1:
##     :...cloud_3pm <= 3: No (3)
##         cloud_3pm > 3: Yes (16/5)
##
## SubTree [S80]
##
## wind_dir_3pm in [WSW-NNW]: Yes (12/2)
## wind_dir_3pm in [N-SW]:
## :...cloud_3pm <= 7: No (33/11)
##     cloud_3pm > 7: Yes (21/8)
##
## SubTree [S81]
##
## wind_speed_3pm <= 7: No (4)
## wind_speed_3pm > 7: Yes (23/7)
##
## SubTree [S82]
##
## cloud_3pm <= 6: Yes (4/1)
## cloud_3pm > 6: No (50/19)
##
```

```
## SubTree [S83]
##
## min_temp > 17.4: Yes (10)
## min_temp <= 17.4:
## :...pressure_3pm <= 1014.8: Yes (3)
##     pressure_3pm > 1014.8: No (8/2)
##
## SubTree [S84]
##
## wind_dir_9am in [WSW-NNW]: Yes (24/5)
## wind_dir_9am in [N-SW]:
## :...cloud_3pm <= 7: No (24/8)
##     cloud_3pm > 7:
##     :...rainfall > 0.6: No (2)
##         rainfall <= 0.6:
##         :...wind_gust_speed <= 33: No (7/2)
##             wind_gust_speed > 33: Yes (23/8)
##
## SubTree [S85]
##
## wind_gust_speed <= 37: No (283/80)
## wind_gust_speed > 37:
## :...wind_dir_3pm in [N-SW]: No (155/57)
##     wind_dir_3pm in [WSW-NNW]: Yes (35/9)
##
## SubTree [S86]
##
## pressure_9am <= 1017.5: No (5)
## pressure_9am > 1017.5:
## :...temp_3pm <= 18.4: Yes (12)
##     temp_3pm > 18.4:
##     :...max_temp <= 26.1: No (5)
##         max_temp > 26.1: Yes (4)
##
## SubTree [S87]
##
```

```
## wind_gust_dir in [SSE-NNW]: No (312/148)
## wind_gust_dir in [N-SE]:
## :...wind_speed_9am > 20: Yes (7)
##     wind_speed_9am <= 20:
##     :...humidity_9am <= 77: No (12/3)
##         humidity_9am > 77:
##         :...wind_speed_9am <= 6:
##             :...wind_speed_3pm <= 26: Yes (20)
##             :   wind_speed_3pm > 26: No (2)
##             wind_speed_9am > 6:
##             :...wind_gust_dir in [N-ENE]: No (31/14)
##                 wind_gust_dir in [E-SE]: Yes (37/11)
##
## SubTree [S88]
##
## humidity_9am <= 90: No (4)
## humidity_9am > 90: Yes (54/23)
##
## SubTree [S89]
##
## humidity_3pm <= 99: No (11/2)
## humidity_3pm > 99: Yes (3)
##
## SubTree [S90]
##
## wind_dir_3pm in [N-ENE]: Yes (3)
## wind_dir_3pm in [E-NNW]: No (38/11)
##
## SubTree [S91]
##
## humidity_3pm <= 82: No (2)
## humidity_3pm > 82: Yes (21/2)
##
## SubTree [S92]
##
## wind_speed_3pm <= 30: No (5)
```

```
## wind_speed_3pm > 30:
## :...humidity_9am <= 87: Yes (9/1)
##     humidity_9am > 87: No (3)
##
## SubTree [S93]
##
## wind_dir_3pm in [N-E]: No (14/3)
## wind_dir_3pm in [ESE-WNW]:
## :...rainfall <= 0.9: Yes (48/17)
##     rainfall > 0.9: No (4)
##
## SubTree [S94]
##
## wind_speed_9am <= 4: Yes (4)
## wind_speed_9am > 4:
## :...max_temp <= 25.6: No (8/1)
##     max_temp > 25.6: Yes (2)
##
## SubTree [S95]
##
## wind_gust_dir in [N-WNW]: Yes (18/2)
## wind_gust_dir in [NW-NNW]: No (2)
##
## SubTree [S96]
##
## wind_gust_dir in [N-ESE]: Yes (4)
## wind_gust_dir in [SE-NNW]: No (17/4)
##
## SubTree [S97]
##
## min_temp <= 3.4: No (10)
## min_temp > 3.4: Yes (132/63)
##
## SubTree [S98]
##
## temp_3pm <= 22.7: Yes (126/26)
```

```
## temp_3pm > 22.7: No (6/1)
##
## SubTree [S99]
##
## min_temp > 12.1: Yes (11/1)
## min_temp <= 12.1:
## :...wind_dir_3pm in [N-WSW]: No (24/5)
##     wind_dir_3pm in [W-NNW]: Yes (2)
##
## SubTree [S100]
##
## pressure_3pm > 1026.1: No (3)
## pressure_3pm <= 1026.1:
## :...wind_gust_dir in [N-ESE]: No (3)
##     wind_gust_dir in [SE-NNW]: Yes (39/7)
##
## SubTree [S101]
##
## humidity_9am <= 67: Yes (55/18)
## humidity_9am > 67:
## :...wind_gust_speed <= 37:
##     :...wind_speed_9am <= 17: No (72/12)
##     :   wind_speed_9am > 17:
##     :   :...sunshine <= 6.6: Yes (6)
##     :       sunshine > 6.6:
##     :       :...pressure_9am <= 1020.5: Yes (3)
##     :           pressure_9am > 1020.5: No (8/1)
##     wind_gust_speed > 37:
##     :...cloud_3pm > 7:
##         :...evaporation <= 2.9: Yes (8)
##         :   evaporation > 2.9:
##         :   :...wind_gust_dir in [N-S]: Yes (31/11)
##         :       wind_gust_dir in [SSW-NNW]:
##         :       :...rainfall <= 0.1: No (16/2)
##         :           rainfall > 0.1:
##         :           :...humidity_3pm <= 85: Yes (6/1)
```

```
##         :               humidity_3pm > 85: No (3)
##         cloud_3pm <= 7:
##         :...cloud_9am > 7: No (12/1)
##             cloud_9am <= 7:
##             :...cloud_9am > 5:
##                 :...cloud_9am <= 6: Yes (7)
##                 :   cloud_9am > 6:
##                 :   :...pressure_3pm <= 1017.4: Yes (5)
##                 :       pressure_3pm > 1017.4:
##                 :       :...humidity_9am > 79: No (7)
##                 :           humidity_9am <= 79:
##                 :           :...pressure_9am <= 1019.4: No (2)
##                 :               pressure_9am > 1019.4: Yes (3)
##                 cloud_9am <= 5:
##                 :...cloud_3pm > 5: No (5)
##                     cloud_3pm <= 5:
##                     :...wind_speed_9am > 20: No (9)
##                         wind_speed_9am <= 20:
##                         :...sunshine <= 6.7: Yes (4)
##                             sunshine > 6.7:
##                             :...wind_dir_3pm in [N-E]: No (7)
##                                 wind_dir_3pm in [ESE-NNW]:
##                                 :...wind_speed_9am <= 11: Yes (10/1)
##                                     wind_speed_9am > 11:
##                                     :...humidity_3pm <= 83: No (10)
##                                         humidity_3pm > 83: Yes (24/11)
##
##
## Evaluation on training data (123722 cases):
##
##      Decision Tree
##    ----------------
##    Size      Errors
##
##     845 15385(12.4%)   <<
##
```

```
##
##     (a)   (b)    <-classified as
##    ----  ----
##   93147  4474    (a): class No
##   10911 15190    (b): class Yes
##
##
##  Attribute usage:
##
##  100.00% humidity_3pm
##   90.80% sunshine
##   89.91% wind_gust_speed
##   88.29% rain_today
##   36.63% pressure_3pm
##   25.09% pressure_9am
##   19.31% cloud_3pm
##   18.42% rainfall
##   14.99% temp_9am
##   12.89% wind_dir_3pm
##   10.27% temp_3pm
##   10.27% wind_speed_9am
##    9.98% wind_speed_3pm
##    9.74% humidity_9am
##    8.49% cloud_9am
##    7.55% wind_dir_9am
##    7.17% wind_gust_dir
##    6.07% min_temp
##    4.22% evaporation
##    2.14% max_temp
##
##
## Time: 3.5 secs
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.71 C5.0 Decision Tree Performance

Here we plot the performance of the decision tree, showing a risk chart. The areas under the recall and risk curves are also reported.

```
predicted <- predict(model, ds[te, vars], type="prob")[,2]
riskchart(predicted, actual_te, risk_te)
```

An error matrix shows, clockwise from the top left, the percentages of true negatives, false positives, true positives, and false negatives.

```
predicted <- predict(model, ds[te, vars], type="class")
sum(actual_te != predicted)/length(predicted) # Overall error rate
```

```
## [1] 0.1552446
```

```
round(100*table(actual_te, predicted, dnn=c("Actual", "Predicted"))/length(predicte
```

```
##       Predicted
## Actual No Yes
##    No  74   5
##    Yes 10  10
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.72 C5.0 Rules Model

```
library(C50)
model <- C5.0(form, ds[tr, vars], rules=TRUE)
```

```
model
```

```
##
## Call:
## C5.0.formula(formula = form, data = ds[tr, vars], rules = TRUE)
##
## Rule-Based Model
## Number of samples: 123722
## Number of predictors: 20
##
## Number of Rules: 204
##
## Non-standard options: attempt to group attributes
```

```
C5imp(model)
```

```
##                 Overall
## humidity_3pm     100.00
## pressure_3pm      53.22
## wind_gust_speed   47.86
## rainfall          44.72
## wind_dir_3pm      41.80
## cloud_3pm         31.60
## temp_3pm          30.43
## sunshine          29.96
## rain_today        29.42
## wind_speed_3pm    22.57
## cloud_9am         17.70
## wind_gust_dir     16.99
## humidity_9am      16.32
## temp_9am          14.69
## min_temp          14.40
## wind_speed_9am    12.50
## wind_dir_9am       6.33
## evaporation        4.65
## pressure_9am       1.90
## max_temp           1.08
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.73 C5.0 Rules Summary

```
summary(model)
```

```
##
## Call:
## C5.0.formula(formula = form, data = ds[tr, vars], rules = TRUE)
##
##
## C5.0 [Release 2.07 GPL Edition]      Thu Jun 24 16:06:49 2021
## -------------------------------
##
## Class specified by attribute `outcome'
##
## Read 123722 cases (21 attributes) from undefined.data
##
## Rules:
##
## Rule 1: (2089/51, lift 1.2)
##  temp_9am > 6.4
##  cloud_3pm <= 5
##  pressure_3pm > 1015.1
##  humidity_3pm <= 88
##  humidity_9am <= 96
##  wind_speed_3pm <= 13
##  rainfall <= 2.3
##  min_temp <= 3.4
##  ->  class No  [0.975]
##
## Rule 2: (9862/345, lift 1.2)
##  temp_3pm > 12.4
##  cloud_3pm <= 2
##  pressure_3pm > 1013.8
##  wind_gust_speed <= 41
##  ->  class No  [0.965]
##
## Rule 3: (23, lift 1.2)
##  temp_3pm > 30.3
##  humidity_3pm > 81
```

```
##  rainfall <= 9.6
##  ->  class No  [0.960]
##
## Rule 4: (19, lift 1.2)
##  cloud_3pm > 7
##  humidity_3pm <= 66
##  wind_dir_3pm in [N-NE]
##  sunshine > 8.6
##  ->  class No  [0.952]
##
## Rule 5: (9707/462, lift 1.2)
##  temp_3pm > 18.4
##  cloud_9am <= 4
##  pressure_3pm > 1013.8
##  humidity_3pm <= 81
##  wind_gust_speed <= 41
##  rainfall <= 5.1
##  ->  class No  [0.952]
##
## Rule 6: (13718/673, lift 1.2)
##  rain_today = No
##  temp_3pm > 20.9
##  cloud_3pm <= 5
##  humidity_3pm <= 88
##  wind_gust_speed <= 33
##  sunshine > 8.3
##  ->  class No  [0.951]
##
## Rule 7: (5960/305, lift 1.2)
##  rain_today = No
##  pressure_3pm > 1008.5
##  humidity_3pm <= 81
##  wind_speed_3pm <= 15
##  wind_speed_9am > 7
##  wind_gust_speed <= 41
##  wind_gust_dir in [ENE-SSE]
```

```
##  sunshine > 0.7
##  ->  class No  [0.949]
##
## Rule 8: (5208/269, lift 1.2)
##  rain_today = No
##  pressure_3pm > 1015.1
##  humidity_3pm <= 88
##  wind_speed_3pm <= 13
##  wind_gust_speed <= 22
##  min_temp <= 12.1
##  ->  class No  [0.948]
##
## Rule 9: (18596/987, lift 1.2)
##  rain_today = No
##  temp_3pm > 17.7
##  cloud_3pm <= 5
##  humidity_3pm <= 88
##  wind_gust_speed <= 33
##  sunshine > 8.3
##  ->  class No  [0.947]
##
## Rule 10: (8879/479, lift 1.2)
##  pressure_3pm > 1021.8
##  humidity_3pm <= 88
##  wind_gust_speed <= 31
##  rainfall <= 2.3
##  ->  class No  [0.946]
##
## Rule 11: (6270/522, lift 1.2)
##  humidity_3pm <= 88
##  humidity_9am <= 95
##  wind_speed_3pm > 9
##  wind_dir_3pm in [E-NNW]
##  wind_gust_speed <= 30
##  wind_gust_dir in [SSE-NNW]
##  min_temp <= 22.2
```

```
##  ->  class No  [0.917]
##
## Rule 12: (249/22, lift 1.2)
##  temp_9am <= 6.4
##  cloud_9am <= 6
##  pressure_3pm > 1014
##  pressure_3pm <= 1015.3
##  humidity_3pm > 66
##  humidity_3pm <= 94
##  humidity_9am > 94
##  wind_dir_9am in [N-W]
##  ->  class No  [0.908]
##
## Rule 13: (1476/141, lift 1.1)
##  humidity_3pm <= 81
##  wind_speed_3pm > 26
##  wind_gust_speed <= 41
##  wind_gust_dir in [N-SE]
##  ->  class No  [0.904]
##
## Rule 14: (1107/110, lift 1.1)
##  temp_3pm > 17
##  humidity_3pm <= 66
##  wind_speed_9am > 26
##  wind_dir_3pm in [N-W]
##  wind_gust_speed > 46
##  wind_gust_speed <= 59
##  wind_gust_dir in [SSE-NNW]
##  ->  class No  [0.900]
##
## Rule 15: (42320/4253, lift 1.1)
##  pressure_3pm > 1015.4
##  humidity_3pm <= 93
##  wind_dir_3pm in [N-WNW]
##  rainfall <= 2.3
##  ->  class No  [0.899]
```

```
##
## Rule 16: (1251/138, lift 1.1)
##  temp_9am > 4.4
##  humidity_3pm <= 81
##  wind_speed_3pm > 24
##  wind_gust_speed > 41
##  wind_gust_speed <= 43
##  ->  class No  [0.889]
##
## Rule 17: (337/39, lift 1.1)
##  humidity_3pm <= 66
##  humidity_9am > 93
##  evaporation <= 0.9
##  ->  class No  [0.882]
##
## Rule 18: (4772/678, lift 1.1)
##  temp_9am <= 6.4
##  pressure_3pm > 1014
##  ->  class No  [0.858]
##
## Rule 19: (114316/18928, lift 1.1)
##  humidity_3pm <= 81
##  ->  class No  [0.834]
##
## Rule 20: (24, lift 4.6)
##  temp_9am > 6.4
##  pressure_3pm > 1015.4
##  humidity_3pm > 88
##  wind_dir_3pm in [NNE-WNW]
##  min_temp <= 2.9
##  ->  class Yes  [0.962]
##
## Rule 21: (22, lift 4.5)
##  cloud_9am > 1
##  pressure_3pm > 1013.8
##  humidity_3pm > 66
```

```
##  humidity_3pm <= 77
##  humidity_9am > 77
##  wind_speed_3pm <= 26
##  wind_speed_9am <= 6
##  wind_gust_speed > 28
##  wind_gust_speed <= 41
##  wind_gust_dir in [N-SE]
##  evaporation > 1.3
##  rainfall > 5.1
##  ->  class Yes  [0.958]
##
## Rule 22: (45/1, lift 4.5)
##  rain_today = No
##  pressure_3pm <= 1013.7
##  humidity_9am > 80
##  wind_gust_speed > 61
##  sunshine <= 7.7
##  evaporation <= 4.4
##  ->  class Yes  [0.957]
##
## Rule 23: (19, lift 4.5)
##  rain_today = Yes
##  cloud_9am > 5
##  pressure_3pm <= 1000
##  humidity_3pm > 50
##  humidity_3pm <= 66
##  wind_gust_speed <= 46
##  sunshine <= 9
##  rainfall <= 27.6
##  min_temp > 6.2
##  ->  class Yes  [0.952]
##
## Rule 24: (18, lift 4.5)
##  temp_3pm > 17.4
##  pressure_3pm > 1014.3
##  humidity_9am <= 82
```

```
##  wind_dir_3pm in [NE-W]
##  rainfall > 6.5
##  max_temp <= 14
##  min_temp > 3
##  ->  class Yes  [0.950]
##
## Rule 25: (17, lift 4.5)
##  pressure_3pm <= 1017.1
##  humidity_3pm > 14
##  wind_speed_3pm <= 33
##  wind_dir_3pm in [N-NNE]
##  wind_gust_speed > 57
##  wind_gust_speed <= 63
##  sunshine > 8.3
##  sunshine <= 10.6
##  evaporation > 5.9
##  rainfall <= 0.1
##  ->  class Yes  [0.947]
##
## Rule 26: (17, lift 4.5)
##  rain_today = No
##  cloud_3pm > 5
##  pressure_3pm <= 1007.6
##  humidity_3pm > 50
##  humidity_3pm <= 66
##  wind_speed_3pm > 28
##  wind_dir_3pm in [E-NNW]
##  wind_gust_speed > 50
##  wind_gust_speed <= 65
##  sunshine > 7.7
##  sunshine <= 10.6
##  min_temp <= 25.1
##  ->  class Yes  [0.947]
##
## Rule 27: (17, lift 4.5)
##  rain_today = Yes
```

```
##  pressure_3pm <= 1009.3
##  humidity_3pm > 50
##  humidity_3pm <= 66
##  humidity_9am <= 86
##  wind_dir_9am in [N-NNE]
##  wind_gust_speed > 39
##  wind_gust_speed <= 46
##  sunshine <= 9
##  ->  class Yes  [0.947]
##
## Rule 28: (17, lift 4.5)
##  rain_today = Yes
##  cloud_3pm > 4
##  cloud_3pm <= 6
##  cloud_9am > 3
##  pressure_3pm > 1018.6
##  humidity_3pm > 48
##  humidity_3pm <= 66
##  humidity_9am > 68
##  wind_dir_9am in [E-W]
##  wind_gust_speed > 46
##  wind_gust_dir in [E-SSW]
##  sunshine <= 7
##  rainfall <= 20.2
##  ->  class Yes  [0.947]
##
## Rule 29: (35/1, lift 4.5)
##  cloud_9am > 1
##  pressure_3pm <= 1012.2
##  pressure_9am > 1011.5
##  humidity_3pm > 51
##  humidity_3pm <= 66
##  wind_speed_3pm <= 33
##  wind_dir_3pm in [SSE-NNW]
##  wind_gust_speed > 50
##  wind_gust_speed <= 65
```

```
##  wind_gust_dir in [WSW-NNW]
##  sunshine > 7.7
##  sunshine <= 10.6
##  rainfall <= 0.3
##  ->  class Yes  [0.946]
##
## Rule 30: (16, lift 4.5)
##  rain_today = No
##  temp_9am > 6.4
##  cloud_3pm <= 7
##  cloud_9am > 5
##  cloud_9am <= 6
##  pressure_3pm > 1015.1
##  humidity_3pm > 81
##  humidity_3pm <= 88
##  wind_speed_3pm > 13
##  wind_dir_9am in [N-WNW]
##  ->  class Yes  [0.944]
##
## Rule 31: (15, lift 4.5)
##  rain_today = No
##  pressure_3pm <= 1017.2
##  humidity_3pm > 66
##  humidity_3pm <= 81
##  humidity_9am > 82
##  humidity_9am <= 94
##  wind_speed_9am > 9
##  wind_dir_3pm in [S-NW]
##  wind_gust_speed <= 41
##  sunshine <= 0.7
##  ->  class Yes  [0.941]
##
## Rule 32: (131/7, lift 4.5)
##  pressure_3pm <= 1006.1
##  humidity_3pm > 22
##  wind_gust_speed > 54
```

```
##  sunshine <= 7.7
##  min_temp > 22.7
##  ->  class Yes  [0.940]
##
## Rule 33: (14, lift 4.4)
##  rain_today = Yes
##  wind_dir_9am in [ENE-SE]
##  wind_gust_speed > 87
##  sunshine <= 11
##  ->  class Yes  [0.938]
##
## Rule 34: (109/6, lift 4.4)
##  pressure_3pm <= 1011.6
##  humidity_3pm > 50
##  wind_speed_3pm <= 22
##  wind_gust_speed > 50
##  wind_gust_dir in [N-SW]
##  sunshine > 7.7
##  max_temp <= 20.6
##  ->  class Yes  [0.937]
##
## Rule 35: (59/3, lift 4.4)
##  rain_today = Yes
##  pressure_3pm <= 1000
##  humidity_3pm > 50
##  wind_gust_speed > 31
##  wind_gust_speed <= 46
##  sunshine <= 9
##  rainfall <= 27.6
##  min_temp > 6.2
##  min_temp <= 13.3
##  ->  class Yes  [0.934]
##
## Rule 36: (13, lift 4.4)
##  rain_today = Yes
##  pressure_9am <= 1017.8
```

```
##  humidity_3pm > 45
##  humidity_3pm <= 48
##  humidity_9am > 71
##  humidity_9am <= 80
##  wind_speed_9am <= 15
##  wind_gust_speed > 46
##  sunshine > 6.7
##  ->  class Yes  [0.933]
##
## Rule 37: (13, lift 4.4)
##  rain_today = Yes
##  pressure_3pm > 1012.5
##  humidity_3pm > 48
##  humidity_3pm <= 66
##  humidity_9am > 68
##  wind_dir_9am in [N-ENE]
##  wind_gust_speed > 46
##  wind_gust_dir in [NE-ENE]
##  sunshine <= 9
##  rainfall <= 20.2
##  ->  class Yes  [0.933]
##
## Rule 38: (13, lift 4.4)
##  rain_today = No
##  cloud_9am > 5
##  pressure_3pm <= 1008.5
##  humidity_3pm > 66
##  humidity_3pm <= 72
##  sunshine > 0.7
##  sunshine <= 5
##  rainfall > 0.5
##  ->  class Yes  [0.933]
##
## Rule 39: (41/2, lift 4.4)
##  temp_3pm > 26.3
##  humidity_3pm > 66
```

```
##  humidity_3pm <= 81
##  humidity_9am > 86
##  wind_gust_speed > 54
##  ->  class Yes  [0.930]
##
## Rule 40: (12, lift 4.4)
##  rain_today = No
##  cloud_3pm > 7
##  pressure_9am <= 1012.2
##  humidity_3pm > 49
##  humidity_3pm <= 66
##  wind_speed_9am <= 7
##  sunshine > 7.7
##  max_temp <= 25.3
##  ->  class Yes  [0.929]
##
## Rule 41: (12, lift 4.4)
##  temp_3pm > 29.1
##  cloud_9am <= 5
##  humidity_3pm > 24
##  humidity_3pm <= 50
##  humidity_9am <= 65
##  wind_speed_3pm <= 13
##  wind_dir_3pm in [ENE-WNW]
##  wind_dir_9am in [N-E]
##  wind_gust_speed > 50
##  wind_gust_dir in [S-NNW]
##  sunshine > 7.7
##  ->  class Yes  [0.929]
##
## Rule 42: (12, lift 4.4)
##  rain_today = No
##  temp_3pm > 26.1
##  cloud_9am > 1
##  pressure_3pm > 1002.2
##  pressure_9am <= 1014.4
```

```
##  humidity_3pm > 51
##  humidity_3pm <= 66
##  wind_gust_speed > 50
##  wind_gust_dir in [WSW-NW]
##  sunshine > 7.7
##  min_temp <= 25.1
##  ->  class Yes  [0.929]
##
## Rule 43: (26/1, lift 4.4)
##  rain_today = No
##  temp_9am > 17.4
##  pressure_3pm > 1012.2
##  pressure_9am > 1014.7
##  humidity_3pm > 50
##  wind_gust_speed > 50
##  wind_gust_dir in [WSW-W]
##  sunshine > 7.7
##  sunshine <= 8.5
##  ->  class Yes  [0.929]
##
## Rule 44: (12, lift 4.4)
##  rain_today = Yes
##  temp_3pm > 12.4
##  temp_3pm <= 18.4
##  cloud_3pm > 2
##  cloud_9am > 1
##  cloud_9am <= 4
##  pressure_9am > 1017.5
##  humidity_3pm > 66
##  humidity_3pm <= 77
##  wind_gust_speed > 28
##  wind_gust_speed <= 41
##  evaporation > 1.3
##  rainfall <= 5.1
##  ->  class Yes  [0.929]
##
```

```
## Rule 45: (81/5, lift 4.4)
##  rain_today = Yes
##  cloud_9am > 3
##  pressure_9am <= 996.6
##  wind_gust_speed > 46
##  sunshine <= 6.7
##  ->  class Yes  [0.928]
##
## Rule 46: (25/1, lift 4.4)
##  temp_9am <= 6.4
##  pressure_3pm <= 1015.3
##  humidity_3pm > 81
##  humidity_3pm <= 94
##  humidity_9am <= 94
##  rainfall > 2.7
##  ->  class Yes  [0.926]
##
## Rule 47: (11, lift 4.4)
##  rain_today = No
##  cloud_9am > 4
##  cloud_9am <= 5
##  pressure_3pm <= 1013.7
##  humidity_3pm <= 66
##  wind_speed_3pm > 15
##  wind_speed_9am <= 7
##  wind_dir_9am in [E-NNW]
##  wind_gust_speed > 41
##  wind_gust_speed <= 54
##  sunshine <= 7.7
##  ->  class Yes  [0.923]
##
## Rule 48: (11, lift 4.4)
##  cloud_9am <= 5
##  pressure_3pm <= 1002
##  humidity_3pm > 49
##  humidity_3pm <= 66
```

```
##  wind_speed_3pm <= 20
##  wind_gust_speed > 37
##  wind_gust_speed <= 46
##  wind_gust_dir in [WSW-NNW]
##  sunshine > 7.7
##  rainfall <= 0.3
##  ->  class Yes  [0.923]
##
## Rule 49: (11, lift 4.4)
##  rain_today = No
##  pressure_3pm <= 1007.4
##  humidity_3pm > 14
##  humidity_3pm <= 50
##  wind_speed_3pm > 41
##  wind_dir_3pm in [N-NNE]
##  sunshine > 8.3
##  sunshine <= 9.9
##  evaporation > 4.3
##  ->  class Yes  [0.923]
##
## Rule 50: (11, lift 4.4)
##  rain_today = Yes
##  pressure_3pm <= 1016
##  humidity_3pm > 50
##  humidity_3pm <= 66
##  wind_dir_9am in [N-NNE]
##  wind_gust_speed <= 31
##  sunshine > 3.4
##  sunshine <= 5.2
##  ->  class Yes  [0.923]
##
## Rule 51: (11, lift 4.4)
##  rain_today = Yes
##  cloud_9am <= 3
##  pressure_3pm <= 1014.3
##  humidity_3pm > 50
```

```
##  humidity_9am <= 86
##  wind_dir_9am in [N-NNE]
##  wind_gust_speed > 35
##  wind_gust_speed <= 39
##  sunshine <= 9
##  ->  class Yes  [0.923]
##
## Rule 52: (11, lift 4.4)
##  rain_today = Yes
##  pressure_3pm <= 1014.3
##  pressure_9am > 1011.2
##  humidity_3pm > 50
##  humidity_3pm <= 66
##  humidity_9am <= 86
##  wind_dir_9am in [N-NNE]
##  wind_gust_speed > 39
##  wind_gust_speed <= 46
##  sunshine <= 9
##  rainfall <= 6.8
##  ->  class Yes  [0.923]
##
## Rule 53: (10, lift 4.3)
##  rain_today = No
##  cloud_3pm > 5
##  pressure_3pm > 1008.5
##  humidity_3pm > 66
##  humidity_3pm <= 81
##  wind_speed_3pm <= 15
##  wind_gust_speed > 28
##  wind_gust_speed <= 41
##  wind_gust_dir in [ENE-SSE]
##  max_temp > 28.6
##  ->  class Yes  [0.917]
##
## Rule 54: (21/1, lift 4.3)
##  rain_today = Yes
```

```
##  cloud_3pm <= 6
##  cloud_9am > 3
##  humidity_3pm > 50
##  humidity_3pm <= 52
##  humidity_9am > 68
##  wind_dir_9am in [SSE-NNW]
##  wind_gust_speed > 46
##  sunshine <= 7
##  rainfall <= 20.2
##  ->  class Yes  [0.913]
##
## Rule 55: (1044/91, lift 4.3)
##  temp_3pm > -1.2
##  humidity_3pm > 94
##  rainfall > 3.3
##  ->  class Yes  [0.912]
##
## Rule 56: (155/13, lift 4.3)
##  humidity_3pm > 99
##  wind_gust_speed > 35
##  wind_gust_dir in [N-WNW]
##  rainfall > 0.1
##  ->  class Yes  [0.911]
##
## Rule 57: (9, lift 4.3)
##  temp_9am > 28.5
##  cloud_3pm <= 7
##  humidity_3pm > 14
##  humidity_3pm <= 66
##  wind_dir_3pm = N
##  wind_gust_speed > 50
##  wind_gust_speed <= 74
##  wind_gust_dir in [SSW-NNW]
##  sunshine > 7.7
##  sunshine <= 10.6
##  ->  class Yes  [0.909]
```

```
##
## Rule 58: (9, lift 4.3)
##  rain_today = No
##  humidity_3pm > 24
##  humidity_3pm <= 50
##  wind_speed_3pm <= 13
##  wind_dir_3pm in [ENE-WNW]
##  wind_gust_speed > 50
##  sunshine <= 10.6
##  evaporation > 12.2
##  ->  class Yes  [0.909]
##
## Rule 59: (9, lift 4.3)
##  rain_today = Yes
##  cloud_3pm > 6
##  pressure_3pm > 1021.7
##  humidity_3pm > 50
##  humidity_3pm <= 66
##  humidity_9am > 68
##  wind_speed_9am <= 15
##  wind_gust_speed > 31
##  rainfall <= 6.5
##  max_temp > 21.3
##  ->  class Yes  [0.909]
##
## Rule 60: (9, lift 4.3)
##  rain_today = Yes
##  pressure_3pm <= 1002.6
##  humidity_3pm <= 48
##  wind_dir_3pm in [N-NNE]
##  wind_gust_speed > 46
##  wind_gust_speed <= 72
##  sunshine <= 9
##  ->  class Yes  [0.909]
##
## Rule 61: (20/1, lift 4.3)
```

```
##  rain_today = No
##  cloud_3pm > 3
##  pressure_3pm <= 1013.7
##  humidity_3pm > 27
##  wind_speed_3pm > 15
##  wind_speed_9am <= 7
##  wind_dir_3pm in [WNW-NNW]
##  wind_dir_9am in [E-NNW]
##  wind_gust_speed > 41
##  wind_gust_speed <= 54
##  sunshine <= 7.7
##  ->  class Yes  [0.909]
##
## Rule 62: (9, lift 4.3)
##  rain_today = Yes
##  cloud_3pm > 2
##  cloud_9am > 1
##  cloud_9am <= 4
##  pressure_9am > 1017.5
##  humidity_3pm > 66
##  max_temp > 26.1
##  ->  class Yes  [0.909]
##
## Rule 63: (31/2, lift 4.3)
##  rain_today = No
##  cloud_3pm > 5
##  humidity_3pm > 81
##  humidity_9am <= 96
##  wind_gust_speed <= 22
##  min_temp > 12.1
##  ->  class Yes  [0.909]
##
## Rule 64: (894/87, lift 4.3)
##  temp_9am > 6.4
##  cloud_3pm > 7
##  humidity_3pm > 81
```

```
##  wind_gust_speed > 37
##  wind_gust_dir in [SSW-NNW]
##  rainfall > 0.1
##  ->  class Yes  [0.902]
##
## Rule 65: (179/17, lift 4.3)
##  rain_today = Yes
##  cloud_9am > 7
##  pressure_3pm <= 1012.5
##  humidity_9am > 77
##  wind_speed_9am > 28
##  wind_gust_speed > 59
##  sunshine <= 9
##  ->  class Yes  [0.901]
##
## Rule 66: (18/1, lift 4.3)
##  cloud_3pm > 7
##  cloud_9am <= 5
##  pressure_3pm <= 1017.1
##  humidity_3pm > 25
##  humidity_3pm <= 50
##  wind_dir_3pm in [N-NE]
##  wind_gust_speed > 50
##  sunshine > 7.7
##  ->  class Yes  [0.900]
##
## Rule 67: (18/1, lift 4.3)
##  rain_today = No
##  temp_3pm > 32.6
##  humidity_3pm > 21
##  humidity_3pm <= 50
##  wind_dir_3pm in [NW-NNW]
##  wind_gust_speed > 70
##  sunshine > 7.7
##  sunshine <= 10.6
##  ->  class Yes  [0.900]
```

```
##
## Rule 68: (8, lift 4.3)
##  rain_today = Yes
##  pressure_3pm <= 1012.5
##  pressure_9am > 1012.4
##  humidity_3pm <= 50
##  wind_dir_9am in [N-W]
##  wind_gust_speed <= 37
##  sunshine <= 6.4
##  min_temp <= 18.1
##  ->  class Yes  [0.900]
##
## Rule 69: (8, lift 4.3)
##  rain_today = Yes
##  pressure_3pm > 1014
##  pressure_3pm <= 1014.3
##  humidity_3pm > 50
##  humidity_3pm <= 66
##  wind_gust_speed > 44
##  wind_gust_speed <= 46
##  sunshine <= 9
##  ->  class Yes  [0.900]
##
## Rule 70: (88/8, lift 4.3)
##  rain_today = Yes
##  cloud_9am <= 4
##  humidity_3pm > 66
##  wind_speed_9am <= 26
##  wind_dir_3pm in [SSW-NNW]
##  wind_gust_speed > 54
##  ->  class Yes  [0.900]
##
## Rule 71: (28/2, lift 4.3)
##  temp_9am <= 6.4
##  cloud_3pm <= 7
##  pressure_9am <= 1018.9
```

```
##  humidity_3pm > 81
##  sunshine <= 7.7
##  ->  class Yes  [0.900]
##
## Rule 72: (17/1, lift 4.2)
##  rain_today = No
##  temp_9am > 15.2
##  humidity_3pm <= 50
##  humidity_9am > 65
##  wind_speed_3pm <= 13
##  wind_dir_3pm in [ENE-WNW]
##  wind_gust_speed > 61
##  sunshine <= 10.6
##  ->  class Yes  [0.895]
##
## Rule 73: (634/66, lift 4.2)
##  cloud_3pm > 7
##  pressure_3pm <= 1017.2
##  pressure_9am <= 1019.3
##  humidity_3pm > 66
##  wind_dir_3pm in [N-SSE]
##  wind_gust_dir in [N-NW]
##  sunshine <= 0.7
##  ->  class Yes  [0.895]
##
## Rule 74: (36/3, lift 4.2)
##  rain_today = No
##  pressure_3pm <= 1017.2
##  humidity_3pm > 66
##  humidity_9am <= 82
##  wind_speed_3pm <= 11
##  wind_dir_3pm in [S-NNW]
##  sunshine <= 0.7
##  ->  class Yes  [0.895]
##
## Rule 75: (776/82, lift 4.2)
```

```
##  temp_9am > 6.4
##  cloud_3pm > 7
##  humidity_3pm > 81
##  humidity_9am <= 96
##  wind_speed_9am <= 31
##  wind_dir_9am in [N-WNW]
##  wind_gust_speed > 37
##  wind_gust_dir in [N-S]
##  ->  class Yes  [0.893]
##
## Rule 76: (1144/124, lift 4.2)
##  temp_9am > 6.4
##  cloud_9am > 2
##  humidity_3pm > 81
##  wind_speed_3pm <= 43
##  wind_speed_9am <= 31
##  wind_gust_speed > 56
##  ->  class Yes  [0.891]
##
## Rule 77: (7, lift 4.2)
##  rain_today = No
##  cloud_3pm <= 4
##  pressure_9am <= 1012.2
##  humidity_3pm > 49
##  wind_gust_speed > 37
##  wind_gust_speed <= 50
##  wind_gust_dir in [N-ENE]
##  sunshine <= 11.6
##  rainfall > 0.3
##  ->  class Yes  [0.889]
##
## Rule 78: (7, lift 4.2)
##  rain_today = No
##  temp_9am > 25.7
##  humidity_3pm > 21
##  wind_dir_3pm = NNW
```

```
##  wind_gust_speed > 50
##  wind_gust_speed <= 70
##  wind_gust_dir in [NNE-ESE]
##  sunshine <= 10.6
##  evaporation <= 6.7
##  ->  class Yes  [0.889]
##
## Rule 79: (7, lift 4.2)
##  rain_today = Yes
##  cloud_3pm <= 4
##  pressure_3pm <= 1016
##  humidity_3pm > 61
##  humidity_3pm <= 66
##  wind_gust_speed <= 31
##  sunshine <= 5.8
##  ->  class Yes  [0.889]
##
## Rule 80: (7, lift 4.2)
##  rain_today = Yes
##  pressure_3pm > 1014.3
##  humidity_3pm > 50
##  humidity_3pm <= 51
##  wind_speed_9am <= 15
##  wind_dir_3pm in [WSW-NNW]
##  wind_gust_speed > 41
##  wind_gust_speed <= 46
##  rainfall <= 6.5
##  ->  class Yes  [0.889]
##
## Rule 81: (7, lift 4.2)
##  humidity_3pm > 50
##  humidity_3pm <= 66
##  wind_speed_9am > 31
##  wind_gust_speed <= 46
##  sunshine <= 5.9
##  rainfall > 6.5
```

```
##  ->  class Yes  [0.889]
##
## Rule 82: (264/29, lift 4.2)
##  humidity_3pm > 94
##  wind_gust_dir in [NW-NNW]
##  ->  class Yes  [0.887]
##
## Rule 83: (2175/252, lift 4.2)
##  rain_today = Yes
##  cloud_3pm > 6
##  pressure_3pm <= 1013.8
##  humidity_3pm > 75
##  ->  class Yes  [0.884]
##
## Rule 84: (1396/162, lift 4.2)
##  cloud_9am > 6
##  pressure_3pm <= 1015.3
##  humidity_3pm > 81
##  wind_dir_9am in [N-SE]
##  wind_gust_speed > 19
##  ->  class Yes  [0.883]
##
## Rule 85: (1250/145, lift 4.2)
##  temp_9am > 6.4
##  cloud_3pm > 5
##  cloud_9am > 2
##  humidity_3pm > 81
##  humidity_9am <= 96
##  wind_speed_3pm <= 13
##  wind_gust_speed > 22
##  ->  class Yes  [0.883]
##
## Rule 86: (3198/374, lift 4.2)
##  temp_3pm <= 30.3
##  temp_9am > 6.4
##  humidity_3pm > 88
```

```
##  humidity_9am <= 96
##  ->  class Yes  [0.883]
##
## Rule 87: (1225/146, lift 4.2)
##  temp_9am > 4.4
##  cloud_9am > 4
##  humidity_3pm > 66
##  wind_gust_speed > 63
##  rainfall > 0.2
##  ->  class Yes  [0.880]
##
## Rule 88: (96/11, lift 4.2)
##  rain_today = Yes
##  humidity_9am > 49
##  wind_dir_3pm in [N-ESE]
##  wind_gust_speed > 72
##  sunshine <= 9
##  ->  class Yes  [0.878]
##
## Rule 89: (137/16, lift 4.2)
##  cloud_3pm > 5
##  pressure_3pm <= 1017.2
##  humidity_3pm > 66
##  wind_dir_3pm in [N-NE]
##  wind_gust_speed <= 41
##  wind_gust_dir in [N-NW]
##  sunshine <= 0.7
##  ->  class Yes  [0.878]
##
## Rule 90: (14/1, lift 4.1)
##  cloud_9am <= 5
##  pressure_9am <= 1012.2
##  humidity_3pm > 49
##  humidity_3pm <= 66
##  wind_gust_speed > 37
##  wind_gust_speed <= 50
```

```
##  wind_gust_dir in [WSW-NNW]
##  sunshine > 7.7
##  sunshine <= 11.6
##  evaporation <= 4.5
##  rainfall <= 0.3
##  min_temp > 8.9
##  ->  class Yes  [0.875]
##
## Rule 91: (14/1, lift 4.1)
##  cloud_9am > 7
##  pressure_3pm > 1014.3
##  humidity_3pm > 64
##  humidity_3pm <= 66
##  humidity_9am > 82
##  wind_gust_speed > 31
##  wind_gust_speed <= 46
##  rainfall > 6.5
##  ->  class Yes  [0.875]
##
## Rule 92: (14/1, lift 4.1)
##  rain_today = Yes
##  pressure_9am <= 1017.8
##  humidity_3pm > 45
##  humidity_3pm <= 48
##  humidity_9am <= 80
##  wind_speed_9am <= 15
##  wind_dir_3pm in [SE-NW]
##  wind_dir_9am in [WSW-NW]
##  wind_gust_speed > 46
##  ->  class Yes  [0.875]
##
## Rule 93: (6, lift 4.1)
##  rain_today = Yes
##  cloud_3pm <= 1
##  humidity_3pm <= 48
##  wind_dir_3pm = NNW
```

```
##  wind_gust_speed > 46
##  ->  class Yes  [0.875]
##
## Rule 94: (14/1, lift 4.1)
##  rain_today = Yes
##  temp_9am > 6.3
##  cloud_3pm > 4
##  cloud_9am <= 6
##  pressure_3pm > 1019.3
##  pressure_3pm <= 1020.2
##  humidity_3pm <= 66
##  humidity_9am > 68
##  wind_dir_9am in [SE-WNW]
##  wind_gust_speed > 46
##  sunshine <= 9
##  ->  class Yes  [0.875]
##
## Rule 95: (14/1, lift 4.1)
##  rain_today = Yes
##  temp_9am > 28.5
##  cloud_9am <= 4
##  pressure_9am <= 1014
##  humidity_3pm <= 66
##  wind_dir_9am in [ENE-SE]
##  sunshine > 9
##  sunshine <= 11
##  ->  class Yes  [0.875]
##
## Rule 96: (6, lift 4.1)
##  rain_today = Yes
##  cloud_3pm > 6
##  pressure_9am <= 1014
##  wind_speed_3pm <= 13
##  sunshine > 9
##  evaporation > 6.8
##  ->  class Yes  [0.875]
```

```
##
## Rule 97: (62/7, lift 4.1)
##  cloud_3pm <= 6
##  humidity_3pm > 66
##  humidity_3pm <= 81
##  humidity_9am <= 86
##  wind_speed_9am > 2
##  wind_speed_9am <= 20
##  wind_dir_3pm in [SSW-NNW]
##  wind_gust_speed > 54
##  sunshine <= 8.6
##  rainfall <= 0.2
##  ->  class Yes  [0.875]
##
## Rule 98: (461/57, lift 4.1)
##  temp_9am > 6.4
##  humidity_3pm > 81
##  wind_speed_9am > 31
##  ->  class Yes  [0.875]
##
## Rule 99: (29/3, lift 4.1)
##  rain_today = No
##  humidity_3pm > 21
##  humidity_3pm <= 66
##  humidity_9am > 54
##  wind_speed_9am > 15
##  wind_dir_3pm = NW
##  wind_gust_speed > 70
##  sunshine > 7.7
##  evaporation <= 6.5
##  ->  class Yes  [0.871]
##
## Rule 100: (957/123, lift 4.1)
##  rain_today = Yes
##  cloud_3pm > 3
##  pressure_3pm <= 1012.5
```

```
##  humidity_3pm > 49
##  wind_speed_9am <= 28
##  wind_gust_speed > 59
##  sunshine <= 9
##  ->  class Yes  [0.871]
##
## Rule 101: (21/2, lift 4.1)
##  cloud_3pm <= 4
##  cloud_9am > 3
##  pressure_3pm > 1012.5
##  pressure_9am <= 1015.1
##  humidity_3pm > 48
##  humidity_9am > 68
##  wind_gust_speed > 46
##  evaporation > 4.3
##  rainfall > 3.6
##  ->  class Yes  [0.870]
##
## Rule 102: (603/81, lift 4.1)
##  pressure_3pm <= 1006.1
##  humidity_3pm > 22
##  wind_speed_9am <= 46
##  wind_gust_speed > 54
##  sunshine <= 3.9
##  ->  class Yes  [0.864]
##
## Rule 103: (49/6, lift 4.1)
##  pressure_3pm <= 1012.2
##  pressure_9am > 1014.4
##  humidity_3pm > 50
##  humidity_3pm <= 66
##  wind_gust_speed > 50
##  wind_gust_dir in [WSW-NNW]
##  sunshine > 7.7
##  ->  class Yes  [0.863]
##
```

```
## Rule 104: (908/124, lift 4.1)
##  rain_today = Yes
##  temp_3pm <= 18.1
##  cloud_3pm > 2
##  cloud_3pm <= 5
##  cloud_9am > 1
##  pressure_3pm <= 1013.8
##  humidity_3pm > 66
##  wind_gust_speed > 30
##  sunshine <= 9.4
##  ->  class Yes  [0.863]
##
## Rule 105: (27/3, lift 4.1)
##  cloud_3pm > 5
##  humidity_3pm > 68
##  humidity_3pm <= 81
##  wind_speed_9am <= 7
##  wind_gust_speed > 41
##  wind_gust_speed <= 54
##  sunshine > 7.5
##  rainfall <= 0.2
##  min_temp > 0.3
##  ->  class Yes  [0.862]
##
## Rule 106: (321/44, lift 4.1)
##  temp_9am > 4.4
##  humidity_3pm > 66
##  humidity_9am > 86
##  wind_speed_3pm > 26
##  wind_speed_9am <= 17
##  wind_gust_speed > 43
##  rainfall > 0.2
##  ->  class Yes  [0.861]
##
## Rule 107: (41/5, lift 4.1)
##  rain_today = No
```

```
##  cloud_3pm <= 6
##  pressure_3pm <= 1016
##  humidity_3pm > 21
##  wind_dir_3pm = NNW
##  wind_gust_speed > 70
##  sunshine > 8.1
##  evaporation <= 6.5
##  min_temp <= 19.4
##  ->  class Yes  [0.860]
##
## Rule 108: (378/53, lift 4.1)
##  rain_today = Yes
##  cloud_3pm > 6
##  pressure_9am <= 1014
##  wind_speed_3pm > 13
##  wind_gust_speed > 70
##  ->  class Yes  [0.858]
##
## Rule 109: (19/2, lift 4.1)
##  rain_today = No
##  pressure_3pm > 1013.7
##  pressure_3pm <= 1015.4
##  humidity_3pm <= 66
##  wind_speed_3pm <= 37
##  wind_gust_speed > 46
##  wind_gust_dir in [W-NNW]
##  sunshine > 4.7
##  sunshine <= 7.7
##  max_temp <= 20.1
##  ->  class Yes  [0.857]
##
## Rule 110: (12/1, lift 4.1)
##  rain_today = No
##  pressure_9am <= 1010.4
##  humidity_3pm > 49
##  humidity_3pm <= 66
```

```
##  wind_dir_9am in [N-NNE]
##  wind_gust_speed > 46
##  wind_gust_speed <= 50
##  wind_gust_dir in [N-SSE]
##  sunshine > 7.7
##  ->  class Yes  [0.857]
##
## Rule 111: (19/2, lift 4.1)
##  rain_today = No
##  humidity_3pm > 50
##  humidity_3pm <= 66
##  humidity_9am > 71
##  wind_speed_9am <= 2
##  wind_gust_speed > 52
##  wind_gust_dir in [N-SW]
##  ->  class Yes  [0.857]
##
## Rule 112: (12/1, lift 4.1)
##  rain_today = Yes
##  cloud_9am > 6
##  pressure_9am > 1021.1
##  humidity_3pm <= 66
##  wind_gust_speed <= 31
##  sunshine <= 9
##  min_temp > 18.9
##  ->  class Yes  [0.857]
##
## Rule 113: (12/1, lift 4.1)
##  rain_today = Yes
##  pressure_3pm <= 1014.3
##  humidity_3pm > 50
##  humidity_3pm <= 66
##  wind_dir_9am in [NE-ESE]
##  wind_gust_speed > 44
##  wind_gust_speed <= 46
##  ->  class Yes  [0.857]
```

```
##
## Rule 114: (5, lift 4.1)
##  pressure_3pm > 1021.7
##  humidity_3pm <= 66
##  wind_speed_9am <= 2
##  wind_gust_speed > 31
##  rainfall > 4.7
##  rainfall <= 6.5
##  ->  class Yes  [0.857]
##
## Rule 115: (143/20, lift 4.1)
##  cloud_3pm > 7
##  pressure_3pm <= 1017.1
##  humidity_3pm > 14
##  wind_dir_3pm in [N-NE]
##  wind_gust_speed > 54
##  sunshine > 7.7
##  ->  class Yes  [0.855]
##
## Rule 116: (281/40, lift 4.1)
##  rain_today = Yes
##  cloud_3pm > 6
##  cloud_9am > 3
##  humidity_3pm > 48
##  humidity_9am > 68
##  wind_speed_3pm > 37
##  wind_gust_speed > 46
##  ->  class Yes  [0.855]
##
## Rule 117: (295/42, lift 4.1)
##  temp_9am > 6.3
##  cloud_9am > 3
##  pressure_3pm > 1012.5
##  humidity_3pm > 48
##  humidity_9am > 68
##  wind_speed_9am <= 26
```

```
##  wind_gust_speed > 46
##  rainfall > 20.2
##  ->  class Yes  [0.855]
##
## Rule 118: (32/4, lift 4.0)
##  cloud_9am > 1
##  pressure_3pm > 1002.2
##  pressure_3pm <= 1012.2
##  pressure_9am <= 1014.4
##  humidity_3pm > 50
##  wind_dir_3pm in [SSW-NNW]
##  wind_gust_speed > 50
##  wind_gust_speed <= 65
##  wind_gust_dir in [WSW-NNW]
##  sunshine > 7.7
##  rainfall > 0.1
##  rainfall <= 0.3
##  ->  class Yes  [0.853]
##
## Rule 119: (214/31, lift 4.0)
##  cloud_3pm > 5
##  pressure_3pm <= 1011.6
##  humidity_3pm > 50
##  wind_speed_3pm <= 28
##  wind_speed_9am > 2
##  wind_gust_speed > 50
##  wind_gust_dir in [N-SW]
##  sunshine > 7.7
##  sunshine <= 10.6
##  ->  class Yes  [0.852]
##
## Rule 120: (2303/341, lift 4.0)
##  cloud_3pm > 5
##  pressure_3pm <= 1008.5
##  humidity_3pm > 72
##  ->  class Yes  [0.852]
```

```
##
## Rule 121: (125/18, lift 4.0)
##  cloud_3pm <= 6
##  humidity_3pm > 66
##  humidity_9am <= 86
##  wind_speed_9am <= 20
##  wind_dir_3pm in [SSW-NNW]
##  wind_gust_speed > 54
##  sunshine <= 8.6
##  rainfall <= 0.2
##  ->  class Yes  [0.850]
##
## Rule 122: (1198/180, lift 4.0)
##  rain_today = Yes
##  temp_3pm > 12.4
##  cloud_3pm > 2
##  humidity_3pm > 66
##  sunshine <= 0.5
##  ->  class Yes  [0.849]
##
## Rule 123: (44/6, lift 4.0)
##  rain_today = No
##  cloud_9am > 1
##  pressure_3pm <= 1002.2
##  humidity_3pm > 50
##  humidity_3pm <= 66
##  wind_gust_speed > 50
##  wind_gust_dir in [WSW-NW]
##  sunshine > 7.7
##  ->  class Yes  [0.848]
##
## Rule 124: (727/110, lift 4.0)
##  humidity_3pm > 66
##  wind_dir_3pm in [SSW-NNW]
##  wind_dir_9am in [NW-NNW]
##  wind_gust_speed > 54
```

```
##  sunshine <= 10
##  ->  class Yes  [0.848]
##
## Rule 125: (70/10, lift 4.0)
##  cloud_3pm > 5
##  humidity_3pm > 66
##  wind_dir_3pm in [WSW-NNW]
##  wind_gust_speed > 33
##  wind_gust_dir in [N-NE]
##  sunshine > 2.3
##  rainfall <= 0
##  ->  class Yes  [0.847]
##
## Rule 126: (2698/413, lift 4.0)
##  temp_9am > 4.4
##  cloud_9am > 4
##  humidity_3pm > 66
##  wind_gust_speed > 54
##  rainfall > 0.2
##  ->  class Yes  [0.847]
##
## Rule 127: (398/62, lift 4.0)
##  temp_3pm <= 30.3
##  temp_9am > 6.4
##  cloud_3pm <= 5
##  cloud_9am > 2
##  cloud_9am <= 5
##  humidity_3pm > 83
##  humidity_9am <= 96
##  wind_speed_3pm > 13
##  wind_speed_9am <= 20
##  wind_dir_3pm in [ESE-NNW]
##  wind_dir_9am in [N-WNW]
##  wind_gust_speed > 37
##  ->  class Yes  [0.843]
##
```

```
## Rule 128: (17/2, lift 4.0)
##  rain_today = No
##  pressure_3pm <= 1011.6
##  pressure_9am > 1014.5
##  humidity_3pm > 50
##  humidity_3pm <= 66
##  humidity_9am > 65
##  wind_gust_speed > 50
##  sunshine <= 10.6
##  max_temp > 23.6
##  ->  class Yes  [0.842]
##
## Rule 129: (2126/342, lift 4.0)
##  humidity_3pm > 66
##  wind_gust_speed > 43
##  sunshine <= 5.3
##  rainfall > 0.2
##  ->  class Yes  [0.839]
##
## Rule 130: (750/121, lift 4.0)
##  temp_3pm > 12.4
##  cloud_3pm > 2
##  humidity_3pm > 66
##  humidity_9am > 77
##  wind_gust_speed > 28
##  wind_gust_dir in [E-SE]
##  rainfall > 5.1
##  ->  class Yes  [0.838]
##
## Rule 131: (1726/280, lift 4.0)
##  humidity_3pm > 72
##  wind_dir_9am in [W-NNW]
##  wind_gust_speed > 41
##  rainfall > 0.2
##  ->  class Yes  [0.837]
##
```

```
## Rule 132: (462/75, lift 4.0)
##  rain_today = Yes
##  pressure_3pm <= 1014.3
##  humidity_3pm > 52
##  humidity_9am > 86
##  wind_speed_3pm > 11
##  wind_dir_3pm in [N-ENE]
##  wind_gust_speed > 31
##  sunshine <= 9
##  ->  class Yes  [0.836]
##
## Rule 133: (28/4, lift 4.0)
##  temp_9am <= 6.4
##  cloud_3pm <= 7
##  pressure_3pm <= 1024.1
##  pressure_9am > 1018.9
##  humidity_3pm > 81
##  humidity_3pm <= 94
##  wind_gust_speed > 19
##  ->  class Yes  [0.833]
##
## Rule 134: (10/1, lift 4.0)
##  pressure_3pm <= 1006.1
##  pressure_9am > 1006.5
##  humidity_3pm <= 22
##  wind_speed_9am > 28
##  wind_gust_dir in [SSW-NNW]
##  sunshine <= 7.7
##  ->  class Yes  [0.833]
##
## Rule 135: (1851/310, lift 3.9)
##  pressure_3pm <= 1016.1
##  humidity_3pm > 66
##  wind_gust_speed > 41
##  sunshine <= 7.5
##  min_temp <= 16.1
```

```
##  ->  class Yes  [0.832]
##
## Rule 136: (4114/699, lift 3.9)
##  rain_today = Yes
##  temp_3pm > 12.4
##  cloud_3pm > 2
##  cloud_9am > 1
##  humidity_3pm > 77
##  wind_gust_speed > 28
##  ->  class Yes  [0.830]
##
## Rule 137: (2604/444, lift 3.9)
##  rain_today = Yes
##  temp_9am <= 29.1
##  cloud_3pm > 5
##  cloud_9am > 1
##  pressure_3pm <= 1013.8
##  humidity_3pm > 66
##  wind_gust_speed > 30
##  sunshine <= 9.4
##  evaporation > 1.3
##  ->  class Yes  [0.829]
##
## Rule 138: (56/9, lift 3.9)
##  rain_today = No
##  cloud_9am > 1
##  pressure_3pm <= 1012.2
##  pressure_9am <= 1014.4
##  humidity_3pm > 51
##  humidity_3pm <= 66
##  wind_gust_speed > 50
##  wind_gust_dir in [WSW-NW]
##  rainfall > 0.5
##  ->  class Yes  [0.828]
##
## Rule 139: (625/107, lift 3.9)
```

```
##  rain_today = Yes
##  temp_9am > 6.3
##  cloud_9am > 3
##  humidity_3pm > 48
##  humidity_9am > 68
##  wind_dir_3pm in [NW-NNW]
##  wind_gust_speed > 46
##  sunshine <= 9
##  ->  class Yes  [0.828]
##
## Rule 140: (2153/370, lift 3.9)
##  temp_3pm <= 30.3
##  temp_9am > 6.4
##  cloud_9am > 2
##  humidity_3pm > 81
##  humidity_9am <= 96
##  wind_speed_3pm <= 13
##  min_temp > 3.4
##  ->  class Yes  [0.828]
##
## Rule 141: (50/8, lift 3.9)
##  rain_today = Yes
##  temp_9am > 6.3
##  cloud_3pm <= 4
##  cloud_9am > 3
##  pressure_3pm > 1012.5
##  humidity_3pm > 50
##  humidity_3pm <= 66
##  humidity_9am > 68
##  wind_speed_9am <= 30
##  wind_gust_speed > 46
##  sunshine <= 7
##  ->  class Yes  [0.827]
##
## Rule 142: (21/3, lift 3.9)
##  rain_today = No
```

```
##  cloud_3pm > 5
##  pressure_3pm <= 1013.7
##  humidity_3pm > 27
##  humidity_3pm <= 66
##  humidity_9am <= 75
##  wind_speed_3pm > 15
##  wind_speed_9am <= 7
##  wind_dir_9am in [E-NNW]
##  wind_gust_speed > 41
##  wind_gust_speed <= 54
##  sunshine <= 7.7
##  ->  class Yes  [0.826]
##
## Rule 143: (21/3, lift 3.9)
##  rain_today = No
##  temp_9am <= 12.6
##  pressure_3pm > 1012.2
##  humidity_3pm > 50
##  wind_gust_speed > 50
##  wind_gust_speed <= 57
##  wind_gust_dir in [WNW-NNW]
##  sunshine > 8.3
##  sunshine <= 8.5
##  rainfall > 0.1
##  ->  class Yes  [0.826]
##
## Rule 144: (78/13, lift 3.9)
##  temp_3pm <= -1.2
##  temp_9am <= 6.4
##  humidity_3pm > 94
##  ->  class Yes  [0.825]
##
## Rule 145: (43/7, lift 3.9)
##  cloud_9am <= 5
##  pressure_3pm <= 1006.1
##  humidity_3pm > 22
```

```
##  wind_dir_3pm in [N-W]
##  wind_dir_9am in [N-ENE]
##  wind_gust_speed > 54
##  wind_gust_speed <= 74
##  sunshine > 3.9
##  sunshine <= 7.7
##  ->  class Yes  [0.822]
##
## Rule 146: (82/14, lift 3.9)
##  rain_today = Yes
##  temp_3pm <= 12.4
##  cloud_9am > 1
##  cloud_9am <= 6
##  humidity_3pm > 78
##  humidity_3pm <= 81
##  wind_gust_speed > 33
##  ->  class Yes  [0.821]
##
## Rule 147: (48/8, lift 3.9)
##  temp_9am > 17
##  humidity_3pm > 14
##  humidity_3pm <= 50
##  wind_dir_3pm in [NNE-NE]
##  wind_gust_speed > 52
##  wind_gust_speed <= 74
##  wind_gust_dir in [SSW-NNW]
##  sunshine > 8.4
##  sunshine <= 10.6
##  ->  class Yes  [0.820]
##
## Rule 148: (114/20, lift 3.9)
##  temp_3pm <= 26.3
##  humidity_3pm > 66
##  humidity_9am > 86
##  wind_speed_9am <= 7
##  wind_gust_speed > 54
```

```
##  ->  class Yes  [0.819]
##
## Rule 149: (617/111, lift 3.9)
##  humidity_3pm > 66
##  wind_gust_speed > 43
##  sunshine <= 9.2
##  evaporation > 5.5
##  rainfall > 0.2
##  ->  class Yes  [0.819]
##
## Rule 150: (2978/541, lift 3.9)
##  rain_today = Yes
##  pressure_3pm <= 1014.3
##  humidity_3pm > 60
##  humidity_9am > 86
##  wind_gust_speed > 31
##  sunshine <= 9
##  ->  class Yes  [0.818]
##
## Rule 151: (1521/283, lift 3.9)
##  pressure_3pm <= 1013.7
##  humidity_3pm > 49
##  wind_speed_9am <= 28
##  wind_gust_speed > 54
##  sunshine <= 7.7
##  ->  class Yes  [0.814]
##
## Rule 152: (411/76, lift 3.9)
##  pressure_3pm <= 1016.1
##  humidity_3pm > 66
##  humidity_9am <= 69
##  wind_gust_speed > 41
##  sunshine <= 7.5
##  rainfall <= 0.2
##  ->  class Yes  [0.814]
##
```

```
## Rule 153: (14/2, lift 3.9)
##  rain_today = Yes
##  humidity_3pm > 41
##  humidity_3pm <= 50
##  humidity_9am > 78
##  wind_gust_speed > 37
##  wind_gust_speed <= 46
##  sunshine <= 4.2
##  ->  class Yes  [0.813]
##
## Rule 154: (351/66, lift 3.8)
##  cloud_3pm <= 5
##  humidity_3pm > 66
##  wind_speed_3pm <= 19
##  wind_dir_3pm in [WSW-NNW]
##  wind_gust_speed > 41
##  sunshine > 7.5
##  max_temp > 13.7
##  ->  class Yes  [0.810]
##
## Rule 155: (40/7, lift 3.8)
##  rain_today = No
##  cloud_9am > 2
##  pressure_3pm > 1015.1
##  pressure_3pm <= 1026.1
##  humidity_3pm > 81
##  humidity_3pm <= 88
##  humidity_9am <= 96
##  wind_speed_3pm > 13
##  wind_dir_9am in [NW-NNW]
##  wind_gust_speed <= 56
##  wind_gust_dir in [SE-NNW]
##  ->  class Yes  [0.810]
##
## Rule 156: (192/36, lift 3.8)
##  pressure_3pm <= 1013.7
```

```
##  humidity_3pm > 49
##  wind_speed_9am > 28
##  wind_gust_speed > 61
##  sunshine <= 7.7
##  evaporation > 4.4
##  evaporation <= 12.9
##  ->  class Yes  [0.809]
##
## Rule 157: (1653/317, lift 3.8)
##  cloud_3pm > 4
##  cloud_9am > 7
##  humidity_3pm > 50
##  humidity_9am > 82
##  wind_gust_speed > 31
##  sunshine <= 9
##  rainfall > 6.5
##  min_temp > 3
##  ->  class Yes  [0.808]
##
## Rule 158: (610/118, lift 3.8)
##  pressure_3pm <= 1013.7
##  humidity_3pm > 27
##  wind_speed_9am > 0
##  wind_dir_9am in [N-ENE]
##  sunshine <= 0.8
##  ->  class Yes  [0.806]
##
## Rule 159: (403/78, lift 3.8)
##  pressure_3pm <= 1006.1
##  humidity_3pm > 22
##  humidity_9am > 52
##  wind_dir_3pm in [WNW-NNW]
##  wind_gust_speed > 54
##  wind_gust_dir in [N-NW]
##  sunshine <= 7.7
##  ->  class Yes  [0.805]
```

```
##
## Rule 160: (69/13, lift 3.8)
##  humidity_3pm > 14
##  wind_dir_3pm in [N-NE]
##  wind_gust_speed > 74
##  wind_gust_dir in [SSW-NNW]
##  sunshine > 7.7
##  sunshine <= 10.6
##  ->  class Yes  [0.803]
##
## Rule 161: (598/122, lift 3.8)
##  cloud_3pm > 7
##  pressure_3pm <= 1021.8
##  wind_speed_9am > 0
##  wind_dir_9am in [N-NE]
##  sunshine <= 4.7
##  evaporation <= 7.1
##  ->  class Yes  [0.795]
##
## Rule 162: (2153/441, lift 3.8)
##  pressure_3pm <= 1013.7
##  humidity_3pm > 49
##  wind_gust_speed > 54
##  sunshine <= 7.7
##  ->  class Yes  [0.795]
##
## Rule 163: (42/8, lift 3.8)
##  cloud_3pm > 3
##  humidity_3pm > 66
##  humidity_3pm <= 81
##  humidity_9am <= 86
##  wind_gust_speed > 54
##  wind_gust_dir in [S-NNW]
##  sunshine > 7.5
##  rainfall <= 0.2
##  min_temp > 21.1
```

```
##  ->  class Yes  [0.795]
##
## Rule 164: (2457/514, lift 3.7)
##  rain_today = Yes
##  pressure_3pm <= 1012.5
##  humidity_3pm > 48
##  wind_speed_9am <= 26
##  wind_gust_speed > 46
##  sunshine <= 9
##  evaporation > 0.9
##  ->  class Yes  [0.791]
##
## Rule 165: (387/81, lift 3.7)
##  cloud_9am > 3
##  humidity_3pm > 66
##  wind_speed_9am > 9
##  wind_gust_speed > 54
##  wind_gust_dir in [ENE-SSE]
##  max_temp > 21
##  ->  class Yes  [0.789]
##
## Rule 166: (822/173, lift 3.7)
##  humidity_9am > 82
##  wind_gust_speed > 46
##  wind_gust_dir in [W-NNW]
##  sunshine <= 7.7
##  ->  class Yes  [0.789]
##
## Rule 167: (31/6, lift 3.7)
##  cloud_9am <= 6
##  pressure_3pm <= 1017.1
##  humidity_3pm > 14
##  humidity_3pm <= 50
##  wind_speed_3pm <= 41
##  wind_dir_3pm in [N-NNE]
##  wind_gust_speed > 63
```

```
##  wind_gust_dir in [N-S]
##  sunshine > 8.3
##  sunshine <= 9.9
##  rainfall <= 0.1
##  ->  class Yes  [0.788]
##
## Rule 168: (616/130, lift 3.7)
##  cloud_3pm > 5
##  pressure_3pm <= 1022.9
##  humidity_3pm > 66
##  wind_speed_3pm <= 15
##  wind_gust_speed > 28
##  wind_gust_dir in [ENE-SSE]
##  ->  class Yes  [0.788]
##
## Rule 169: (2668/569, lift 3.7)
##  cloud_3pm > 5
##  pressure_3pm <= 1011.6
##  humidity_3pm > 50
##  wind_gust_speed > 50
##  sunshine <= 10.6
##  ->  class Yes  [0.787]
##
## Rule 170: (665/143, lift 3.7)
##  rain_today = Yes
##  cloud_3pm <= 6
##  cloud_9am > 3
##  humidity_3pm > 50
##  humidity_9am > 68
##  wind_gust_speed > 46
##  sunshine <= 7
##  rainfall <= 20.2
##  ->  class Yes  [0.784]
##
## Rule 171: (780/169, lift 3.7)
##  humidity_3pm > 50
```

```
##  wind_speed_3pm <= 35
##  wind_gust_speed > 50
##  wind_gust_dir in [WNW-NNW]
##  sunshine > 8.3
##  sunshine <= 8.5
##  rainfall > 0.1
##  ->  class Yes  [0.783]
##
## Rule 172: (1091/246, lift 3.7)
##  rain_today = Yes
##  temp_9am > 6.3
##  cloud_3pm > 4
##  cloud_3pm <= 6
##  cloud_9am > 3
##  cloud_9am <= 5
##  pressure_3pm <= 1020.2
##  humidity_3pm > 48
##  humidity_9am > 68
##  wind_speed_3pm <= 28
##  wind_gust_speed > 46
##  sunshine <= 9
##  ->  class Yes  [0.774]
##
## Rule 173: (610/139, lift 3.7)
##  rain_today = Yes
##  cloud_9am > 7
##  pressure_3pm <= 1016
##  humidity_3pm > 50
##  wind_dir_9am in [N-NNE]
##  ->  class Yes  [0.771]
##
## Rule 174: (158/36, lift 3.6)
##  humidity_3pm > 50
##  wind_gust_speed > 50
##  sunshine <= 10.6
##  min_temp > 25.1
```

```
##  ->  class Yes  [0.769]
##
## Rule 175: (24/5, lift 3.6)
##  rain_today = No
##  cloud_3pm > 5
##  cloud_9am > 7
##  pressure_3pm > 1008.5
##  pressure_3pm <= 1022.9
##  humidity_3pm > 66
##  humidity_3pm <= 81
##  wind_gust_speed <= 39
##  wind_gust_dir in [N-NE]
##  sunshine > 0.7
##  rainfall > 0
##  ->  class Yes  [0.769]
##
## Rule 176: (24/5, lift 3.6)
##  rain_today = No
##  cloud_3pm > 5
##  pressure_3pm > 1008.5
##  pressure_3pm <= 1022.9
##  humidity_3pm > 76
##  humidity_3pm <= 81
##  wind_speed_3pm <= 15
##  wind_dir_9am in [WSW-NNW]
##  wind_gust_speed > 28
##  wind_gust_speed <= 41
##  wind_gust_dir in [S-NNW]
##  sunshine > 0.7
##  ->  class Yes  [0.769]
##
## Rule 177: (80/18, lift 3.6)
##  rain_today = No
##  pressure_3pm <= 1010.8
##  pressure_9am <= 1012.2
##  humidity_3pm > 49
```

```
##  wind_speed_3pm <= 19
##  wind_gust_speed > 37
##  wind_gust_speed <= 50
##  sunshine > 7.7
##  rainfall > 0.3
##  ->  class Yes  [0.768]
##
## Rule 178: (66/15, lift 3.6)
##  rain_today = No
##  pressure_9am <= 1012.2
##  humidity_3pm > 49
##  wind_dir_9am in [N-NNE]
##  wind_gust_speed > 46
##  wind_gust_speed <= 50
##  sunshine > 7.7
##  ->  class Yes  [0.765]
##
## Rule 179: (465/111, lift 3.6)
##  rain_today = Yes
##  humidity_9am > 49
##  wind_speed_3pm <= 22
##  wind_dir_3pm in [N-ESE]
##  wind_dir_9am in [N-S]
##  wind_gust_speed > 46
##  sunshine <= 9
##  ->  class Yes  [0.760]
##
## Rule 180: (1652/398, lift 3.6)
##  rain_today = Yes
##  temp_9am > 6.3
##  cloud_3pm > 4
##  cloud_3pm <= 6
##  cloud_9am > 3
##  cloud_9am <= 5
##  pressure_3pm <= 1020.2
##  humidity_3pm > 48
```

```
##  humidity_9am > 68
##  wind_gust_speed > 46
##  sunshine <= 9
##  ->  class Yes  [0.759]
##
## Rule 181: (597/144, lift 3.6)
##  pressure_3pm <= 1013.7
##  humidity_3pm > 47
##  humidity_9am > 29
##  wind_speed_9am > 0
##  wind_dir_3pm in [NE-WNW]
##  wind_dir_9am in [N-NE]
##  wind_gust_speed > 35
##  sunshine <= 6.5
##  ->  class Yes  [0.758]
##
## Rule 182: (26/6, lift 3.6)
##  rain_today = No
##  pressure_3pm > 1012.2
##  humidity_3pm > 50
##  wind_speed_3pm <= 35
##  wind_dir_3pm in [N-E]
##  wind_gust_speed > 50
##  wind_gust_speed <= 57
##  wind_gust_dir in [WSW-NNW]
##  sunshine > 8.3
##  sunshine <= 8.5
##  ->  class Yes  [0.750]
##
## Rule 183: (45/11, lift 3.5)
##  rain_today = No
##  cloud_3pm > 5
##  pressure_3pm <= 1022.9
##  humidity_3pm > 66
##  humidity_3pm <= 81
##  wind_speed_3pm > 7
```

```
##  wind_gust_speed <= 39
##  wind_gust_dir in [N-NE]
##  sunshine > 0.7
##  rainfall > 0
##  min_temp > 7.1
##  ->  class Yes  [0.745]
##
## Rule 184: (37/9, lift 3.5)
##  rain_today = Yes
##  temp_3pm > 12.4
##  cloud_9am > 4
##  pressure_3pm > 1013.8
##  humidity_3pm > 66
##  humidity_3pm <= 77
##  wind_dir_3pm in [WSW-NNW]
##  wind_gust_speed > 37
##  wind_gust_speed <= 41
##  evaporation > 1.3
##  rainfall <= 5.1
##  ->  class Yes  [0.744]
##
## Rule 185: (270/69, lift 3.5)
##  cloud_3pm > 3
##  pressure_3pm <= 1013.7
##  humidity_3pm > 27
##  wind_speed_9am <= 7
##  wind_dir_9am in [E-NNW]
##  wind_gust_speed > 41
##  sunshine <= 7.7
##  ->  class Yes  [0.743]
##
## Rule 186: (25/6, lift 3.5)
##  rain_today = Yes
##  pressure_3pm > 1000
##  pressure_3pm <= 1014.3
##  humidity_3pm > 50
```

```
##  humidity_3pm <= 66
##  humidity_9am <= 86
##  wind_speed_9am > 26
##  wind_dir_9am in [ENE-NNW]
##  wind_gust_speed <= 39
##  sunshine <= 9
##  ->  class Yes  [0.741]
##
## Rule 187: (1399/362, lift 3.5)
##  pressure_3pm <= 1006.1
##  wind_gust_speed > 54
##  sunshine <= 7.7
##  ->  class Yes  [0.741]
##
## Rule 188: (21/5, lift 3.5)
##  rain_today = Yes
##  temp_3pm > 12.4
##  cloud_3pm > 2
##  cloud_9am > 1
##  cloud_9am <= 4
##  pressure_9am > 1017.5
##  humidity_3pm > 66
##  humidity_3pm <= 77
##  wind_gust_speed > 28
##  wind_gust_speed <= 41
##  evaporation > 1.3
##  rainfall <= 5.1
##  ->  class Yes  [0.739]
##
## Rule 189: (115/30, lift 3.5)
##  humidity_3pm > 66
##  humidity_9am <= 86
##  wind_dir_3pm in [N-S]
##  wind_gust_speed > 54
##  sunshine > 7.5
##  sunshine <= 10
```

```
##  rainfall <= 0.1
##  max_temp <= 21
##  ->  class Yes  [0.735]
##
## Rule 190: (39/10, lift 3.5)
##  rain_today = Yes
##  cloud_3pm > 4
##  humidity_3pm <= 50
##  wind_dir_9am in [N-E]
##  wind_gust_speed > 37
##  wind_gust_speed <= 46
##  sunshine <= 6.4
##  ->  class Yes  [0.732]
##
## Rule 191: (1298/349, lift 3.5)
##  rain_today = Yes
##  wind_speed_3pm > 28
##  sunshine <= 6.4
##  ->  class Yes  [0.731]
##
## Rule 192: (93/26, lift 3.4)
##  rain_today = No
##  pressure_9am <= 1012.2
##  humidity_3pm > 49
##  wind_speed_3pm <= 19
##  wind_gust_speed > 37
##  wind_gust_speed <= 50
##  sunshine > 7.7
##  rainfall > 0.3
##  ->  class Yes  [0.716]
##
## Rule 193: (602/175, lift 3.4)
##  pressure_3pm <= 1013.7
##  humidity_3pm > 27
##  wind_speed_9am > 0
##  wind_dir_3pm in [NW-NNW]
```

```
##  wind_dir_9am in [N-ENE]
##  wind_gust_speed > 35
##  sunshine <= 7.7
##  ->  class Yes  [0.709]
##
## Rule 194: (25/7, lift 3.3)
##  pressure_9am <= 1011.1
##  humidity_3pm > 49
##  humidity_3pm <= 66
##  wind_dir_9am in [ENE-NW]
##  wind_gust_speed > 46
##  wind_gust_speed <= 50
##  wind_gust_dir in [NE-S]
##  sunshine <= 11.6
##  rainfall <= 0.3
##  max_temp > 32
##  ->  class Yes  [0.704]
##
## Rule 195: (378/115, lift 3.3)
##  temp_9am <= 13.6
##  pressure_3pm <= 1021.8
##  pressure_9am <= 1026
##  wind_speed_9am > 0
##  wind_dir_3pm in [W-NNW]
##  wind_dir_9am in [N-NE]
##  wind_gust_speed > 22
##  sunshine <= 4.7
##  ->  class Yes  [0.695]
##
## Rule 196: (397/122, lift 3.3)
##  pressure_3pm <= 1013.7
##  humidity_3pm > 27
##  wind_speed_9am > 0
##  wind_dir_3pm in [N-NNE]
##  wind_dir_9am in [N-ENE]
##  wind_gust_speed > 35
```

```
##  wind_gust_speed <= 54
##  sunshine <= 7.7
##  ->  class Yes  [0.692]
##
## Rule 197: (88/27, lift 3.3)
##  pressure_3pm <= 1017.1
##  humidity_3pm > 14
##  humidity_3pm <= 50
##  wind_dir_3pm in [NNE-NE]
##  wind_gust_speed > 50
##  wind_gust_dir in [SSW-NNW]
##  sunshine > 7.7
##  sunshine <= 10.6
##  ->  class Yes  [0.689]
##
## Rule 198: (394/125, lift 3.2)
##  temp_9am <= 13.6
##  pressure_3pm <= 1021.8
##  pressure_9am <= 1026
##  wind_speed_9am > 0
##  wind_dir_3pm in [W-NNW]
##  wind_dir_9am in [N-NE]
##  sunshine <= 4.7
##  ->  class Yes  [0.682]
##
## Rule 199: (48/16, lift 3.1)
##  rain_today = Yes
##  humidity_3pm > 41
##  humidity_3pm <= 50
##  humidity_9am > 78
##  wind_gust_speed > 37
##  wind_gust_speed <= 46
##  sunshine <= 6.4
##  ->  class Yes  [0.660]
##
## Rule 200: (30/10, lift 3.1)
```

```
##  cloud_3pm <= 7
##  cloud_9am <= 5
##  pressure_9am <= 1012.2
##  humidity_3pm > 49
##  humidity_3pm <= 66
##  wind_dir_3pm in [WSW-NNW]
##  wind_gust_speed > 46
##  wind_gust_speed <= 50
##  wind_gust_dir in [WSW-NW]
##  sunshine <= 11.6
##  evaporation > 4.5
##  rainfall <= 0.3
##  ->  class Yes  [0.656]
##
## Rule 201: (352/127, lift 3.0)
##  pressure_3pm <= 1017.1
##  humidity_3pm > 14
##  wind_dir_3pm in [N-NNE]
##  wind_gust_speed > 63
##  sunshine > 8.3
##  sunshine <= 10.6
##  ->  class Yes  [0.638]
##
## Rule 202: (318/119, lift 3.0)
##  pressure_9am <= 1012.2
##  humidity_3pm > 49
##  wind_dir_9am in [NE-NNW]
##  wind_gust_speed > 46
##  wind_gust_dir in [N-SW]
##  sunshine <= 11.6
##  max_temp > 32
##  ->  class Yes  [0.625]
##
## Rule 203: (44/18, lift 2.8)
##  rain_today = Yes
##  pressure_3pm <= 1014.3
```

```
##  humidity_3pm > 50
##  humidity_3pm <= 66
##  humidity_9am <= 86
##  wind_speed_9am > 26
##  wind_gust_speed <= 46
##  sunshine <= 9
##  ->  class Yes  [0.587]
##
## Rule 204: (27272/13005, lift 2.5)
##  humidity_3pm > 66
##  ->  class Yes  [0.523]
##
## Default class: No
##
##
## Evaluation on training data (123722 cases):
##
##          Rules
##    ----------------
##      No      Errors
##
##     204 16599(13.4%)   <<
##
##
##     (a)   (b)    <-classified as
##    ----  ----
##   93517  4104    (a): class No
##   12495 13606    (b): class Yes
##
##
##  Attribute usage:
##
##  100.00% humidity_3pm
##   53.22% pressure_3pm
##   47.86% wind_gust_speed
##   44.72% rainfall
```

```
##   41.80% wind_dir_3pm
##   31.60% cloud_3pm
##   30.43% temp_3pm
##   29.96% sunshine
##   29.42% rain_today
##   22.57% wind_speed_3pm
##   17.70% cloud_9am
##   16.99% wind_gust_dir
##   16.32% humidity_9am
##   14.69% temp_9am
##   14.40% min_temp
##   12.50% wind_speed_9am
##    6.33% wind_dir_9am
##    4.65% evaporation
##    1.90% pressure_9am
##    1.08% max_temp
##
##
## Time: 38.7 secs
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.74 C5.0 Rules Performance

Here we plot the performance of the decision tree, showing a risk chart. The areas under the recall and risk curves are also reported.

```
predicted <- predict(model, ds[te, vars], type="prob")[,2]
riskchart(predicted, actual_te, risk_te)
```

An error matrix shows, clockwise from the top left, the percentages of true negatives, false positives, true positives, and false negatives.

```
predicted <- predict(model, ds[te, vars], type="class")
sum(ds[te, target] != predicted)/length(predicted) # Overall error rate
round(100*table(ds[te, target], predicted, dnn=c("Actual", "Predicted"))/length(pre
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.75 Regression Trees

The discussion so far has dwelt on classification trees. Regression trees are similarly well catered for in R.

We can plot regression trees as with classification trees, but the node information will be different and some options will not make sense. For example, *extra*= only makes sense for 100 and 101.

First we will build regression tree:

```
target <- "risk_mm"
vars <- c(inputs, target)
form <- formula(paste(target, "~ ."))
(model <- rpart(formula=form, data=ds[tr, vars]))
```

```
## n=120704 (3018 observations deleted due to missingness)
##
## node), split, n, deviance, yval
##       * denotes terminal node
##
##  1) root 120704 8785460.0  2.272201
##    2) humidity_3pm< 83.5 112864 3791677.0  1.486089
##      4) humidity_3pm< 66.5 94010 1731286.0  0.983856 *
##      5) humidity_3pm>=66.5 18854 1918441.0  3.990326 *
##    3) humidity_3pm>=83.5 7840 3919967.0 13.589010
##      6) rainfall< 39.1 7439 2505518.0 11.983020
##       12) min_temp< 15.95 5209  838540.7  9.100749 *
##       13) min_temp>=15.95 2230 1522622.0 18.715650 *
##      7) rainfall>=39.1 401 1039330.0 43.381800 *
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.76 Visualise Regression Trees

```
plot(model)
text(model)
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.77 Visualise Regression Trees: Uniform

```
plot(model, uniform=TRUE)
text(model)
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.78 Visualise Regression Trees: Extra Information

```
plot(model, uniform=TRUE)
text(model, use.n=TRUE, all=TRUE, cex=.8)
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.79 Fancy Plot of Regression Tree

```
fancyRpartPlot(model)
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*

Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.80 Enhanced Plot of Regression Tree: Default

```
prp(model)
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.81 Enhanced Plot of Regression Tree: Favourite

```
prp(model, type=2, extra=101, nn=TRUE, fallen.leaves=TRUE,
    faclen=0, varlen=0, shadow.col="grey", branch.lty=3)
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*

Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.82 Party Regression Tree

The tree drawing facilities of (Hothorn et al. 2021) can again be used to draw the regression tree using partykit::as.party().

```
class(model)
```

```
## [1] "rpart"
```

```
plot(as.party(model))
```

Notice the visualisation of the predictions—this is particularly informative. We have, in an instant, a view of the conditions when there is little or no rain, compared to significant rain.

References

Hothorn, Torsten, Kurt Hornik, Carolin Strobl, and Achim Zeileis. 2021. *Party: A Laboratory for Recursive Partytioning*. http://party.R-forge.R-project.org.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.83 Conditional Regression Tree

We can also build a regression tree using (Hothorn et al. 2021).

```
model <- ctree(formula=form, data=ds[tr, vars])
```

```
model
```

```
##
## Model formula:
## risk_mm ~ rain_today + temp_3pm + temp_9am + cloud_3pm + cloud_9am +
##     pressure_3pm + pressure_9am + humidity_3pm + humidity_9am +
##     wind_speed_3pm + wind_speed_9am + wind_dir_3pm + wind_dir_9am +
##     wind_gust_speed + wind_gust_dir + sunshine + evaporation +
##     rainfall + max_temp + min_temp
##
## Fitted party:
## [1] root
## |   [2] rainfall <= 24
## |   |   [3] humidity_3pm <= 81
## |   |   |   [4] humidity_3pm <= 66
## |   |   |   |   [5] wind_gust_speed <= 52
## |   |   |   |   |   [6] humidity_3pm <= 50
## |   |   |   |   |   |   [7] sunshine <= 7.5
## |   |   |   |   |   |   |   [8] pressure_3pm <= 1014.5
## |   |   |   |   |   |   |   |   [9] wind_dir_9am <= NE
## |   |   |   |   |   |   |   |   |   [10] humidity_3pm <= 33: 1.722 (n = 188, err
## |   |   |   |   |   |   |   |   |   [11] humidity_3pm > 33
## |   |   |   |   |   |   |   |   |   |   [12] cloud_3pm <= 4
## |   |   |   |   |   |   |   |   |   |   |   [13] sunshine <= 5.7: 3.450 (n = 18,
## |   |   |   |   |   |   |   |   |   |   |   [14] sunshine > 5.7: 0.331 (n = 33, e
## |   |   |   |   |   |   |   |   |   |   [15] cloud_3pm > 4: 4.595 (n = 305, err =
## |   |   |   |   |   |   |   |   [16] wind_dir_9am > NE
## |   |   |   |   |   |   |   |   |   [17] pressure_3pm <= 1005.1: 3.847 (n = 127,
## |   |   |   |   |   |   |   |   |   [18] pressure_3pm > 1005.1: 1.345 (n = 1065,
## |   |   |   |   |   |   |   [19] pressure_3pm > 1014.5
## |   |   |   |   |   |   |   |   [20] pressure_3pm <= 1023.5
## |   |   |   |   |   |   |   |   |   [21] cloud_3pm <= 5: 0.423 (n = 855, err = NA
## |   |   |   |   |   |   |   |   |   [22] cloud_3pm > 5: 0.826 (n = 1029, err = NA
## |   |   |   |   |   |   |   |   [23] pressure_3pm > 1023.5
## |   |   |   |   |   |   |   |   |   [24] rainfall <= 9: 0.099 (n = 506, err = NA
## |   |   |   |   |   |   |   |   |   [25] rainfall > 9: 1.000 (n = 12, err = 31.4
## |   |   |   |   |   |   [26] sunshine > 7.5
```

```
## |   |   |   |   |   |   |   [27] pressure_3pm <= 1008.5
## |   |   |   |   |   |   |   |   [28] humidity_3pm <= 31
## |   |   |   |   |   |   |   |   |   [29] humidity_3pm <= 23
## |   |   |   |   |   |   |   |   |   |   [30] humidity_3pm <= 13: 0.046 (n = 757,
## |   |   |   |   |   |   |   |   |   |   [31] humidity_3pm > 13: 0.257 (n = 952, e
## |   |   |   |   |   |   |   |   |   [32] humidity_3pm > 23: 0.705 (n = 788, err =
## |   |   |   |   |   |   |   |   [33] humidity_3pm > 31
## |   |   |   |   |   |   |   |   |   [34] cloud_3pm <= 3
## |   |   |   |   |   |   |   |   |   |   [35] sunshine <= 9.9: 1.001 (n = 243, err
## |   |   |   |   |   |   |   |   |   |   [36] sunshine > 9.9: 0.250 (n = 383, err
## |   |   |   |   |   |   |   |   |   [37] cloud_3pm > 3
## |   |   |   |   |   |   |   |   |   |   [38] temp_3pm <= 32.1
## |   |   |   |   |   |   |   |   |   |   |   [39] sunshine <= 9.3: 1.611 (n = 858,
## |   |   |   |   |   |   |   |   |   |   |   [40] sunshine > 9.3: 0.457 (n = 197,
## |   |   |   |   |   |   |   |   |   |   [41] temp_3pm > 32.1: 3.656 (n = 296, err
## |   |   |   |   |   |   |   [42] pressure_3pm > 1008.5
## |   |   |   |   |   |   |   |   [43] humidity_3pm <= 31
## |   |   |   |   |   |   |   |   |   [44] wind_gust_speed <= 37
## |   |   |   |   |   |   |   |   |   |   [45] humidity_3pm <= 20: 0.016 (n = 3019,
## |   |   |   |   |   |   |   |   |   |   [46] humidity_3pm > 20
## |   |   |   |   |   |   |   |   |   |   |   [47] cloud_3pm <= 5
## |   |   |   |   |   |   |   |   |   |   |   |   [48] pressure_3pm <= 1018.4: 0.09
## |   |   |   |   |   |   |   |   |   |   |   |   [49] pressure_3pm > 1018.4
## |   |   |   |   |   |   |   |   |   |   |   |   |   [50] rainfall <= 0.2: 0.012 (
## |   |   |   |   |   |   |   |   |   |   |   |   |   [51] rainfall > 0.2: 0.484 (n
## |   |   |   |   |   |   |   |   |   |   |   [52] cloud_3pm > 5: 0.296 (n = 321, e
## |   |   |   |   |   |   |   |   |   [53] wind_gust_speed > 37
## |   |   |   |   |   |   |   |   |   |   [54] humidity_3pm <= 21
## |   |   |   |   |   |   |   |   |   |   |   [55] humidity_9am <= 55
## |   |   |   |   |   |   |   |   |   |   |   |   [56] cloud_3pm <= 6: 0.041 (n = 3
## |   |   |   |   |   |   |   |   |   |   |   |   [57] cloud_3pm > 6: 0.231 (n = 13
## |   |   |   |   |   |   |   |   |   |   |   [58] humidity_9am > 55: 0.258 (n = 52
## |   |   |   |   |   |   |   |   |   |   [59] humidity_3pm > 21
## |   |   |   |   |   |   |   |   |   |   |   [60] temp_3pm <= 36.9
## |   |   |   |   |   |   |   |   |   |   |   |   [61] pressure_3pm <= 1015.4
## |   |   |   |   |   |   |   |   |   |   |   |   |   [62] wind_dir_9am <= ENE: 0.0
```

```
## |   |   |   |   |   |   |   |   |   |   |   |   [63] wind_dir_9am > ENE
## |   |   |   |   |   |   |   |   |   |   |   |   |   [64] rain_today in No
## |   |   |   |   |   |   |   |   |   |   |   |   |   |   [65] rainfall <= 0.4
## |   |   |   |   |   |   |   |   |   |   |   |   |   |   [66] rainfall > 0.4:
## |   |   |   |   |   |   |   |   |   |   |   |   |   [67] rain_today in Yes: (
## |   |   |   |   |   |   |   |   |   |   |   [68] pressure_3pm > 1015.4: 0.094
## |   |   |   |   |   |   |   |   |   |   [69] temp_3pm > 36.9: 1.862 (n = 47,
## |   |   |   |   |   |   |   [70] humidity_3pm > 31
## |   |   |   |   |   |   |   |   [71] pressure_3pm <= 1016
## |   |   |   |   |   |   |   |   |   [72] sunshine <= 9.3
## |   |   |   |   |   |   |   |   |   |   [73] wind_gust_speed <= 35: 0.377 (n
## |   |   |   |   |   |   |   |   |   |   [74] wind_gust_speed > 35
## |   |   |   |   |   |   |   |   |   |   |   [75] max_temp <= 31.2
## |   |   |   |   |   |   |   |   |   |   |   |   [76] cloud_3pm <= 6
## |   |   |   |   |   |   |   |   |   |   |   |   |   [77] wind_speed_9am <= 15
## |   |   |   |   |   |   |   |   |   |   |   |   |   |   [78] pressure_3pm <=
## |   |   |   |   |   |   |   |   |   |   |   |   |   |   [79] pressure_3pm > 1
## |   |   |   |   |   |   |   |   |   |   |   |   |   [80] wind_speed_9am > 15
## |   |   |   |   |   |   |   |   |   |   |   |   |   |   [81] wind_gust_speed
## |   |   |   |   |   |   |   |   |   |   |   |   |   |   [82] wind_gust_speed
## |   |   |   |   |   |   |   |   |   |   |   |   [83] cloud_3pm > 6: 1.979 (n
## |   |   |   |   |   |   |   |   |   |   |   [84] max_temp > 31.2: 1.647 (n =
## |   |   |   |   |   |   |   |   |   |   [85] sunshine > 9.3
## |   |   |   |   |   |   |   |   |   |   |   [86] sunshine <= 11.7: 0.279 (n = 234
## |   |   |   |   |   |   |   |   |   |   |   [87] sunshine > 11.7: 0.037 (n = 981,
## |   |   |   |   |   |   |   |   |   [88] pressure_3pm > 1016
## |   |   |   |   |   |   |   |   |   |   [89] pressure_3pm <= 1019
## |   |   |   |   |   |   |   |   |   |   |   [90] wind_dir_9am <= ENE
## |   |   |   |   |   |   |   |   |   |   |   |   [91] wind_gust_speed <= 37: 0.340
## |   |   |   |   |   |   |   |   |   |   |   |   [92] wind_gust_speed > 37: 0.895
## |   |   |   |   |   |   |   |   |   |   |   [93] wind_dir_9am > ENE
## |   |   |   |   |   |   |   |   |   |   |   |   [94] humidity_9am <= 93
## |   |   |   |   |   |   |   |   |   |   |   |   |   [95] cloud_3pm <= 3
## |   |   |   |   |   |   |   |   |   |   |   |   |   |   [96] humidity_3pm <= 45:
## |   |   |   |   |   |   |   |   |   |   |   |   |   |   [97] humidity_3pm > 45: (
## |   |   |   |   |   |   |   |   |   |   |   |   |   [98] cloud_3pm > 3: 0.218 (n
```

```
## |   |   |   |   |   |   |   |   |   |   |   [99] humidity_9am > 93: 0.805 (n
## |   |   |   |   |   |   |   |   |   [100] pressure_3pm > 1019
## |   |   |   |   |   |   |   |   |   |   [101] humidity_3pm <= 44: 0.098 (n =
## |   |   |   |   |   |   |   |   |   |   [102] humidity_3pm > 44
## |   |   |   |   |   |   |   |   |   |   |   [103] rainfall <= 6.6: 0.185 (n =
## |   |   |   |   |   |   |   |   |   |   |   [104] rainfall > 6.6: 0.861 (n =
## |   |   |   |   |   [105] humidity_3pm > 50
## |   |   |   |   |   |   [106] pressure_3pm <= 1009.1
## |   |   |   |   |   |   |   [107] sunshine <= 9.3
## |   |   |   |   |   |   |   |   [108] wind_gust_speed <= 37: 2.790 (n = 1278, err
## |   |   |   |   |   |   |   |   [109] wind_gust_speed > 37
## |   |   |   |   |   |   |   |   |   [110] humidity_9am <= 64
## |   |   |   |   |   |   |   |   |   |   [111] cloud_3pm <= 6
## |   |   |   |   |   |   |   |   |   |   |   [112] max_temp <= 21.5
## |   |   |   |   |   |   |   |   |   |   |   |   [113] wind_gust_speed <= 39: 10.1
## |   |   |   |   |   |   |   |   |   |   |   |   [114] wind_gust_speed > 39: 2.532
## |   |   |   |   |   |   |   |   |   |   |   [115] max_temp > 21.5: 1.147 (n = 237
## |   |   |   |   |   |   |   |   |   |   [116] cloud_3pm > 6: 4.210 (n = 111, err
## |   |   |   |   |   |   |   |   |   [117] humidity_9am > 64
## |   |   |   |   |   |   |   |   |   |   [118] wind_speed_3pm <= 17
## |   |   |   |   |   |   |   |   |   |   |   [119] min_temp <= 22.9: 5.541 (n = 27
## |   |   |   |   |   |   |   |   |   |   |   [120] min_temp > 22.9: 12.330 (n = 10
## |   |   |   |   |   |   |   |   |   |   [121] wind_speed_3pm > 17: 4.506 (n = 871
## |   |   |   |   |   |   |   [122] sunshine > 9.3
## |   |   |   |   |   |   |   |   [123] humidity_3pm <= 64: 1.213 (n = 783, err = 1
## |   |   |   |   |   |   |   |   [124] humidity_3pm > 64: 3.271 (n = 87, err = 978
## |   |   |   |   |   |   [125] pressure_3pm > 1009.1
## |   |   |   |   |   |   |   [126] pressure_3pm <= 1015.3
## |   |   |   |   |   |   |   |   [127] cloud_3pm <= 4
## |   |   |   |   |   |   |   |   |   [128] sunshine <= 10.4
## |   |   |   |   |   |   |   |   |   |   [129] wind_gust_speed <= 44: 0.886 (n = 1
## |   |   |   |   |   |   |   |   |   |   [130] wind_gust_speed > 44: 2.314 (n = 22
## |   |   |   |   |   |   |   |   |   [131] sunshine > 10.4: 0.262 (n = 904, err =
## |   |   |   |   |   |   |   |   [132] cloud_3pm > 4
## |   |   |   |   |   |   |   |   |   [133] rainfall <= 2.1
## |   |   |   |   |   |   |   |   |   |   [134] wind_gust_speed <= 37
```

```
## |   |   |   |   |   |   |   |   |   |   [135] cloud_3pm <= 5: 0.908 (n = 265
## |   |   |   |   |   |   |   |   |   |   [136] cloud_3pm > 5: 1.745 (n = 947,
## |   |   |   |   |   |   |   |   |   [137] wind_gust_speed > 37
## |   |   |   |   |   |   |   |   |   |   [138] cloud_9am <= 1
## |   |   |   |   |   |   |   |   |   |   |   [139] humidity_3pm <= 62: 0.529 (
## |   |   |   |   |   |   |   |   |   |   |   [140] humidity_3pm > 62: 4.769 (n
## |   |   |   |   |   |   |   |   |   |   [141] cloud_9am > 1: 2.046 (n = 4266,
## |   |   |   |   |   |   |   |   [142] rainfall > 2.1
## |   |   |   |   |   |   |   |   |   [143] wind_gust_speed <= 33: 1.801 (n = 3
## |   |   |   |   |   |   |   |   |   [144] wind_gust_speed > 33: 3.625 (n = 10
## |   |   |   |   |   |   [145] pressure_3pm > 1015.3
## |   |   |   |   |   |   |   [146] wind_gust_speed <= 35
## |   |   |   |   |   |   |   |   [147] pressure_3pm <= 1023.3
## |   |   |   |   |   |   |   |   |   [148] sunshine <= 3.3: 1.514 (n = 499, er
## |   |   |   |   |   |   |   |   |   [149] sunshine > 3.3
## |   |   |   |   |   |   |   |   |   |   [150] humidity_3pm <= 56: 0.399 (n =
## |   |   |   |   |   |   |   |   |   |   [151] humidity_3pm > 56
## |   |   |   |   |   |   |   |   |   |   |   [152] cloud_3pm <= 7
## |   |   |   |   |   |   |   |   |   |   |   |   [153] cloud_3pm <= 3: 0.378 (
## |   |   |   |   |   |   |   |   |   |   |   |   [154] cloud_3pm > 3: 0.766 (n
## |   |   |   |   |   |   |   |   |   |   |   [155] cloud_3pm > 7: 1.465 (n = 2
## |   |   |   |   |   |   |   |   [156] pressure_3pm > 1023.3
## |   |   |   |   |   |   |   |   |   [157] rainfall <= 10
## |   |   |   |   |   |   |   |   |   |   [158] wind_gust_speed <= 26: 0.157 (n
## |   |   |   |   |   |   |   |   |   |   [159] wind_gust_speed > 26
## |   |   |   |   |   |   |   |   |   |   |   [160] cloud_9am <= 2: 0.121 (n =
## |   |   |   |   |   |   |   |   |   |   |   [161] cloud_9am > 2
## |   |   |   |   |   |   |   |   |   |   |   |   [162] max_temp <= 20.3
## |   |   |   |   |   |   |   |   |   |   |   |   |   [163] humidity_3pm <= 61
## |   |   |   |   |   |   |   |   |   |   |   |   |   [164] humidity_3pm > 61:
## |   |   |   |   |   |   |   |   |   |   |   |   [165] max_temp > 20.3
## |   |   |   |   |   |   |   |   |   |   |   |   |   [166] sunshine <= 7.6: 1
## |   |   |   |   |   |   |   |   |   |   |   |   |   [167] sunshine > 7.6: 0.4
## |   |   |   |   |   |   |   |   |   [168] rainfall > 10
## |   |   |   |   |   |   |   |   |   |   [169] sunshine <= 3.8: 7.100 (n = 7,
## |   |   |   |   |   |   |   |   |   |   [170] sunshine > 3.8
```

```
## |   |   |   |   |   |   |   |   |   |   |   [171] sunshine <= 6.9: 0.886 (n =
## |   |   |   |   |   |   |   |   |   |   |   [172] sunshine > 6.9: 0.160 (n =
## |   |   |   |   |   |   |   |   [173] wind_gust_speed > 35
## |   |   |   |   |   |   |   |   |   [174] sunshine <= 10.2
## |   |   |   |   |   |   |   |   |   |   [175] rainfall <= 3.2
## |   |   |   |   |   |   |   |   |   |   |   [176] cloud_3pm <= 5: 0.837 (n = 3930
## |   |   |   |   |   |   |   |   |   |   |   [177] cloud_3pm > 5
## |   |   |   |   |   |   |   |   |   |   |   |   [178] temp_3pm <= 16.3
## |   |   |   |   |   |   |   |   |   |   |   |   |   [179] sunshine <= 5.1: 1.302
## |   |   |   |   |   |   |   |   |   |   |   |   |   [180] sunshine > 5.1
## |   |   |   |   |   |   |   |   |   |   |   |   |   |   [181] wind_speed_9am <= 6
## |   |   |   |   |   |   |   |   |   |   |   |   |   |   [182] wind_speed_9am > 6
## |   |   |   |   |   |   |   |   |   |   |   |   [183] temp_3pm > 16.3: 1.600 (n =
## |   |   |   |   |   |   |   |   |   |   [184] rainfall > 3.2
## |   |   |   |   |   |   |   |   |   |   |   [185] humidity_9am <= 79: 1.271 (n =
## |   |   |   |   |   |   |   |   |   |   |   [186] humidity_9am > 79: 2.621 (n = 4
## |   |   |   |   |   |   |   |   |   [187] sunshine > 10.2: 0.104 (n = 674, err =
## |   |   |   |   [188] wind_gust_speed > 52
## |   |   |   |   |   [189] humidity_3pm <= 50
## |   |   |   |   |   |   [190] sunshine <= 7.9
## |   |   |   |   |   |   |   [191] min_temp <= 18
## |   |   |   |   |   |   |   |   [192] pressure_3pm <= 1004.3
## |   |   |   |   |   |   |   |   |   [193] humidity_3pm <= 33
## |   |   |   |   |   |   |   |   |   |   [194] sunshine <= 4.1: 4.429 (n = 31, err
## |   |   |   |   |   |   |   |   |   |   [195] sunshine > 4.1: 0.852 (n = 62, err
## |   |   |   |   |   |   |   |   |   [196] humidity_3pm > 33: 4.166 (n = 226, err
## |   |   |   |   |   |   |   |   [197] pressure_3pm > 1004.3
## |   |   |   |   |   |   |   |   |   [198] wind_gust_speed <= 93
## |   |   |   |   |   |   |   |   |   |   [199] temp_9am <= 22.8
## |   |   |   |   |   |   |   |   |   |   |   [200] sunshine <= 0.1: 10.822 (n = 9,
## |   |   |   |   |   |   |   |   |   |   |   [201] sunshine > 0.1: 1.689 (n = 828,
## |   |   |   |   |   |   |   |   |   |   [202] temp_9am > 22.8: 4.084 (n = 55, err
## |   |   |   |   |   |   |   |   |   [203] wind_gust_speed > 93: 15.778 (n = 9, er
## |   |   |   |   |   |   |   [204] min_temp > 18
## |   |   |   |   |   |   |   |   [205] humidity_9am <= 50: 3.275 (n = 190, err = 
## |   |   |   |   |   |   |   |   [206] humidity_9am > 50
```

```
## |   |   |   |   |   |   |   |   [207] wind_gust_speed <= 74: 5.735 (n = 109,
## |   |   |   |   |   |   |   |   [208] wind_gust_speed > 74: 19.500 (n = 17, e
## |   |   |   |   |   [209] sunshine > 7.9
## |   |   |   |   |   |   [210] wind_gust_speed <= 72
## |   |   |   |   |   |   |   [211] humidity_3pm <= 26
## |   |   |   |   |   |   |   |   [212] pressure_3pm <= 1004.2
## |   |   |   |   |   |   |   |   |   [213] humidity_3pm <= 18
## |   |   |   |   |   |   |   |   |   |   [214] humidity_9am <= 67
## |   |   |   |   |   |   |   |   |   |   |   [215] pressure_9am <= 1008.1: 0.3
## |   |   |   |   |   |   |   |   |   |   |   [216] pressure_9am > 1008.1: 1.80
## |   |   |   |   |   |   |   |   |   |   [217] humidity_9am > 67: 2.631 (n = 1
## |   |   |   |   |   |   |   |   |   [218] humidity_3pm > 18: 1.968 (n = 153,
## |   |   |   |   |   |   |   |   [219] pressure_3pm > 1004.2
## |   |   |   |   |   |   |   |   |   [220] sunshine <= 10.1
## |   |   |   |   |   |   |   |   |   |   [221] wind_dir_9am <= E
## |   |   |   |   |   |   |   |   |   |   |   [222] humidity_3pm <= 20: 0.497 (
## |   |   |   |   |   |   |   |   |   |   |   [223] humidity_3pm > 20: 1.525 (n
## |   |   |   |   |   |   |   |   |   |   [224] wind_dir_9am > E
## |   |   |   |   |   |   |   |   |   |   |   [225] wind_gust_speed <= 70
## |   |   |   |   |   |   |   |   |   |   |   |   [226] wind_gust_speed <= 61:
## |   |   |   |   |   |   |   |   |   |   |   |   [227] wind_gust_speed > 61
## |   |   |   |   |   |   |   |   |   |   |   |   |   [228] rainfall <= 1.8: 0
## |   |   |   |   |   |   |   |   |   |   |   |   |   [229] rainfall > 1.8: 2.5
## |   |   |   |   |   |   |   |   |   |   |   [230] wind_gust_speed > 70: 1.531
## |   |   |   |   |   |   |   |   |   [231] sunshine > 10.1: 0.117 (n = 915, er
## |   |   |   |   |   |   |   |   [232] humidity_3pm > 26
## |   |   |   |   |   |   |   |   |   [233] temp_3pm <= 29.8
## |   |   |   |   |   |   |   |   |   |   [234] cloud_3pm <= 6
## |   |   |   |   |   |   |   |   |   |   |   [235] pressure_3pm <= 1001.8
## |   |   |   |   |   |   |   |   |   |   |   |   [236] wind_speed_3pm <= 13: 8.363
## |   |   |   |   |   |   |   |   |   |   |   |   [237] wind_speed_3pm > 13
## |   |   |   |   |   |   |   |   |   |   |   |   |   [238] wind_dir_3pm <= E: 6.65
## |   |   |   |   |   |   |   |   |   |   |   |   |   [239] wind_dir_3pm > E: 1.441
## |   |   |   |   |   |   |   |   |   |   |   [240] pressure_3pm > 1001.8
## |   |   |   |   |   |   |   |   |   |   |   |   [241] humidity_3pm <= 34: 0.370 (
## |   |   |   |   |   |   |   |   |   |   |   |   [242] humidity_3pm > 34: 0.917 (n
```

```
## |   |   |   |   |   |   |   |   |   [243] cloud_3pm > 6
## |   |   |   |   |   |   |   |   |   |   [244] temp_3pm <= 25.9
## |   |   |   |   |   |   |   |   |   |   |   [245] wind_speed_9am <= 17: 2.937
## |   |   |   |   |   |   |   |   |   |   |   [246] wind_speed_9am > 17: 0.798
## |   |   |   |   |   |   |   |   |   |   [247] temp_3pm > 25.9: 5.046 (n = 60,
## |   |   |   |   |   |   |   |   [248] temp_3pm > 29.8
## |   |   |   |   |   |   |   |   |   [249] sunshine <= 9.6
## |   |   |   |   |   |   |   |   |   |   [250] wind_speed_3pm <= 24: 6.031 (n
## |   |   |   |   |   |   |   |   |   |   [251] wind_speed_3pm > 24: 1.889 (n =
## |   |   |   |   |   |   |   |   |   [252] sunshine > 9.6: 0.996 (n = 182, err
## |   |   |   |   |   |   [253] wind_gust_speed > 72
## |   |   |   |   |   |   |   [254] min_temp <= 19.3
## |   |   |   |   |   |   |   |   [255] humidity_9am <= 62
## |   |   |   |   |   |   |   |   |   [256] temp_3pm <= 32.6: 1.026 (n = 509, e
## |   |   |   |   |   |   |   |   |   [257] temp_3pm > 32.6
## |   |   |   |   |   |   |   |   |   |   [258] humidity_3pm <= 24: 1.530 (n =
## |   |   |   |   |   |   |   |   |   |   [259] humidity_3pm > 24: 9.338 (n = 1
## |   |   |   |   |   |   |   |   [260] humidity_9am > 62
## |   |   |   |   |   |   |   |   |   [261] humidity_3pm <= 43
## |   |   |   |   |   |   |   |   |   |   [262] wind_dir_3pm <= ESE: 5.975 (n =
## |   |   |   |   |   |   |   |   |   |   [263] wind_dir_3pm > ESE
## |   |   |   |   |   |   |   |   |   |   |   [264] wind_dir_3pm <= W: 0.239 (n
## |   |   |   |   |   |   |   |   |   |   |   [265] wind_dir_3pm > W: 2.013 (n
## |   |   |   |   |   |   |   |   |   [266] humidity_3pm > 43
## |   |   |   |   |   |   |   |   |   |   [267] temp_3pm <= 27.4: 4.097 (n = 40
## |   |   |   |   |   |   |   |   |   |   [268] temp_3pm > 27.4: 17.300 (n = 10
## |   |   |   |   |   |   |   [269] min_temp > 19.3
## |   |   |   |   |   |   |   |   [270] humidity_3pm <= 32: 3.560 (n = 135, err
## |   |   |   |   |   |   |   |   [271] humidity_3pm > 32: 11.947 (n = 36, err
## |   |   |   |   [272] humidity_3pm > 50
## |   |   |   |   |   [273] pressure_3pm <= 1011.1
## |   |   |   |   |   |   [274] temp_3pm <= 25.6
## |   |   |   |   |   |   |   [275] humidity_9am <= 68
## |   |   |   |   |   |   |   |   [276] pressure_3pm <= 997.4: 8.420 (n = 40, e
## |   |   |   |   |   |   |   |   [277] pressure_3pm > 997.4: 3.219 (n = 504, e
## |   |   |   |   |   |   |   [278] humidity_9am > 68
```

```
## |   |   |   |   |   |   |   |   [279] wind_speed_9am <= 24: 6.474 (n = 549, e
## |   |   |   |   |   |   |   |   [280] wind_speed_9am > 24
## |   |   |   |   |   |   |   |   |   [281] pressure_3pm <= 998.1: 11.343 (n =
## |   |   |   |   |   |   |   |   |   [282] pressure_3pm > 998.1
## |   |   |   |   |   |   |   |   |   |   [283] wind_gust_speed <= 67: 2.361 (n
## |   |   |   |   |   |   |   |   |   |   [284] wind_gust_speed > 67
## |   |   |   |   |   |   |   |   |   |   |   [285] temp_3pm <= 21.1: 4.196 (n
## |   |   |   |   |   |   |   |   |   |   |   [286] temp_3pm > 21.1: 13.756 (n
## |   |   |   |   |   |   |   [287] temp_3pm > 25.6
## |   |   |   |   |   |   |   |   [288] sunshine <= 8.3: 14.437 (n = 122, err = 426
## |   |   |   |   |   |   |   |   [289] sunshine > 8.3
## |   |   |   |   |   |   |   |   |   [290] wind_speed_3pm <= 28: 8.899 (n = 122, e
## |   |   |   |   |   |   |   |   |   [291] wind_speed_3pm > 28
## |   |   |   |   |   |   |   |   |   |   [292] rain_today in No: 0.926 (n = 88, er
## |   |   |   |   |   |   |   |   |   |   [293] rain_today in Yes: 5.897 (n = 29, e
## |   |   |   |   |   |   [294] pressure_3pm > 1011.1
## |   |   |   |   |   |   |   [295] wind_speed_3pm <= 22
## |   |   |   |   |   |   |   |   [296] temp_3pm <= 20.1
## |   |   |   |   |   |   |   |   |   [297] max_temp <= 16.5
## |   |   |   |   |   |   |   |   |   |   [298] min_temp <= 5.2: 0.727 (n = 88, err
## |   |   |   |   |   |   |   |   |   |   [299] min_temp > 5.2: 1.861 (n = 157, err
## |   |   |   |   |   |   |   |   |   [300] max_temp > 16.5: 2.927 (n = 155, err =
## |   |   |   |   |   |   |   |   [301] temp_3pm > 20.1: 8.017 (n = 191, err = NA)
## |   |   |   |   |   |   |   [302] wind_speed_3pm > 22
## |   |   |   |   |   |   |   |   [303] sunshine <= 8.5
## |   |   |   |   |   |   |   |   |   [304] sunshine <= 2.4: 4.085 (n = 120, err =
## |   |   |   |   |   |   |   |   |   [305] sunshine > 2.4
## |   |   |   |   |   |   |   |   |   |   [306] wind_speed_9am <= 24: 2.743 (n = 74
## |   |   |   |   |   |   |   |   |   |   [307] wind_speed_9am > 24: 1.705 (n = 815
## |   |   |   |   |   |   |   |   [308] sunshine > 8.5
## |   |   |   |   |   |   |   |   |   [309] rain_today in No: 0.509 (n = 213, err =
## |   |   |   |   |   |   |   |   |   [310] rain_today in Yes: 1.562 (n = 64, err =
## |   |   |   [311] humidity_3pm > 66
## |   |   |   |   [312] pressure_3pm <= 1008.9
## |   |   |   |   |   [313] cloud_3pm <= 5
## |   |   |   |   |   |   [314] wind_gust_speed <= 56
```

```
## |   |   |   |   |   |   |   [315] cloud_9am <= 3
## |   |   |   |   |   |   |   |   [316] rain_today in No: 0.783 (n = 84, err = NA)
## |   |   |   |   |   |   |   |   [317] rain_today in Yes: 2.822 (n = 37, err = 61
## |   |   |   |   |   |   |   [318] cloud_9am > 3: 4.516 (n = 931, err = NA)
## |   |   |   |   |   |   [319] wind_gust_speed > 56
## |   |   |   |   |   |   |   [320] wind_dir_3pm <= SW: 6.172 (n = 165, err = NA)
## |   |   |   |   |   |   |   [321] wind_dir_3pm > SW: 9.119 (n = 208, err = NA)
## |   |   |   |   |   [322] cloud_3pm > 5
## |   |   |   |   |   |   [323] temp_3pm <= 25
## |   |   |   |   |   |   |   [324] cloud_3pm <= 7: 5.294 (n = 510, err = NA)
## |   |   |   |   |   |   |   [325] cloud_3pm > 7
## |   |   |   |   |   |   |   |   [326] sunshine <= 4.3: 11.096 (n = 133, err = NA
## |   |   |   |   |   |   |   |   [327] sunshine > 4.3
## |   |   |   |   |   |   |   |   |   [328] pressure_3pm <= 1006.2: 7.486 (n = 148,
## |   |   |   |   |   |   |   |   |   [329] pressure_3pm > 1006.2
## |   |   |   |   |   |   |   |   |   |   [330] pressure_9am <= 1010.3
## |   |   |   |   |   |   |   |   |   |   |   [331] humidity_3pm <= 78: 1.721 (n =
## |   |   |   |   |   |   |   |   |   |   |   [332] humidity_3pm > 78: 5.133 (n = 1
## |   |   |   |   |   |   |   |   |   |   [333] pressure_9am > 1010.3: 7.341 (n = 3
## |   |   |   |   |   |   [334] temp_3pm > 25
## |   |   |   |   |   |   |   [335] humidity_3pm <= 72: 8.030 (n = 303, err = NA)
## |   |   |   |   |   |   |   [336] humidity_3pm > 72
## |   |   |   |   |   |   |   |   [337] wind_speed_3pm <= 31: 14.861 (n = 266, err
## |   |   |   |   |   |   |   |   [338] wind_speed_3pm > 31: 60.914 (n = 14, err =
## |   |   |   |   [339] pressure_3pm > 1008.9
## |   |   |   |   |   [340] wind_gust_speed <= 41
## |   |   |   |   |   |   [341] rainfall <= 2.6
## |   |   |   |   |   |   |   [342] pressure_3pm <= 1019
## |   |   |   |   |   |   |   |   [343] cloud_3pm <= 6
## |   |   |   |   |   |   |   |   |   [344] cloud_3pm <= 3: 0.737 (n = 487, err = N
## |   |   |   |   |   |   |   |   |   [345] cloud_3pm > 3
## |   |   |   |   |   |   |   |   |   |   [346] humidity_3pm <= 73
## |   |   |   |   |   |   |   |   |   |   |   [347] rain_today in No: 1.433 (n = 14
## |   |   |   |   |   |   |   |   |   |   |   [348] rain_today in Yes: 2.577 (n = 1
## |   |   |   |   |   |   |   |   |   |   [349] humidity_3pm > 73: 2.363 (n = 981,
## |   |   |   |   |   |   |   |   [350] cloud_3pm > 6
```

```
## |   |   |   |   |   |   |   |   [351] humidity_3pm <= 78: 3.016 (n = 1220, er
## |   |   |   |   |   |   |   |   [352] humidity_3pm > 78
## |   |   |   |   |   |   |   |   |   [353] max_temp <= 26.6: 3.809 (n = 190, e
## |   |   |   |   |   |   |   |   |   [354] max_temp > 26.6: 16.170 (n = 20, er
## |   |   |   |   |   |   |   [355] pressure_3pm > 1019
## |   |   |   |   |   |   |   |   [356] min_temp <= 8.7
## |   |   |   |   |   |   |   |   |   [357] pressure_3pm <= 1025.3: 0.983 (n = 820,
## |   |   |   |   |   |   |   |   |   [358] pressure_3pm > 1025.3
## |   |   |   |   |   |   |   |   |   |   [359] humidity_3pm <= 78
## |   |   |   |   |   |   |   |   |   |   |   [360] temp_9am <= 13.4: 0.326 (n = 63
## |   |   |   |   |   |   |   |   |   |   |   [361] temp_9am > 13.4: 1.221 (n = 19,
## |   |   |   |   |   |   |   |   |   |   [362] humidity_3pm > 78: 1.059 (n = 92, e
## |   |   |   |   |   |   |   |   [363] min_temp > 8.7
## |   |   |   |   |   |   |   |   |   [364] cloud_3pm <= 6
## |   |   |   |   |   |   |   |   |   |   [365] pressure_9am <= 1018.9: 7.782 (n =
## |   |   |   |   |   |   |   |   |   |   [366] pressure_9am > 1018.9: 1.140 (n = 1
## |   |   |   |   |   |   |   |   |   [367] cloud_3pm > 6: 2.003 (n = 536, err = NA
## |   |   |   |   |   |   [368] rainfall > 2.6
## |   |   |   |   |   |   |   [369] min_temp <= 16.2
## |   |   |   |   |   |   |   |   [370] wind_gust_speed <= 33
## |   |   |   |   |   |   |   |   |   [371] humidity_3pm <= 73
## |   |   |   |   |   |   |   |   |   |   [372] pressure_3pm <= 1019.2: 2.036 (n =
## |   |   |   |   |   |   |   |   |   |   [373] pressure_3pm > 1019.2: 0.736 (n = 2
## |   |   |   |   |   |   |   |   |   [374] humidity_3pm > 73
## |   |   |   |   |   |   |   |   |   |   [375] temp_3pm <= 17.1: 2.062 (n = 192, e
## |   |   |   |   |   |   |   |   |   |   [376] temp_3pm > 17.1: 4.419 (n = 92, er
## |   |   |   |   |   |   |   |   [377] wind_gust_speed > 33: 3.650 (n = 691, err =
## |   |   |   |   |   |   |   [378] min_temp > 16.2
## |   |   |   |   |   |   |   |   [379] humidity_3pm <= 75
## |   |   |   |   |   |   |   |   |   [380] cloud_3pm <= 7: 2.968 (n = 283, err = N
## |   |   |   |   |   |   |   |   |   [381] cloud_3pm > 7: 8.663 (n = 51, err = 108
## |   |   |   |   |   |   |   |   [382] humidity_3pm > 75: 7.936 (n = 167, err = NA
## |   |   |   |   |   [383] wind_gust_speed > 41
## |   |   |   |   |   |   [384] wind_gust_speed <= 57
## |   |   |   |   |   |   |   [385] sunshine <= 0.1: 10.140 (n = 85, err = NA)
## |   |   |   |   |   |   |   [386] sunshine > 0.1
```

```
## |   |   |   |   |   |   |   |   [387] wind_speed_3pm <= 19
## |   |   |   |   |   |   |   |   |   [388] temp_3pm <= 14.3
## |   |   |   |   |   |   |   |   |   |   [389] temp_3pm <= 10.5
## |   |   |   |   |   |   |   |   |   |   |   [390] min_temp <= 3.3: 0.460 (n = 60,
## |   |   |   |   |   |   |   |   |   |   |   [391] min_temp > 3.3: 1.928 (n = 39,
## |   |   |   |   |   |   |   |   |   |   [392] temp_3pm > 10.5
## |   |   |   |   |   |   |   |   |   |   |   [393] pressure_3pm <= 1014.6: 5.800 (
## |   |   |   |   |   |   |   |   |   |   |   [394] pressure_3pm > 1014.6: 2.539 (n
## |   |   |   |   |   |   |   |   |   [395] temp_3pm > 14.3: 5.820 (n = 608, err =
## |   |   |   |   |   |   |   |   [396] wind_speed_3pm > 19
## |   |   |   |   |   |   |   |   |   [397] sunshine <= 5.1: 4.879 (n = 524, err =
## |   |   |   |   |   |   |   |   |   [398] sunshine > 5.1
## |   |   |   |   |   |   |   |   |   |   [399] rain_today in No
## |   |   |   |   |   |   |   |   |   |   |   [400] humidity_3pm <= 76: 2.075 (n =
## |   |   |   |   |   |   |   |   |   |   |   [401] humidity_3pm > 76
## |   |   |   |   |   |   |   |   |   |   |   |   [402] cloud_3pm <= 5: 2.291 (n =
## |   |   |   |   |   |   |   |   |   |   |   |   [403] cloud_3pm > 5: 5.559 (n = 1
## |   |   |   |   |   |   |   |   |   |   [404] rain_today in Yes
## |   |   |   |   |   |   |   |   |   |   |   [405] min_temp <= 23.8
## |   |   |   |   |   |   |   |   |   |   |   |   [406] wind_speed_9am <= 7: 8.011
## |   |   |   |   |   |   |   |   |   |   |   |   [407] wind_speed_9am > 7
## |   |   |   |   |   |   |   |   |   |   |   |   |   [408] wind_speed_9am <= 17: 3
## |   |   |   |   |   |   |   |   |   |   |   |   |   [409] wind_speed_9am > 17
## |   |   |   |   |   |   |   |   |   |   |   |   |   |   [410] min_temp <= 15.2: 1
## |   |   |   |   |   |   |   |   |   |   |   |   |   |   [411] min_temp > 15.2: 3
## |   |   |   |   |   |   |   |   |   |   |   [412] min_temp > 23.8: 17.956 (n = 10
## |   |   |   |   |   |   [413] wind_gust_speed > 57
## |   |   |   |   |   |   |   [414] rainfall <= 3.2: 4.858 (n = 648, err = NA)
## |   |   |   |   |   |   |   [415] rainfall > 3.2
## |   |   |   |   |   |   |   |   [416] wind_gust_dir <= S: 11.218 (n = 133, err =
## |   |   |   |   |   |   |   |   [417] wind_gust_dir > S
## |   |   |   |   |   |   |   |   |   [418] humidity_9am <= 87
## |   |   |   |   |   |   |   |   |   |   [419] wind_gust_dir <= WNW: 4.092 (n = 10
## |   |   |   |   |   |   |   |   |   |   [420] wind_gust_dir > WNW: 9.543 (n = 7,
## |   |   |   |   |   |   |   |   |   [421] humidity_9am > 87
## |   |   |   |   |   |   |   |   |   |   [422] temp_9am <= 8.9: 4.394 (n = 34, err
```

```
## |   |   |   |   |   |   |   |   |   [423] temp_9am > 8.9: 10.664 (n = 54, err
## |   |   [424] humidity_3pm > 81
## |   |   |   [425] min_temp <= 11.7
## |   |   |   |   [426] pressure_3pm <= 1011.6
## |   |   |   |   |   [427] humidity_3pm <= 89
## |   |   |   |   |   |   [428] pressure_3pm <= 1002.6
## |   |   |   |   |   |   |   [429] min_temp <= 5.1: 6.438 (n = 22, err = NA)
## |   |   |   |   |   |   |   [430] min_temp > 5.1: 13.671 (n = 115, err = NA)
## |   |   |   |   |   |   [431] pressure_3pm > 1002.6
## |   |   |   |   |   |   |   [432] temp_9am <= 13.5: 7.454 (n = 361, err = NA)
## |   |   |   |   |   |   |   [433] temp_9am > 13.5: 11.831 (n = 69, err = NA)
## |   |   |   |   |   [434] humidity_3pm > 89
## |   |   |   |   |   |   [435] pressure_3pm <= 1003.7: 15.977 (n = 163, err = NA)
## |   |   |   |   |   |   [436] pressure_3pm > 1003.7
## |   |   |   |   |   |   |   [437] temp_9am <= 13.5
## |   |   |   |   |   |   |   |   [438] rainfall <= 16: 10.004 (n = 277, err = NA)
## |   |   |   |   |   |   |   |   [439] rainfall > 16: 19.680 (n = 10, err = 2621.9
## |   |   |   |   |   |   |   [440] temp_9am > 13.5: 17.274 (n = 68, err = NA)
## |   |   |   |   [441] pressure_3pm > 1011.6
## |   |   |   |   |   [442] humidity_3pm <= 90
## |   |   |   |   |   |   [443] min_temp <= 6.7
## |   |   |   |   |   |   |   [444] rainfall <= 13.2
## |   |   |   |   |   |   |   |   [445] max_temp <= 14.9
## |   |   |   |   |   |   |   |   |   [446] max_temp <= 11.8
## |   |   |   |   |   |   |   |   |   |   [447] rainfall <= 6.2
## |   |   |   |   |   |   |   |   |   |   |   [448] max_temp <= 9.8: 0.969 (n = 149
## |   |   |   |   |   |   |   |   |   |   |   [449] max_temp > 9.8
## |   |   |   |   |   |   |   |   |   |   |   |   [450] humidity_9am <= 84: 4.589 (
## |   |   |   |   |   |   |   |   |   |   |   |   [451] humidity_9am > 84: 1.804 (n
## |   |   |   |   |   |   |   |   |   |   [452] rainfall > 6.2: 4.148 (n = 24, err
## |   |   |   |   |   |   |   |   |   [453] max_temp > 11.8: 3.006 (n = 256, err =
## |   |   |   |   |   |   |   |   [454] max_temp > 14.9
## |   |   |   |   |   |   |   |   |   [455] pressure_9am <= 1016: 11.615 (n = 13, e
## |   |   |   |   |   |   |   |   |   [456] pressure_9am > 1016
## |   |   |   |   |   |   |   |   |   |   [457] wind_gust_speed <= 28: 1.127 (n = 3
## |   |   |   |   |   |   |   |   |   |   [458] wind_gust_speed > 28: 5.428 (n = 63
```

```
## |   |   |   |   |   |   |   [459] rainfall > 13.2: 8.965 (n = 17, err = 1668.4)
## |   |   |   |   |   |   [460] min_temp > 6.7
## |   |   |   |   |   |   |   [461] wind_gust_speed <= 35
## |   |   |   |   |   |   |   |   [462] pressure_3pm <= 1018.3: 5.450 (n = 154, er
## |   |   |   |   |   |   |   |   [463] pressure_3pm > 1018.3
## |   |   |   |   |   |   |   |   |   [464] min_temp <= 8.9: 1.554 (n = 131, err =
## |   |   |   |   |   |   |   |   |   [465] min_temp > 8.9: 3.215 (n = 137, err = 
## |   |   |   |   |   |   |   [466] wind_gust_speed > 35: 6.755 (n = 582, err = NA
## |   |   |   |   |   [467] humidity_3pm > 90
## |   |   |   |   |   |   [468] min_temp <= 6.9
## |   |   |   |   |   |   |   [469] wind_gust_speed <= 54
## |   |   |   |   |   |   |   |   [470] pressure_3pm <= 1014.9: 7.843 (n = 57, err
## |   |   |   |   |   |   |   |   [471] pressure_3pm > 1014.9: 3.820 (n = 519, err
## |   |   |   |   |   |   |   [472] wind_gust_speed > 54
## |   |   |   |   |   |   |   |   [473] min_temp <= 1.8
## |   |   |   |   |   |   |   |   |   [474] wind_gust_dir <= WNW: 4.368 (n = 141, 
## |   |   |   |   |   |   |   |   |   [475] wind_gust_dir > WNW: 14.445 (n = 22, er
## |   |   |   |   |   |   |   |   [476] min_temp > 1.8
## |   |   |   |   |   |   |   |   |   [477] wind_gust_dir <= ENE: 25.167 (n = 18, 
## |   |   |   |   |   |   |   |   |   [478] wind_gust_dir > ENE: 7.904 (n = 72, er
## |   |   |   |   |   |   [479] min_temp > 6.9
## |   |   |   |   |   |   |   [480] wind_gust_speed <= 80
## |   |   |   |   |   |   |   |   [481] evaporation <= 3.6
## |   |   |   |   |   |   |   |   |   [482] cloud_3pm <= 7: 4.948 (n = 131, err = 
## |   |   |   |   |   |   |   |   |   [483] cloud_3pm > 7: 8.537 (n = 93, err = NA
## |   |   |   |   |   |   |   |   [484] evaporation > 3.6
## |   |   |   |   |   |   |   |   |   [485] rainfall <= 6.4
## |   |   |   |   |   |   |   |   |   |   [486] wind_gust_speed <= 56: 9.098 (n = 
## |   |   |   |   |   |   |   |   |   |   [487] wind_gust_speed > 56: 15.994 (n = 
## |   |   |   |   |   |   |   |   |   [488] rainfall > 6.4: 16.273 (n = 89, err = 
## |   |   |   |   |   |   |   [489] wind_gust_speed > 80: 40.950 (n = 8, err = 669
## |   |   |   [490] min_temp > 11.7
## |   |   |   |   [491] humidity_3pm <= 88
## |   |   |   |   |   [492] min_temp <= 22.2
## |   |   |   |   |   |   [493] wind_gust_speed <= 44
## |   |   |   |   |   |   |   [494] cloud_3pm <= 6: 5.119 (n = 527, err = NA)
```

```
## |   |   |   |   |   |   |   [495] cloud_3pm > 6: 7.835 (n = 591, err = NA)
## |   |   |   |   |   |   [496] wind_gust_speed > 44
## |   |   |   |   |   |   |   [497] sunshine <= 0.3: 16.145 (n = 88, err = NA)
## |   |   |   |   |   |   |   [498] sunshine > 0.3: 9.266 (n = 614, err = NA)
## |   |   |   |   |   [499] min_temp > 22.2: 18.588 (n = 222, err = NA)
## |   |   |   |   [500] humidity_3pm > 88
## |   |   |   |   |   [501] min_temp <= 23.2
## |   |   |   |   |   |   [502] wind_gust_speed <= 48
## |   |   |   |   |   |   |   [503] cloud_9am <= 7
## |   |   |   |   |   |   |   |   [504] humidity_3pm <= 91: 10.262 (n = 263, err =
## |   |   |   |   |   |   |   |   [505] humidity_3pm > 91
## |   |   |   |   |   |   |   |   |   [506] min_temp <= 18.6: 12.683 (n = 488, err
## |   |   |   |   |   |   |   |   |   [507] min_temp > 18.6
## |   |   |   |   |   |   |   |   |   |   [508] rainfall <= 4.4
## |   |   |   |   |   |   |   |   |   |   |   [509] temp_3pm <= 22.8: 17.884 (n = 9
## |   |   |   |   |   |   |   |   |   |   |   [510] temp_3pm > 22.8: 3.487 (n = 30,
## |   |   |   |   |   |   |   |   |   |   [511] rainfall > 4.4: 27.840 (n = 60, err
## |   |   |   |   |   |   |   [512] cloud_9am > 7
## |   |   |   |   |   |   |   |   [513] temp_3pm <= 19.3: 15.931 (n = 323, err = NA
## |   |   |   |   |   |   |   |   [514] temp_3pm > 19.3: 22.554 (n = 175, err = NA)
## |   |   |   |   |   |   [515] wind_gust_speed > 48
## |   |   |   |   |   |   |   [516] wind_gust_dir <= S: 27.872 (n = 353, err = NA)
## |   |   |   |   |   |   |   [517] wind_gust_dir > S
## |   |   |   |   |   |   |   |   [518] min_temp <= 17
## |   |   |   |   |   |   |   |   |   [519] pressure_3pm <= 1005: 17.110 (n = 51, e
## |   |   |   |   |   |   |   |   |   [520] pressure_3pm > 1005: 10.327 (n = 111, e
## |   |   |   |   |   |   |   |   [521] min_temp > 17: 19.910 (n = 84, err = NA)
## |   |   |   |   |   [522] min_temp > 23.2
## |   |   |   |   |   |   [523] max_temp <= 28.6
## |   |   |   |   |   |   |   [524] pressure_9am <= 1007.5: 107.775 (n = 8, err = 1
## |   |   |   |   |   |   |   [525] pressure_9am > 1007.5: 36.563 (n = 34, err = NA
## |   |   |   |   |   |   [526] max_temp > 28.6: 28.633 (n = 130, err = NA)
## |   [527] rainfall > 24
## |   |   [528] humidity_3pm <= 86
## |   |   |   [529] humidity_3pm <= 75
## |   |   |   |   [530] rainfall <= 89
```

```
## |   |   |   |   |   [531] sunshine <= 6.3
## |   |   |   |   |   |   [532] temp_3pm <= 25.5: 5.931 (n = 236, err = 25679.3)
## |   |   |   |   |   |   [533] temp_3pm > 25.5: 14.468 (n = 113, err = NA)
## |   |   |   |   |   [534] sunshine > 6.3
## |   |   |   |   |   |   [535] rainfall <= 43.2
## |   |   |   |   |   |   |   [536] humidity_3pm <= 48
## |   |   |   |   |   |   |   |   [537] pressure_3pm <= 1003.7: 2.471 (n = 17, err
## |   |   |   |   |   |   |   |   [538] pressure_3pm > 1003.7: 0.446 (n = 98, err =
## |   |   |   |   |   |   |   [539] humidity_3pm > 48
## |   |   |   |   |   |   |   |   [540] wind_gust_speed <= 48: 3.523 (n = 351, err
## |   |   |   |   |   |   |   |   [541] wind_gust_speed > 48: 6.302 (n = 160, err =
## |   |   |   |   |   |   [542] rainfall > 43.2: 7.317 (n = 243, err = NA)
## |   |   |   |   [543] rainfall > 89: 17.349 (n = 50, err = NA)
## |   |   |   [544] humidity_3pm > 75
## |   |   |   |   [545] cloud_9am <= 7: 10.755 (n = 290, err = NA)
## |   |   |   |   [546] cloud_9am > 7
## |   |   |   |   |   [547] temp_3pm <= 25.8: 14.494 (n = 133, err = NA)
## |   |   |   |   |   [548] temp_3pm > 25.8: 35.914 (n = 65, err = 101726.9)
## |   |   [549] humidity_3pm > 86
## |   |   |   [550] min_temp <= 20.8
## |   |   |   |   [551] wind_speed_3pm <= 52
## |   |   |   |   |   [552] min_temp <= 15.9
## |   |   |   |   |   |   [553] rainfall <= 76.2
## |   |   |   |   |   |   |   [554] wind_gust_speed <= 54
## |   |   |   |   |   |   |   |   [555] min_temp <= 10: 7.625 (n = 40, err = 3723.9
## |   |   |   |   |   |   |   |   [556] min_temp > 10: 19.007 (n = 121, err = NA)
## |   |   |   |   |   |   |   [557] wind_gust_speed > 54: 26.775 (n = 109, err = 60
## |   |   |   |   |   |   [558] rainfall > 76.2: 45.659 (n = 18, err = NA)
## |   |   |   |   |   [559] min_temp > 15.9
## |   |   |   |   |   |   [560] humidity_3pm <= 99: 31.517 (n = 197, err = NA)
## |   |   |   |   |   |   [561] humidity_3pm > 99: 71.800 (n = 15, err = 48335.1)
## |   |   |   |   [562] wind_speed_3pm > 52: 87.533 (n = 12, err = 27945.9)
## |   |   |   [563] min_temp > 20.8
## |   |   |   |   [564] humidity_3pm <= 95
## |   |   |   |   |   [565] cloud_9am <= 7: 33.440 (n = 46, err = NA)
## |   |   |   |   |   [566] cloud_9am > 7: 63.596 (n = 73, err = NA)
```

```
## |   |   |   |   [567] humidity_3pm > 95: 101.979 (n = 40, err = NA)
##
## Number of inner nodes:    283
## Number of terminal nodes: 284
```

References

Hothorn, Torsten, Kurt Hornik, Carolin Strobl, and Achim Zeileis. 2021. *Party: A Laboratory for Recursive Partytioning*. http://party.R-forge.R-project.org.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.84 CTree Plot

```
plot(model)
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

20.85 Weka Regression Tree

Weka's RWeka::J48() does not support regression trees.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

21 Text Mining

20190221 Text Mining (or Text Analytics) applies analytic tools to learn from collections of text data, like social media, books, newspapers, emails, etc. The goal can be considered to be similar to humans learning by reading such material. However, using automated algorithms we can learn from massive amounts of text, very much more than a human can. The material could consist of millions of newspaper articles to perhaps summarise the main themes and to identify those that are of most interest to particular people. Or we might be monitoring twitter feeds to identify emerging topics that we might need to act upon, as it emerges.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

21.1 Test Mining Setup

20180603 Packages used in this chapter include RColorBrewer (Neuwirth 2014), Rgraphviz (Hansen et al. 2018), dplyr (Wickham et al. 2021), ggplot2 (Wickham et al. 2020), qdap (Rinker 2020a), qdapDictionaries (Rinker 2018), scales (Wickham and Seidel 2020), stringr (Wickham 2019b), tm (Feinerer and Hornik 2020), wordcloud (Fellows 2018), and rattle (G. Williams 2020).

Packages are loaded into the currently running R session from your local library directories on disk. Missing packages can be installed using utils::install.packages() within R. On Ubuntu, for example, R packages can also be installed using `$ wajig install r-cran-<pkgname>`.

```
# Load required packages from local library into the R session.

library(RColorBrewer) # Generate palette of colours for plots.
library(Rgraphviz)    # Correlation plots.
library(dplyr)        # Data wrangling, pipe operator %>%().
library(ggplot2)      # Plot word frequencies.
library(qdap)         # Quantitative discourse analysis of transcripts.
library(qdapDictionaries)
library(rattle)       # Support: riskchart() errorMatrix().
library(scales)       # Support: commas(), percent().
library(stringr)      # str_replace_all().
library(tm)           # Framework for text mining.
library(wordcloud)
```

References

Feinerer, Ingo, and Kurt Hornik. 2020. *Tm: Text Mining Package*. http://tm.r-forge.r-project.org/.

Fellows, Ian. 2018. *Wordcloud: Word Clouds*. https://CRAN.R-project.org/package=wordcloud.

Hansen, Kasper Daniel, Jeff Gentry, Li Long, Robert Gentleman, Seth Falcon, Florian Hahne, and Deepayan Sarkar. 2018. *Rgraphviz: Provides Plotting Capabilities for r Graph Objects*.

Neuwirth, Erich. 2014. *RColorBrewer: ColorBrewer Palettes*. https://CRAN.R-project.org/package=RColorBrewer.

Rinker, Tyler. 2018. *qdapDictionaries: Dictionaries and Word Lists for the Qdap Package*. http://trinker.github.com/qdapDictionaries/.

———. 2020a. *Qdap: Bridging the Gap Between Qualitative Data and Quantitative Analysis*. http://trinker.github.io/qdap/.

———. 2019b. *Stringr: Simple, Consistent Wrappers for Common String Operations*. https://CRAN.R-project.org/package=stringr.

Wickham, Hadley, Winston Chang, Lionel Henry, Thomas Lin Pedersen, Kohske Takahashi, Claus Wilke, Kara Woo, Hiroaki Yutani, and Dewey Dunnington. 2020. *Ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics*. https://CRAN.R-project.org/package=ggplot2.

Wickham, Hadley, Romain François, Lionel Henry, and Kirill Müller. 2021. *Dplyr: A Grammar of Data Manipulation*. https://CRAN.R-project.org/package=dplyr.

Wickham, Hadley, and Dana Seidel. 2020. *Scales: Scale Functions for Visualization*. https://CRAN.R-project.org/package=scales.

Williams, Graham. 2020. *Rattle: Graphical User Interface for Data Science in r*. https://rattle.togaware.com/.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*

Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

21.2 Corpus as a Data Set

The primary package for text mining, (Feinerer and Hornik 2020), provides a framework within which we perform our text mining. A collection of other standard R packages add value to the data processing and visualizations for text mining.

The basic concept is that of a corpus. This is a collection of texts, usually stored electronically, and from which we perform our analysis. A corpus might be a collection of news articles from Reuters or the published works of Shakespeare. Within each corpus we will have separate documents, which might be articles, stories, or book volumes. Each document is treated as a separate entity or record.

Documents which we wish to analyse come in many different formats. Quite a few formats are supported by (Feinerer and Hornik 2020), the package we will illustrate text mining with in this module. The supported formats include text, PDF, Microsoft Word, and XML.

A number of open source tools are also available to convert most document formats to text files. For our corpus used initially in this module, a collection of PDF documents were converted to text using `pdftotext` from the `xpdf` application which is available for GNU/Linux and MS/Windows and others. On GNU/Linux we can convert a folder of PDF documents to text with:

```
system("for f in *.pdf; do pdftotext -enc ASCII7 -nopgbrk $f; done")
```

The `-enc ASCII7` ensures the text is converted to ASCII since otherwise we may end up with binary characters in our text documents.

We can also convert Word documents to text using `anitword`, which is another application available for GNU/Linux.

```
system("for f in *.doc; do antiword $f; done")
```

References

Feinerer, Ingo, and Kurt Hornik. 2020. *Tm: Text Mining Package*. http://tm.r-forge.r-project.org/.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

21.3 Corpus Sources and Readers

There are a variety of sources supported by . We can use tm::getSources() to list them.

```
getSources()
```

```
## [1] "DataframeSource" "DirSource"       "URISource"       "VectorSource"
## [5] "XMLSource"       "ZipSource"
```

In addition to different kinds of sources of documents, our documents for text analysis will come in many different formats. A variety are supported by :

```
getReaders()
```

```
##  [1] "readDataframe"           "readDOC"
##  [3] "readPDF"                 "readPlain"
##  [5] "readRCV1"                "readRCV1asPlain"
##  [7] "readReut21578XML"        "readReut21578XMLasPlain"
##  [9] "readTagged"              "readXML"
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

21.4 Text Documents

We load a sample corpus of text documents. Our corpus consists of a collection of research papers all stored in the folder we identify below. To work along with us in this module, you can create your own folder called `corpus/txt` and place into that folder a collection of text documents. It does not need to be as many as we use here but a reasonable number makes it more interesting.

```
cname <- file.path(".", "corpus", "txt")
cname
```

```
## [1] "./corpus/txt"
```

We can list some of the file names.

```
length(dir(cname))
```

```
## [1] 46
```

```
dir(cname)
```

```
##  [1] "acnn96.txt"                          "adm02.txt"
##  [3] "ai02.txt"                            "ai03.txt"
##  [5] "ai97.txt"                            "atobmars.txt"
##  [7] "ausdm07.txt"                         "ctac99.txt"
##  [9] "dawak02.txt"                         "dawak02w.txt"
## [11] "dmkd03.txt"                          "eJHI06.txt"
## [13] "gjwthesis.txt"                       "hdm05.txt"
## [15] "his03.txt"                           "hwrf12.txt"
## [17] "icdm02.txt"                          "icdm08.txt"
## [19] "ijdwm2012.txt"                       "karisk.txt"
## [21] "kdd00.txt"                           "kdd05.txt"
## [23] "kddmodel.txt"                        "kddrisk.txt"
## [25] "kes05.txt"                           "kes05full.txt"
## [27] "lca04.txt"                           "medinfo04.txt"
## [29] "miningmodels.txt"                    "pakdd01.txt"
## [31] "pakdd01w.txt"                        "pakdd03.txt"
## [33] "pakdd04.txt"                         "pakdd07.txt"
## [35] "pakdd99.txt"                         "pepnet.txt"
## [37] "rfpakdd12.txt"                       "RJournal_2009-1_Guazzelli+et+al.txt"
## [39] "RJournal_2009-2_Williams.txt"        "seal98.txt"
## [41] "story.txt"                           "templates.txt"
## [43] "thim.txt"                            "titb08.txt"
## [45] "tkde05.txt"                          "tr02102.txt"
```

There are documents in this particular corpus.

After loading the (Feinerer and Hornik 2020) package into the R library we are ready to load the files from the directory as the source of the files making up the corpus, using tm::DirSource(). The source object is passed on to tm::Corpus() which loads the documents. We save the resulting collection of documents in memory, stored in a variable called `docs`.

```
docs <- Corpus(DirSource(cname))
docs
```

```
## <<SimpleCorpus>>
## Metadata:  corpus specific: 1, document level (indexed): 0
## Content:  documents: 46
```

```r
class(docs)
```

```
## [1] "SimpleCorpus" "Corpus"
```

```r
class(docs[[1]])
```

```
## [1] "PlainTextDocument" "TextDocument"
```

```r
summary(docs)
```

```
##                                     Length Class             Mode
## acnn96.txt                          2      PlainTextDocument list
## adm02.txt                           2      PlainTextDocument list
## ai02.txt                            2      PlainTextDocument list
## ai03.txt                            2      PlainTextDocument list
## ai97.txt                            2      PlainTextDocument list
## atobmars.txt                        2      PlainTextDocument list
## ausdm07.txt                         2      PlainTextDocument list
## ctac99.txt                          2      PlainTextDocument list
## dawak02.txt                         2      PlainTextDocument list
## dawak02w.txt                        2      PlainTextDocument list
## dmkd03.txt                          2      PlainTextDocument list
## eJHI06.txt                          2      PlainTextDocument list
## gjwthesis.txt                       2      PlainTextDocument list
## hdm05.txt                           2      PlainTextDocument list
## his03.txt                           2      PlainTextDocument list
## hwrf12.txt                          2      PlainTextDocument list
## icdm02.txt                          2      PlainTextDocument list
## icdm08.txt                          2      PlainTextDocument list
## ijdwm2012.txt                       2      PlainTextDocument list
## karisk.txt                          2      PlainTextDocument list
## kdd00.txt                           2      PlainTextDocument list
## kdd05.txt                           2      PlainTextDocument list
## kddmodel.txt                        2      PlainTextDocument list
## kddrisk.txt                         2      PlainTextDocument list
## kes05.txt                           2      PlainTextDocument list
## kes05full.txt                       2      PlainTextDocument list
## lca04.txt                           2      PlainTextDocument list
## medinfo04.txt                       2      PlainTextDocument list
## miningmodels.txt                    2      PlainTextDocument list
## pakdd01.txt                         2      PlainTextDocument list
## pakdd01w.txt                        2      PlainTextDocument list
## pakdd03.txt                         2      PlainTextDocument list
## pakdd04.txt                         2      PlainTextDocument list
## pakdd07.txt                         2      PlainTextDocument list
```

```
## pakdd99.txt                          2      PlainTextDocument list
## pepnet.txt                           2      PlainTextDocument list
## rfpakdd12.txt                        2      PlainTextDocument list
## RJournal_2009-1_Guazzelli+et+al.txt 2      PlainTextDocument list
## RJournal_2009-2_Williams.txt         2      PlainTextDocument list
## seal98.txt                           2      PlainTextDocument list
## story.txt                            2      PlainTextDocument list
## templates.txt                        2      PlainTextDocument list
## thim.txt                             2      PlainTextDocument list
## titb08.txt                           2      PlainTextDocument list
## tkde05.txt                           2      PlainTextDocument list
## tr02102.txt                          2      PlainTextDocument list
```

References

Feinerer, Ingo, and Kurt Hornik. 2020. *Tm: Text Mining Package*. http://tm.r-forge.r-project.org/.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

21.5 PDF Documents

If instead of text documents we have a corpus of PDF documents then we can use the tm::readPDF() reader function to convert PDF into text and have that loaded as out Corpus.

```
docs <- Corpus(DirSource(cname), readerControl=list(reader=readPDF))
```

This will use, by default, the `pdftotext` command from `xpdf` to convert the PDF into text format. The `xpdf` application needs to be installed for tm::readPDF() to work.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

21.6 Word Documents

A simple open source tool to convert Microsoft Word documents into text is `antiword`. The separate `antiword` application needs to be installed, but once it is available it is used by to convert Word documents into text for loading into R.

To load a corpus of Word documents we use the tm::readDOC() reader function:

```
docs <- Corpus(DirSource(cname), readerControl=list(reader=readDOC))
```

Once we have loaded our corpus the remainder of the processing of the corpus within R is then as follows.

The `antiword` program takes some useful command line arguments. We can pass these through to the program from tm::readDOC() by specifying them as the character string argument:

```
docs <- Corpus(DirSource(cname), readerControl=list(reader=readDOC("-r -s")))
```

Here, `-r` requests that removed text be included in the output, and `-s` requests that text hidden by Word be included.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

21.7 Exploring the Corpus

We can (and should) inspect the documents using tm::inspect(). This will assure us that data has been loaded properly and as we expect.

```
inspect(docs[16])
```

```
## <<SimpleCorpus>>
## Metadata:  corpus specific: 1, document level (indexed): 0
## Content:  documents: 1
##
##
## Hybrid weighted random forests for\nclassifying very high-dimensional data\nBaoxu
```

```
viewDocs <- function(d, n) {d %>% extract2(n) %>% as.character() %>% writeLines()}
viewDocs(docs, 16)
```

Hybrid weighted random forests for
classifying very high-dimensional data
Baoxun Xu1 , Joshua Zhexue Huang2 , Graham Williams2 and
Yunming Ye1
1
##
Department of Computer Science, Harbin Institute of Technology Shenzhen Graduate
School, Shenzhen 518055, China
2
Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhe
518055, China
Email: amusing002@gmail.com
Random forests are a popular classification method based on an ensemble of a
single type of decision trees from subspaces of data. In the literature, there
are many different types of decision tree algorithms, including C4.5, CART, and
CHAID. Each type of decision tree algorithm may capture different information
and structure. This paper proposes a hybrid weighted random forest algorithm,
simultaneously using a feature weighting method and a hybrid forest method to
classify very high dimensional data. The hybrid weighted random forest algorithm
can effectively reduce subspace size and improve classification performance
without increasing the error bound. We conduct a series of experiments on eight
high dimensional datasets to compare our method with traditional random forest
methods and other classification methods. The results show that our method
consistently outperforms these traditional methods.
Keywords: Random Forests; Hybrid Weighted Random Forest; Classification; Decisio
##
1.
##
INTRODUCTION
##
Random forests [1, 2] are a popular classification
method which builds an ensemble of a single type
of decision trees from different random subspaces of
data. The decision trees are often either built using
C4.5 [3] or CART [4], but only one type within

```
## a single random forest. In recent years, random
## forests have attracted increasing attention due to
## (1) its competitive performance compared with other
## classification methods, especially for high-dimensional
## data, (2) algorithmic intuitiveness and simplicity, and
## (3) its most important capability - "ensemble" using
## bagging [5] and stochastic discrimination [2].
## Several methods have been proposed to grow random
## forests from subspaces of data [1, 2, 6, 7, 8, 9, 10]. In
## these methods, the most popular forest construction
## procedure was proposed by Breiman [1] to first use
## bagging to generate training data subsets for building
## individual trees.
## A subspace of features is then
## randomly selected at each node to grow branches of
## a decision tree. The trees are then combined as an
## ensemble into a forest. As an ensemble learner, the
## performance of a random forest is highly dependent
## on two factors: the performance of each tree and the
## diversity of the trees in the forests [11]. Breiman
## formulated the overall performance of a set of trees as
## the average strength and proved that the generalization
##
## error of a random forest is bounded by the ratio of the
## average correlation between trees divided by the square
## of the average strength of the trees.
## For very high dimensional data, such as text data,
## there are usually a large portion of features that are
## uninformative to the classes. During this forest building
## process, informative features would have the large
## chance to be missed, if we randomly select a small
## subspace (Breiman suggested selecting log2 (M ) + 1
## features in a subspace, where M is the number of
## independent features in the data) from high dimensional
## data [12]. As a result, weak trees are created from these
## subspaces, the average strength of those trees is reduced
```

```
## and the error bound of the random forest is enlarged.
## Therefore, when a large proportion of such "weak"
## trees are generated in a random forest, the forest has a
## large likelihood to make a wrong decision which mainly
## results from those "weak" trees' classification power.
## To address this problem, we aim to optimize decision
## trees of a random forest by two strategies. One
## straightforward strategy is to enhance the classification
## performance of individual trees by a feature weighting
## method for subspace sampling [12, 13, 14]. In this
## method, feature weights are computed with respect
## to the correlations of features to the class feature
## and regarded as the probabilities of the feature to
## be selected in subspaces. This method obviously
## increases the classification performance of individual
##
## The Computer Journal, Vol. ??,
##
## No. ??,
##
## ????
##
## 2
##
## Baoxun Xu, Joshua Zhexue Huang, Graham Williams, Yunming Ye
##
## trees because the subspaces will be biased to contain
## more informative features. However, the chance of more
## correlated trees is also increased since the features with
## large weights are likely to be repeatedly selected.
## The second strategy is more straightforward: use
## several different types of decision trees for each training
## data subset, to increase the diversity of the trees,
## and then select the optimal tree as the individual
## tree classifier in the random forest model. The work
## presented here extends the algorithm developed in [15].
```

```
## Specifically, we build three different types of tree
## classifiers (C4.5, CART, and CHAID [16, 17]) for each
## training data subset. We then evaluate the performance
## of the three classifiers and select the best tree. In
## this way, we build a hybrid random forest which may
## include different types of decision trees in the ensemble.
## The added diversity of the decision trees can effectively
## improve the accuracy of each tree in the forest, and
## hence the classification performance of the ensemble.
## However, when we use this method to build the best
## random forest model for classifying high dimensional
## data, we can not be sure of what subspace size is best.
## In this paper, we propose a hybrid weighted random
## forest algorithm by simultaneously using a new feature
## weighting method together with the hybrid random
## forest method to classify high dimensional data. In
## this new random forest algorithm, we calculate feature
## weights and use weighted sampling to randomly select
## features for subspaces at each node in building different
## types of trees classifiers (C4.5, CART, and CHAID) for
## each training data subset, and select the best tree as
## the individual tree in the final ensemble model.
## Experiments were performed on 8 high dimensional
## text datasets with dimensions ranging from 2000 to
## 13195. We compared the performance of eight random
## forest methods and well-known classification methods:
## C4.5 random forest, CART random forest, CHAID
## random forest, hybrid random forest, C4.5 weighted
## random forest, CART weighted random forest, CHAID
## weighted random forest, hybrid weighted random
## forest, support vector machines [18], naive Bayes [19],
## and k-nearest neighbors [20].
## The experimental
## results show that our hybrid weighted random forest
## achieves improved classification performance over the
## ten competitive methods.
```

```
## The remainder of this paper is organized as follows.
## In Section 2, we introduce a framework for building
## a hybrid weighted random forest, and describe a new
## random forest algorithm. Section 3 summarizes four
## measures to evaluate random forest models. We present
## experimental results on 8 high dimensional text datasets
## in Section 4. Section 5 contains our conclusions.
##
## TABLE 1. Contingency table of input feature A and class
## feature Y
## Y = y1 . . .
## Y = yj . . .
## Y = yq Total
## A = a1
## 11
## ...
## 1j
## ...
## 1q
## 1*
## ..
## ..
## ..
## ..
## ..
## ..
## .
## .
## ...
## .
## .
## .
## .
## A = ai
## i1
## ...
```

```
## ij
## ...
## iq
## i*
## ..
## ..
## ..
## ..
## ..
## ..
## ...
## .
## .
## .
## .
## .
## .
## A = ap
## p1
## ...
## pj
## ...
## pq
## p*
## Total
## *1
## ...
## *j
## ...
## *q
##
##
## general framework for building hybrid random forests.
## By integrating these two methods, we propose a novel
## hybrid weighted random forest algorithm.
## 2.1.
```

```
##
## Let Y be the class (or target) feature with q distinct
## class labels yj for j = 1, * * * , q. For the purposes of
## our discussion we consider a single categorical feature
## A in dataset D with p distinct category values. We
## denote the distinct values by ai for i = 1, * * * , p.
## Numeric features can be discretized into p intervals with
## a supervised discretization method.
## Assume D has val objects. The size of the subset of
## D satisfying the condition that A = ai and Y = yj is
## denoted by ij . Considering all combinations of the
## categorical values of A and the labels of Y , we can
## obtain a contingency table [21] of A against Y as shown
## in Table 1. The far right column contains the marginal
## totals for feature A:
##
## HYBRID
## FORESTS
##
## WEIGHTED
##
## RANDOM
##
## In this section, we first introduce a feature weighting
## method for subspace sampling. Then we present a
##
## q
##
##
## i. =
##
## ij
##
## for i = 1, * * * , p
##
## (1)
```

```
##
## j=1
##
## and the bottom row is the marginal totals for class
## feature Y :
## .j =
##
## p
##
##
## ij
##
## for j = 1, * * * , q
##
## (2)
##
## i=1
##
## The grand total (the total number of samples) is in
## the bottom right corner:
## =
##
## q
## p
##
##
## ij
##
## (3)
##
## j=1 i=1
##
## Given a training dataset D and feature A we first
## compute the contingency table. The feature weights are
## then computed using the two methods to be discussed
## in the following subsection.
```

```
## 2.2.
##
## 2.
##
## Notation
##
## Feature Weighting Method
##
## In this subsection, we give the details of the feature
## weighting method for subspace sampling in random
## forests. Consider an M-dimensional feature space
## {A1 , A2 , . . . , AM }. We present how to compute the
##
## The Computer Journal, Vol. ??,
##
## No. ??,
##
## ????
##
## Hybrid weighted random forests for classifying very high-dimensional data
## weights {w1 , w2 , . . . , wM } for every feature in the space.
## These weights are then used in the improved algorithm
## to grow each decision tree in the random forest.
## 2.2.1. Feature Weight Computation
## The weight of feature A represents the correlation
## between the values of feature A and the values of the
## class feature Y . A larger weight will indicate that the
## class labels of objects in the training dataset are more
## correlated with the values of feature A, indicating that
## A is more informative to the class of objects. Thus it
## is suggested that A has a stronger power in predicting
## the classes of new objects.
## In the following, we propose to use the chi-square
## statistic to compute feature weights because this
## method can quantify the correspondence between two
## categorical variables.
```

```
## Given the contingency table of an input feature A and
## the class feature Y of dataset D, the chi-square statistic
## of the two features is computed as:
## corr(A, Y ) =
##
## q
## p
##
## (ij - tij )2
## tij
## i=1 j=1
##
## (4)
##
## where ij is the observed frequency from the
## contingency table and tij is the expected frequency
## computed as
## i* x *j
## tij =
##
##
## (5)
##
## The larger the measure corr(A, Y ), the more
## informative the feature A is in predicting class Y .
## 2.2.2. Normalized Feature Weight
## In practice, feature weights are normalized for feature
## subspace sampling. We use corr(A, Y ) to measure the
## informativeness of these features and consider them
## as feature weights. However, to treat the weights as
## probabilities of features, we normalize the measures to
## ensure the sum of the normalized feature weights is
## equal to 1. Let corr(Ai , Y ) (1  i  M ) be the set
## of M feature measures. We compute the normalized
## weights as
##
```

```
## corr(Ai , Y )
## wi = N
## (6)
## i=1 corr(Ai , Y )
## Here, we use the square root to smooth the values of
## the measures. wi can be considered as the probability
## that feature Ai is randomly sampled in a subspace. The
## more informative a feature is, the larger the weight and
## the higher the probability of the feature being selected.
##
## Diversity is commonly obtained by using bagging and
## random subspace sampling. We introduce a further
## element of diversity by using different types of trees.
## Considering an analogy with forestry, the different data subsets from bagging rep
## one is to use three types of decision tree algorithms to
## generate three different tree classifiers for each training data subset; the othe
## of each tree as the measure of tree importance. In this
## paper, we use the out-of-bag accuracy to assess the importance of a tree.
## Following Breiman [1], we use bagging to generate
## a series of training data subsets from which we build
## trees. For each tree, the data subset used to grow
## the tree is called the "in-of-bag" (IOB) data and the
## remaining data subset is called the "out-of-bag" (OOB)
## data. Since OOB data is not used for building trees
## we can use this data to objectively evaluate each tree's
## accuracy and importance. The OOB accuracy gives an
## unbiased estimate of the true accuracy of a model.
## Given n instances in a training dataset D and a tree
## classifier hk (IOBk ) built from the k'th training data
## subset IOBk , we define the OOB accuracy of the tree
## hk (IOBk ), for di  D, as:
## n
## OOBAcck =
##
## Framework for Building a Hybrid Random
## Forest
```

```
##
## As an ensemble learner, the performance of a random
## forest is highly dependent on two factors: the diversity
## among the trees and the accuracy of each tree [11].
##
## i=1
##
## I(hk (di ) = yi ; di  OOBk )
## n
## i=1 I(di  OOBk )
##
## (7)
##
## where I(.) is an indicator function. The larger the
## OOBAcck , the better the classification quality of a tree.
## We use the out-of-bag data subset OOBi to calculate
## the out-of-bag accuracies of the three types of trees
## (C4.5, CART and CHAID) with evaluation values E1 ,
## E2 and E3 respectively.
## Fig. 1 illustrates the procedure for building a hybrid
## random forest model. Firstly, a series of IOB/OOB
## datasets are generated from the entire training dataset
## by bagging. Then, three types of tree classifiers (C4.5,
## CART and CHAID) are built using each IOB dataset.
## The corresponding OOB dataset is used to calculate the
## OOB accuracies of the three tree classifiers. Finally,
## we select the tree with the highest OOB accuracy as
## the final tree classifier, which is included in the hybrid
## random forest.
## Building a hybrid random forest model in this
## way will increase the diversity among the trees.
## The classification performance of each individual tree
## classifier is also maximized.
## 2.4.
##
## 2.3.
```

```
##
## 3
##
## Decision Tree Algorithms
##
## The core of our approach is the diversity of decision
## tree algorithms in our random forest. Different decision
## tree algorithms grow structurally different trees from
## the same training data. Selecting a good decision tree
## algorithm to grow trees for a random forest is critical
##
## The Computer Journal, Vol. ??,
##
## No. ??,
##
## ????
##
## 4
##
## Baoxun Xu, Joshua Zhexue Huang, Graham Williams, Yunming Ye
## the difference lies in the way to split a node, such
## as the split functions and binary branches or multibranches. In this work we use
## tree algorithms to build a hybrid random forest.
##
## 2.5.
##
## FIGURE 1. The Hybrid Random Forests framework.
##
## for the performance of the random forest. Few studies
## have considered how different decision tree algorithms
## affect a random forest. We do so in this paper.
## The common decision tree algorithms are as follows:
## Classification Trees 4.5 (C4.5) is a supervised
## learning classification algorithm used to construct
## decision trees. Given a set of pre-classified objects, each
## described by a vector of attribute values, we construct
```

```
## a mapping from attribute values to classes. C4.5 uses
## a divide-and-conquer approach to grow decision trees.
## Beginning with the entire dataset, a tree is constructed
## by considering each predictor variable for dividing the
## dataset. The best predictor is chosen at each node
## using a impurity or diversity measure. The goal is
## to produce subsets of the data which are homogeneous
## with respect to the target variable. C4.5 selects the test
## that maximizes the information gain ratio (IGR) [3].
## Classification and Regression Tree (CART) is
## a recursive partitioning method that can be used for
## both regression and classification. The main difference
## between C4.5 and CART is the test selection and
## evaluation process.
## Chi-squared Automatic Interaction Detector
## (CHAID) method is based on the chi-square test of
## association. A CHAID decision tree is constructed
## by repeatedly splitting subsets of the space into two
## or more nodes. To determine the best split at any
## node, any allowable pair of categories of the predictor
## variables is merged until there is no statistically
## significant difference within the pair with respect to the
## target variable [16, 17].
## From these decision tree algorithms, we can see that
##
## Hybrid Weighted Random Forest Algorithm
##
## In this subsection we present a hybrid weighted
## random forest algorithm by simultaneously using the
## feature weights and a hybrid method to classify high
## dimensional data. The benefits of our algorithm has
## two aspects: Firstly, compared with hybrid forest
## method [15], we can use a small subspace size to
## create accurate random forest models.
## Secondly,
## compared with building a random forest using feature
```

```
## weighting [14], we can use several different types of
## decision trees for each training data subset to increase
## the diversities of trees. The added diversity of the
## decision trees can effectively improve the classification
## performance of the ensemble model. The detailed steps
## are introduced in Algorithm 1.
## Input parameters to Algorithm 1 include a training
## dataset D, the set of features A, the class feature Y ,
## the number of trees in the random forest K and the
## size of subspaces m. The output is a random forest
## model M . Lines 9-16 form the loop for building K
## decision trees. In the loop, Line 10 samples the training
## data D by sampling with replacement to generate an
## in-of-bag data subset IOBi for building a decision tree.
## Line 11-14 build three types of tree classifiers (C4.5,
## CART, and CHAID). In this procedure, Line 12 calls
## the function createT reej () to build a tree classifier.
## Line 13 calculates the out-of-bag accuracy of the tree
## classifier. After this procedure, Line 15 selects the tree
## classifier with the maximum out-of-bag accuracy. K
## decision tree trees are thus generated to form a hybrid
## weighted random forest model M .
## Generically, function createT reej () first creates a
## new node. Then, it tests the stopping criteria to decide
## whether to return to the upper node or to split this
## node. If we choose to split this node, then the feature
## weighting method is used to randomly select m features
## as the subspace for node splitting. These features
## are used as candidates to generate the best split to
## partition the node. For each subset of the partition,
## createT reej () is called again to create a new node under
## the current node. If a leaf node is created, it returns to
## the parent node. This recursive process continues until
## a full tree is generated.
##
## The Computer Journal, Vol. ??,
```

```
##
## No. ??,
##
## ????
##
## Hybrid weighted random forests for classifying very high-dimensional data
## Algorithm 1 New Random Forest Algorithm
## 1: Input:
## 2: - D : the training dataset,
## 3: - A : the features space {A1 , A2 , ..., AM },
## 4: - Y : the class features space {y1 , y2 , ..., yq },
## 5: - K : the number of trees,
## 6: - m : the size of subspaces.
## 7: Output: A random forest M ;
## 8: Method:
## 9: for i = 1 to K do
## 10:
## draw a bootstrap sample in-of-bag data subset
## IOBi and out-of-bag data subset OOBi from
## training dataset D;
## 11:
## for j = 1 to 3 do
## 12:
## hi,j (IOBi ) = createT reej ();
## use out-of-bag data subset OOBi to calculate
## 13:
## the out-of-bag accuracy OOBAcci, j of the tree
## classifier hi,j (IOBi ) by Equation(1);
## 14:
## end for
## 15:
## select hi (IOBi ) with the highest out-of-bag
## accuracy OOBAcci as Optimal tree i;
## 16: end for
## 17: combine
## the
```

```
## K
## tree
## classifiers
## h1 (IOB1 ), h2 (IOB2 ), ..., hK (IOBK ) into a random
## forest M ;
## 18:
## 19:
## 20:
## 21:
## 22:
## 23:
## 24:
## 25:
## 26:
## 27:
## 28:
## 29:
## 30:
## 31:
## 32:
##
## 3.
##
## Function createTree()
## create a new node N ;
## if stopping criteria is met then
## return N as a leaf node;
## else
## for j = 1 to M do
## compute
## the
## informativeness
## measure
## corr(Aj , Y ) by Equation (4);
## end for
## compute feature weights {w1 , w2 , ..., wM } by
```

```
## Equation (6);
## use the feature weighting method to randomly
## select m features;
## use these m features as candidates to generate
## the best split for the node to be partitioned;
## call createTree() for each split;
## end if
## return N ;
## EVALUATION MEASURES
##
## In this paper, we use five measures, i.e., strength,
## correlation, error bound c/s2 , test accuracy, and F1
## metric, to evaluate our random forest models. Strength
## measures the collective performance of individual trees
## in a random forest and the correlation measures the
## diversity of the trees. The ratio of the correlation
## over the square of the strength c/s2 indicates the
## generalization error bound of the random forest model.
## These three measures were introduced in [1]. The
## accuracy measures the performance of a random forest
## model on unseen test data. The F1 metric is a
##
## 5
##
## commonly used measure of classification performance.
## 3.1.
##
## Strength and Correlation Measures
##
## We follow Breiman's method described in [1] to
## calculate the strength, correlation and the ratio c/s2 .
## Following Breiman's notation, we denote strength as
## s and correlation as . Let hk (IOBk ) be the kth
## tree classifier grown from the kth training data IOBk
## sampled from D with replacement.
## Assume the
```

```
## random forest model contains K trees. The out-of-bag
## proportion of votes for di  D on class j is
## K
## I(hk (di ) = j; di
## / IOBk )
## Q(di , j) = k=1K
## (8)
## / IOBk )
## k=1 I(di
## This is the number of trees in the random forest
## which are trained without di and classify di into class
## j, divided by the number of training datasets not
## containing di .
## The strength s is computed as:
## 1
## (Q(di , yi ) - maxj=yi Q(di , j))
## n i=1
## n
##
## s=
##
## (9)
##
## where n is the number of objects in D and yi indicates
## the true class of di .
## The correlation  is computed as:
## n
## 1
## 2
## 2
## i=1 (Q(di , yi ) - maxj=yi Q(di , j)) - s
## n
## (10)
##  =
##
##
```

```
## K
## 1
## (K
## k + (pk - pk )2 )2
## k=1 pk + p
## where
##
## n
## pk =
##
## i=1
##
## I(hk (di ) = yi ; di
## / IOBk )
## n
## / IOBk )
## i=1 I(di
##
## (11)
##
## and
## n
## pk =
##
## i=1
##
## I(hk (di ) = j(di , Y ); di
## / IOBk )
## n
## I(d
##
## /
## IOB
## )
## i
## k
```

```
## i=1
##
## (12)
##
## where
## j(di , Y ) = argmaxj=yi Q(d, j)
##
## (13)
##
## is the class that obtains the maximal number of votes
## among all classes but the true class.
## 3.2.
##
## General Error Bound Measure c/s2
##
## Given the strength and correlation, the out-of-bag
## estimate of the c/s2 measure can be computed.
## An important theoretical result in Breiman's method
## is the upper bound of the generalization error of the
## random forest ensemble that is derived as
## P E  (1 - s2 )/s2
##
## (14)
##
## where  is the mean value of correlations between all
## pairs of individual classifiers and s is the strength of
## the set of individual classifiers that is estimated as the
##
## The Computer Journal, Vol. ??,
##
## No. ??,
##
## ????
##
## 6
##
```

```
## Baoxun Xu, Joshua Zhexue Huang, Graham Williams, Yunming Ye
##
## average accuracy of individual classifiers on D with
## out-of-bag evaluation. This inequality shows that the
## generalization error of a random forest is affected by
## the strength of individual classifiers and their mutual
## correlations. Therefore, Breiman defined the c/s2 ratio
## to measure a random forest as
## c/s2 = /s2
##
## (15)
##
## The smaller the ratio, the better the performance of
## the random forest. As such, c/s2 gives guidance for
## reducing the generalization error of random forests.
## 3.3.
##
## Test Accuracy
##
## The test accuracy measures the classification performance of a random forest on t
## Dt be a test data and Yt be the class labels. Given
## di  Dt , the number of votes for di on class j is
## N (di , j) =
##
## K
##
##
## I(hk (di ) = j)
##
## (16)
##
## TABLE 2.
## Summary statistic of 8 high-dimensional
## datasets
## Name
## Features
```

```
## Instances
## Classes % Minority
## Fbis
## 2000
## 2463
## 17
## 1.54
## Re0
## 2886
## 1504
## 13
## 0.73
## Re1
## 3758
## 1657
## 25
## 0.6
## Tr41
## 7454
## 878
## 10
## 1.03
## Wap
## 8460
## 1560
## 20
## 0.32
## Tr31
## 10,128
## 927
## 7
## 0.22
## La2s
## 12,432
## 3075
## 6
```

```
## 8.07
## La1s
## 13,195
## 3204
## 6
## 8.52
##
## It emphasizes the performance of a classifier on rare
## categories. Define  and  as follows:
##
## i =
##
## T Pi
## T Pi
## , i =
## (T Pi + F Pi )
## (T Pi + F Ni )
##
## (20)
##
## F 1 for each category i and the macro-averaged F1
## are computed as:
##
## k=1
##
## The test accuracy is calculated as
## F 1i =
## 1
## I(N (di , yi ) - maxj=yi N (di , j) > 0) (17)
## n i=1
##
## 2i i
## , M acroF 1 =
## i + i
##
## q
```

```
## i=1
##
## q
##
## F 1i
##
## (21)
##
## n
##
## Acc =
##
## where n is the number of objects in Dt and yi indicates
## the true class of di .
## 3.4.
##
## F1 Metric
##
## To evaluate the performance of classification methods
## in dealing with an unbalanced class distribution, we use
## the F1 metric introduced by Yang and Liu [22]. This
## measure is equal to the harmonic mean of recall ()
## and precision (). The overall F1 score of the entire
## classification problem can be computed by a microaverage and a macro-average.
## Micro-averaged F1 is computed globally over all
## classes, and emphasizes the performance of a classifier
## on common classes. Define  and  as follows:
## q
##
## q
## T Pi
## i=1 T Pi
##  = q i=1
## ,  = q
## (18)
## i=1 (T Pi + F Pi )
```

```
## i=1 (T Pi + F Ni )
## where q is the number of classes. T Pi (True Positives)
## is the number of objects correctly predicted as class i,
## F Pi (False Positives) is the number of objects that are
## predicted to belong to class i but do not. The microaveraged F1 is computed as:
## M icroF 1 =
##
## 2
## +
##
## (19)
##
## Macro-averaged F1 is first computed locally over
## each class, and then the average over all classes is taken.
##
## The larger the MicroF1 and MacroF1 values are, the
## higher the classification performance of the classifier.
## 4.
##
## EXPERIMENTS
##
## In this section, we present two experiments that
## demonstrate the effectiveness of the new random
## forest algorithm for classifying high dimensional data.
## High dimensional datasets with various sizes and
## characteristics were used in the experiments. The
## first experiment is designed to show how our proposed
## method can reduce the generalization error bound
## c/s2 , and improve test accuracy when the size of
## the selected subspace is not too large. The second
## experiment is used to demonstrate the classification
## performance of our proposed method in comparison to
## other classification methods, i.e. SVM, NB and KNN.
## 4.1.
##
## Datasets
```

```
##
## In the experiments, we used eight real-world high
## dimensional datasets. These datasets were selected
## due to their diversities in the number of features, the
## number of instances, and the number of classes. Their
## dimensionalities vary from 2000 to 13,195. Instances
## vary from 878 to 3204 and the minority class rate varies
## from 0.22% to 8.52%. In each dataset, we randomly
## select 70% of instances as the training dataset, and
## the remaining data as the test dataset. Detailed
## information of the eight datasets is listed in Table 2.
## The Fbis, Re0, Re1, Tr41, Wap, Tr31, La2s
## and La1s datasets are classical text classification
## benchmark datasets which were carefully selected and
##
## The Computer Journal, Vol. ??,
##
## No. ??,
##
## ????
##
## Hybrid weighted random forests for classifying very high-dimensional data
## preprocessed by Han and Karypis [23]. Dataset Fbis
## was compiled from the Foreign Broadcast Information
## Service TREC-5 [24]. The datasets Re0 and Re1 were
## selected from the Reuters-21578 text categorization test
## collection Distribution 1.0 [25]. The datasets Tr41 and
## Tr31 were derived from TREC-5 [24], TREC-6 [24],
## and TREC-7 [24]. Dataset Wap is from the WebACE
## project (WAP) [26]. The datasets La2s and La1s were
## selected from the Los Angeles Times for TREC-5 [24].
## The classes of these datasets were generated from the
## relevance judgment provided in these collections.
## 4.2.
##
## Performance Comparisons between Random Forest Methods
```

```
##
## The purpose of this experiment was to evaluate
## the effect of the hybrid weighted random forest
## method (H W RF) on strength, correlation, c/s2 ,
## and test accuracy.
## The eight high dimensional
## datasets were analyzed and results were compared
## with seven other random forest methods, i.e., C4.5
## random forest (C4.5 RF), CART random forest
## (CART RF), CHAID random forest (CHAID RF),
## hybrid random forest (H RF), C4.5 weighted random
## forest (C4.5 W RF), CART weighted random forest
## (CART W RF), CHAID weighted random forest
## (CHAID W RF). For each dataset, we ran each
## random forest algorithm against different sizes of the
## feature subspaces. Since the number of features in these
## datasets was very large, we started with a subspace
## of 10 features and increased the subspace by 5 more
## features each time. For a given subspace size, we built
## 100 trees for each random forest model. In order to
## obtain a stable result, we built 80 random forest models
## for each subspace size, each dataset and each algorithm,
## and computed the average values of the four measures
## of strength, correlation, c/s2 , and test accuracy as the
## final results for comparison. The performance of the
## eight random forest algorithms on the four measures
## for each of the 8 datasets is shown in Figs. 2, 3, 4, and
## 5.
## Fig. 2 plots the strength for the eight methods against
## different subspace sizes on each of the 8 datasets.
## In the same subspace, the higher the strength, the
## better the result. From the curves, we can see that
## the new algorithm (H W RF) consistently performs
## better than the seven other random forest algorithms.
## The advantages are more obvious for small subspaces.
## The new algorithm quickly achieved higher strength
```

```
## as the subspace size increases.
## The seven other
## random forest algorithms require larger subspaces to
## achieve a higher strength. These results indicate that
## the hybrid weighted random forest algorithm enables
## random forest models to achieve a higher strength
## for small subspace sizes compared to the seven other
## random forest algorithms.
## Fig. 3 plots the curves for the correlations for the
## eight random forest methods on the 8 datasets. For
##
## 7
##
## small subspace sizes, H RF, C4.5 RF, CART RF,
## and CHAID RF produce higher correlations between
## the trees on all datasets. The correlation decreases
## as the subspace size increases. For the random forest
## models the lower the correlation between the trees
## then the better the final model.
## With our new
## random forest algorithm (H W RF) a low correlation
## level was achieved with very small subspaces in all
## 8 datasets. We also note that as the subspace size
## increased the correlation level increased as well. This is
## understandable because as the subspace size increases,
## the same informative features are more likely to be
## selected repeatedly in the subspaces, increasing the
## similarity of the decision trees. Therefore, the feature
## weighting method for subspace selection works well for
## small subspaces, at least from the point of view of the
## correlation measure.
## Fig. 4 shows the error bound indicator c/s2 for the
## eight methods on the 8 datasets. From these figures
## we can observe that as the subspace size increases, c/s2
## consistently reduces. The behaviour indicates that a
## subspace size larger than log2 (M )+1 benefits all eight
```

```
## algorithms. However, the new algorithm (H W RF)
## achieved a lower level of c/s2 at subspace size of
## log2 (M ) + 1 than the seven other algorithms.
## Fig. 5 plots the curves showing the accuracy of the
## eight random forest models on the test datasets from
## the 8 datasets. We can clearly see that the new random
## forest algorithm (H W RF) outperforms the seven
## other random forest algorithms in all eight data sets.
## It can be seen that the new method is more stable
## in classification performance than other methods. In
## all of these figures, it is observed that the highest test
## accuracy is often obtained with the default subspace size
## of log2 (M ) + 1. This implies that in practice, large
## size subspaces are not necessary to grow high-quality
## trees for random forests.
## 4.3.
##
## Performance Comparisons
## Classification Methods
##
## with
##
## Other
##
## We conducted a further experimental comparison
## against three other widely used text classification
## methods: support vector machines (SVM), Naive
## Bayes (NB), and k-nearest neighbor (KNN). The
## support vector machine used a linear Kernel with a
## regularization parameter of 0.03125, which was often
## used in text categorization. For Naive Bayes, we
## adopted the multi-variate Bernoulli event model that
## is frequently used in text classification [27]. For knearest neighbor (KNN), we s
## neighbors to 13. In the experiments, we used WEKA's
## implementation for these three text classification
## methods [28]. We used a single subspace size of
```

```
## features in all eight datasets to run the random forest
## algorithms. For H RF, C4.5 RF, CART RF, and
## CHAID RF, we used a subspace size of 90 features in
## the first 6 datasets (i.e., Fbis, Re0, Re1, Tr41, Wap, and
##
## The Computer Journal, Vol. ??,
##
## No. ??,
##
## ????
##
## 8
##
## Baoxun Xu, Joshua Zhexue Huang, Graham Williams, Yunming Ye
## Fbis
##
## Re0
##
## 0.52
##
## 0.52
##
## 0.48
##
## 0.48
##
## 0.44
##
## Strength
##
## Strength
##
## 0.44
##
## 0.40
##
```

```
## 0.40
##
## H_W_RF
##
## 0.36
##
## C4.5_W_RF
## CART_W_RF
##
## 0.32
##
## H_W_RF
## C4.5_W_RF
##
## 0.36
##
## CART_W_RF
##
## CHAID_W_RF
##
## CHAID_W_RF
##
## 0.32
##
## H_RF
##
## 0.28
##
## H_RF
##
## C4.5_RF
##
## C4.5_RF
##
## 0.28
##
```

```
## CART_RF
##
## 0.24
##
## CART_RF
##
## CHAID_RF
##
## CHAID_RF
##
## 0.24
## 10
##
## 20
##
## 30
##
## 40
##
## 50
##
## 60
##
## 70
##
## 80
##
## 90
##
## 100
##
## 10
##
## 20
##
## 30
```

```
##
## 50
##
## 60
##
## 70
##
## 80
##
## 90
##
## 100
##
## Number of features
##
## Number of features
##
## Re1
##
## 0.60
##
## 40
##
## Tr41
##
## 0.8
##
## 0.55
##
## 0.7
##
## 0.50
## 0.6
##
## 0.40
##
```

```
## H_W_RF
## C4.5_W_RF
##
## 0.35
##
## CART_W_RF
##
## Strength
##
## Strength
##
## 0.45
## 0.5
## H_W_RF
## C4.5_W_RF
##
## 0.4
##
## CART_W_RF
##
## CHAID_W_RF
##
## 0.30
##
## CHAID_W_RF
##
## H_RF
##
## H_RF
##
## 0.3
##
## C4.5_RF
##
## 0.25
##
```

```
## C4.5_RF
##
## CART_RF
##
## CART_RF
##
## 0.2
##
## CHAID_RF
##
## CHAID_RF
##
## 0.20
## 10
##
## 20
##
## 30
##
## 40
##
## 50
##
## 60
##
## 70
##
## 80
##
## 90
##
## 100
##
## 10
##
## 20
```

```
##
## 30
##
## 40
##
## 50
##
## 60
##
## 70
##
## 80
##
## 90
##
## 100
##
## Number of features
##
## Number of features
##
## Wap
##
## Tr31
##
## 0.9
##
## 0.44
## 0.8
##
## 0.40
##
## 0.36
##
## Strength
##
```

```
## H_W_RF
##
## 0.28
##
## C4.5_W_RF
## CART_W_RF
##
## 0.24
##
## Strength
##
## 0.7
##
## 0.32
##
## 0.6
## H_W_RF
## C4.5_W_RF
##
## 0.5
##
## CART_W_RF
## CHAID_W_RF
##
## CHAID_W_RF
## H_RF
##
## 0.20
##
## H_RF
##
## 0.4
##
## C4.5_RF
##
## C4.5_RF
```

```
##
## CART_RF
##
## CART_RF
##
## 0.16
##
## 0.3
##
## CHAID_RF
##
## CHAID_RF
##
## 0.12
## 10
##
## 20
##
## 30
##
## 40
##
## 50
##
## 60
##
## 70
##
## 80
##
## 90
##
## 100
##
## 10
##
```

```
## 20
##
## 30
##
## La2s
##
## 60
##
## 70
##
## 80
##
## 90
##
## 100
##
## La1s
##
## 0.60
##
## 0.55
##
## 0.55
##
## 0.50
##
## 0.50
##
## 0.45
##
## 0.45
## H_W_RF
##
## 0.40
##
## C4.5_W_RF
```

```
## CART_W_RF
## CHAID_W_RF
##
## 0.35
##
## Strength
##
## Strength
##
## 50
##
## Number of features
##
## Number of features
##
## 0.60
##
## 40
##
## H_W_RF
##
## 0.40
##
## C4.5_W_RF
## CART_W_RF
##
## 0.35
##
## CHAID_W_RF
##
## H_RF
## C4.5_RF
##
## 0.30
##
## H_RF
```

```
##
## 0.30
##
## C4.5_RF
##
## CART_RF
## CHAID_RF
##
## CART_RF
##
## 0.25
##
## CHAID_RF
##
## 0.25
## 10
##
## 20
##
## 30
##
## 40
##
## 50
##
## 60
##
## 70
##
## 80
##
## 90
##
## 100
##
## 110
```

```
##
## 120
##
## 130
##
## 10
##
## 20
##
## Number of features
##
## 30
##
## 40
##
## 50
##
## 60
##
## 70
##
## 80
##
## 90
##
## 100
##
## 110
##
## 120
##
## 130
##
## Number of features
##
## FIGURE 2. Strength changes against the number of features in the subspace on the
```

```
##
## Tr31) to run the random forest algorithms, and used
## a subspace size of 120 features in the last 2 datasets
## (La2s and La1s) to run these random forest algorithms.
## For H W RF, C4.5 W RF, CART W RF, and
## CHAID W RF, we used Breiman's subspace size of
##
## log2 (M ) + 1 to run these random forest algorithms.
## This number of features provided a consistent result as
## shown in Fig. 5. In order to obtain stable results, we
## built 20 random forest models for each random forest
## algorithm and each dataset and present the average
##
## The Computer Journal, Vol. ??,
##
## No. ??,
##
## ????
##
## Hybrid weighted random forests for classifying very high-dimensional data
## Fbis
##
## 9
##
## Re0
##
## 0.216
##
## 0.285
##
## 0.208
##
## 0.270
##
## Correlation
##
```

```
## Correlation
##
## 0.255
##
## 0.200
##
## 0.240
##
## 0.192
## H_W_RF
## C4.5_W_RF
##
## 0.184
##
## CART_W_RF
## CHAID_W_RF
##
## 0.176
##
## H_W_RF
##
## 0.225
##
## C4.5_W_RF
## CART_W_RF
##
## 0.210
##
## CHAID_W_RF
##
## H_RF
## C4.5_RF
##
## 0.168
##
## H_RF
```

```
##
## 0.195
##
## C4.5_RF
##
## CART_RF
## CHAID_RF
##
## CART_RF
##
## 0.180
##
## CHAID_RF
##
## 0.160
## 10
##
## 20
##
## 30
##
## 40
##
## 50
##
## 60
##
## 70
##
## 80
##
## 90
##
## 100
##
## 10
```

```
##
## 20
##
## 30
##
## Number of features
##
## 40
##
## 50
##
## 70
##
## 80
##
## 90
##
## 100
##
## Number of features
##
## Re1
##
## 0.27
##
## 60
##
## Tr41
## 0.18
##
## 0.26
##
## 0.16
##
## 0.25
##
```

```
## 0.23
## H_W_RF
## C4.5_W_RF
##
## 0.22
##
## CART_W_RF
## CHAID_W_RF
##
## 0.21
##
## Correlation
##
## Correlation
##
## 0.24
##
## 0.14
##
## H_W_RF
##
## 0.12
##
## C4.5_W_RF
## CART_W_RF
##
## 0.10
##
## CHAID_W_RF
## H_RF
##
## H_RF
## C4.5_RF
##
## 0.20
##
```

```
## C4.5_RF
##
## 0.08
##
## CART_RF
##
## CART_RF
##
## 0.19
##
## CHAID_RF
##
## CHAID_RF
##
## 0.06
## 10
##
## 20
##
## 30
##
## 40
##
## 50
##
## 60
##
## 70
##
## 80
##
## 90
##
## 100
##
## 10
```

```
##
## 20
##
## 30
##
## 40
##
## 50
##
## 60
##
## 70
##
## 80
##
## 90
##
## 100
##
## Number of features
##
## Number of features
##
## Wap
##
## Tr31
##
## 0.27
##
## 0.14
##
## 0.26
## 0.12
##
## Correlation
##
```

```
## 0.24
##
## 0.23
##
## H_W_RF
## C4.5_W_RF
##
## 0.22
##
## CART_W_RF
## CHAID_W_RF
##
## 0.21
##
## Correlation
##
## 0.25
##
## 0.10
##
## H_W_RF
## C4.5_W_RF
##
## 0.08
##
## CART_W_RF
## CHAID_W_RF
##
## 0.06
##
## H_RF
##
## H_RF
##
## C4.5_RF
##
```

```
## 0.20
##
## C4.5_RF
##
## CART_RF
##
## CART_RF
##
## 0.04
##
## CHAID_RF
##
## 0.19
##
## 10
##
## 20
##
## 30
##
## 40
##
## 50
##
## 60
##
## 70
##
## 80
##
## 90
##
## 100
##
## CHAID_RF
##
```

```
## 10
##
## 20
##
## 30
##
## 40
##
## 50
##
## 60
##
## 70
##
## 80
##
## 90
##
## 100
##
## Number of features
##
## Number of features
##
## La2s
##
## La1s
##
## 0.162
##
## 0.165
##
## 0.156
##
## 0.160
##
```

```
## 0.155
##
## 0.150
##
## H_W_RF
##
## 0.138
##
## C4.5_W_RF
## CART_W_RF
##
## 0.132
##
## CHAID_W_RF
##
## 0.126
##
## Correlation
##
## Correlation
##
## 0.150
##
## 0.144
##
## 0.145
##
## H_W_RF
## C4.5_W_RF
##
## 0.140
##
## CART_W_RF
## CHAID_W_RF
##
## 0.135
```

```
##
## H_RF
##
## H_RF
## C4.5_RF
##
## 0.120
##
## C4.5_RF
##
## 0.130
##
## CART_RF
##
## CART_RF
## CHAID_RF
##
## CHAID_RF
##
## 0.125
##
## 0.114
## 10
##
## 20
##
## 30
##
## 40
##
## 50
##
## 60
##
## 70
##
```

```
## 80
##
## 90
##
## 100
##
## 110
##
## 120
##
## 130
##
## 10
##
## 20
##
## Number of features
##
## 30
##
## 40
##
## 50
##
## 60
##
## 70
##
## 80
##
## 90
##
## 100
##
## 110
##
```

```
## 120
##
## 130
##
## Number of features
##
## FIGURE 3. Correlation changes against the number of features in the subspace on t
##
## results, noting that the range of values are less than
## 0.005 and the hybrid trees are always more accurate.
## The comparison results of classification performance
## of eleven methods are shown in Table 3.
## The
## performance is estimated using test accuracy (Acc),
##
## Micro F1 (Mic), and Macro F1 (Mac). Boldface
## denotes best results between eleven classification
## methods.
## While the improvement is often quite
## small, there is always an improvement demonstrated.
## We observe that our proposed method (H W RF)
##
## The Computer Journal, Vol. ??,
##
## No. ??,
##
## ????
##
## 10
##
## Baoxun Xu, Joshua Zhexue Huang, Graham Williams, Yunming Ye
## Fbis
##
## 3.50
##
## Re0
```

```
## 4.0
##
## 3.15
##
## log (M)+1
##
## 3.5
##
## 2
##
## 2.80
##
## 3.0
##
## 2.45
##
## C4.5_W_RF
##
## 1.75
##
## 2.5
##
## 2
##
## H_W_RF
##
## C/S
##
## C/S
##
## 2
##
## 2.10
##
## CART_W_RF
##
```

```
## H_W_RF
## C4.5_W_RF
##
## 2.0
##
## CART_W_RF
##
## CHAID_W_RF
##
## 1.40
##
## CHAID_W_RF
##
## 1.5
##
## H_RF
##
## H_RF
##
## log (M)+1
## 2
##
## C4.5_RF
##
## 1.05
##
## C4.5_RF
##
## 1.0
##
## CART_RF
##
## CART_RF
##
## CHAID_RF
##
```

```
## 0.70
##
## CHAID_RF
##
## 0.5
## 10
##
## 20
##
## 30
##
## 40
##
## 50
##
## 60
##
## 70
##
## 80
##
## 90
##
## 100
##
## 10
##
## 20
##
## 30
##
## Number of features
##
## 70
##
## 80
```

```
##
## 90
##
## 100
##
## 3.5
##
## 4.9
##
## log (M)+1
##
## 3.0
##
## 4.2
##
## 2
##
## C/S
##
## H_W_RF
##
## 2
##
## 2.5
##
## 3.5
##
## 2
##
## 60
##
## Tr41
##
## 4.0
##
## 5.6
```

```
##
## C/S
##
## 50
##
## Number of features
##
## Re1
##
## 6.3
##
## 40
##
## C4.5_W_RF
##
## 2.8
##
## 2.0
##
## H_W_RF
## C4.5_W_RF
##
## 1.5
##
## CART_W_RF
##
## CART_W_RF
##
## CHAID_W_RF
##
## 2.1
##
## CHAID_W_RF
##
## 1.0
##
```

```
## H_RF
##
## H_RF
##
## C4.5_RF
##
## log (M)+1
##
## 1.4
##
## 2
##
## C4.5_RF
##
## 0.5
##
## CART_RF
##
## CART_RF
##
## CHAID_RF
##
## 0.7
##
## CHAID_RF
##
## 0.0
## 10
##
## 20
##
## 30
##
## 40
##
## 50
```

```
##
## 60
##
## 70
##
## 80
##
## 90
##
## 100
##
## 10
##
## 20
##
## 30
##
## 40
##
## 50
##
## 60
##
## 70
##
## 80
##
## 90
##
## 100
##
## Number of features
##
## Number of features
##
## Wap
```

```
##
## Tr31
##
## 14
##
## 1.50
##
## 12
##
## log (M)+1
##
## log (M)+1
## 2
##
## 1.25
##
## 2
##
## 10
##
## C4.5_W_RF
##
## C/S
##
## 2
##
## C/S
##
## H_W_RF
##
## 6
##
## 2
##
## 1.00
##
```

```
## 8
##
## H_W_RF
##
## 0.75
##
## C4.5_W_RF
## CART_W_RF
##
## CART_W_RF
## 0.50
##
## CHAID_W_RF
##
## 4
##
## CHAID_W_RF
## H_RF
##
## H_RF
## C4.5_RF
##
## 2
##
## C4.5_RF
##
## 0.25
##
## CART_RF
##
## CART_RF
## CHAID_RF
##
## CHAID_RF
##
## 0
```

```
##
## 10
##
## 20
##
## 30
##
## 40
##
## 50
##
## 60
##
## 70
##
## 80
##
## 90
##
## 0.00
##
## 100
##
## 10
##
## 20
##
## 30
##
## 40
##
## 50
##
## 60
##
## 70
```

```
##
## 80
##
## 90
##
## 100
##
## Number of features
##
## Number of features
##
## La1s
##
## La2s
## 2.2
##
## 1.8
##
## 2.0
## 1.6
## 1.8
## 1.4
## 1.6
## 1.2
##
## C4.5_W_RF
## CART_W_RF
##
## 0.8
##
## 2
##
## H_RF
##
## 0.6
##
```

```
## C4.5_RF
## CART_W_RF
##
## 0.4
##
## CHAID_RF
##
## 20
##
## 30
##
## 40
##
## 2
##
## H_W_RF
##
## 1.2
##
## C4.5_W_RF
## CART_W_RF
##
## 1.0
##
## CHAID_W_RF
##
## log (M)+1
##
## 10
##
## C/S
##
## C/S
##
## 2
##
```

```
## 1.4
## H_W_RF
##
## 1.0
##
## 50
##
## 60
##
## 70
##
## 80
##
## 90
##
## 100
##
## 110
##
## 120
##
## CHAID_W_RF
##
## log (M)+1
## 2
##
## 0.8
##
## H_RF
## C4.5_RF
##
## 0.6
##
## CART_RF
## CHAID_RF
##
```

```
## 0.4
##
## 130
##
## 10
##
## 20
##
## Number of features
##
## 30
##
## 40
##
## 50
##
## 60
##
## 70
##
## 80
##
## 90
##
## 100
##
## 110
##
## 120
##
## 130
##
## Number of features
##
## FIGURE 4. c/s2 changes against the number of features in the subspace on the 8 h
##
```

```
## outperformed the other classification methods in all
## datasets.
##
## 5.
##
## CONCLUSIONS
##
## In this paper, we presented a hybrid weighted random
## forest algorithm by simultaneously using a feature
## weighting method and a hybrid forest method to classify
## The Computer Journal, Vol. ??,
##
## No. ??,
##
## ????
##
## Hybrid weighted random forests for classifying very high-dimensional data
## Fbis
##
## 0.86
##
## 11
##
## Re0
##
## 0.88
##
## 0.84
## 0.84
## 0.80
##
## 0.76
##
## 0.80
## H_W_RF
## C4.5_W_RF
```

```
##
## 0.78
##
## CART_W_RF
##
## Accuracy
##
## Accuracy
##
## 0.82
##
## CHAID_W_RF
## H_RF
##
## 0.76
##
## 0.72
## H_W_RF
##
## 0.68
##
## C4.5_W_RF
## CART_W_RF
##
## 0.64
##
## CHAID_W_RF
## H_RF
##
## 0.60
##
## C4.5_RF
##
## C4.5_RF
##
## log (M)+1
```

```
##
## CART_RF
##
## 2
##
## 0.74
##
## CART_RF
##
## log (M)+1
##
## 0.56
##
## 2
##
## CHAID_RF
##
## CHAID_RF
##
## 0.52
## 10
##
## 20
##
## 30
##
## 40
##
## 50
##
## 60
##
## 70
##
## 80
##
```

```
## 90
##
## 100
##
## 10
##
## 20
##
## 30
##
## Number of features
##
## 40
##
## 50
##
## 60
##
## 70
##
## 80
##
## 90
##
## 100
##
## Number of features
##
## Re1
##
## Tr41
##
## 1.00
##
## 0.86
## 0.95
```

```
##
## 0.84
##
## log (M)+1
##
## 0.82
##
## 0.90
##
## 2
##
## 0.78
## H_W_RF
## C4.5_W_RF
##
## 0.76
##
## CART_W_RF
##
## 0.74
##
## Accuracy
##
## Accuracy
##
## 0.80
##
## CHAID_W_RF
##
## 0.85
##
## H_W_RF
##
## 0.80
##
## C4.5_W_RF
```

```
## CART_W_RF
##
## 0.75
##
## CHAID_W_RF
## H_RF
##
## H_RF
##
## 0.72
##
## log (M)+1
##
## 0.70
##
## C4.5_RF
##
## C4.5_RF
##
## 2
##
## CART_RF
##
## CART_RF
##
## 0.70
##
## 0.65
##
## CHAID_RF
##
## CHAID_RF
##
## 0.68
## 10
##
```

```
## 20
##
## 30
##
## 40
##
## 50
##
## 60
##
## 70
##
## 80
##
## 90
##
## 100
##
## 10
##
## 20
##
## 30
##
## 50
##
## 60
##
## 70
##
## 80
##
## 90
##
## 100
##
```

```
## Number of features
##
## Number of features
##
## Wap
##
## 0.84
##
## 40
##
## Tr31
##
## 1.000
##
## 0.81
##
## 0.975
##
## 0.78
##
## 0.950
##
## 0.75
##
## H_W_RF
## C4.5_W_RF
##
## 0.69
##
## CART_W_RF
##
## Accuracy
##
## Accuracy
##
## 0.925
```

```
##
## 0.72
##
## 0.900
##
## H_W_RF
## C4.5_W_RF
##
## 0.875
##
## CART_W_RF
## CHAID_W_RF
##
## CHAID_W_RF
##
## 0.66
##
## H_RF
##
## log (M)+1
## 2
##
## 0.850
##
## H_RF
##
## C4.5_RF
##
## 0.63
##
## CART_RF
##
## 0.60
##
## C4.5_RF
##
```

```
## log (M)+1
## 2
##
## 0.825
##
## CART_RF
## CHAID_RF
##
## CHAID_RF
##
## 0.800
## 10
##
## 20
##
## 30
##
## 40
##
## 50
##
## 60
##
## 70
##
## 80
##
## 90
##
## 100
##
## 10
##
## 20
##
## 30
```

```
##
## 50
##
## 60
##
## 70
##
## 80
##
## 90
##
## 100
##
## Number of features
##
## Number of features
##
## La2s
##
## 0.900
##
## 40
##
## La1s
##
## 0.88
##
## 0.885
## 0.86
## 0.870
##
## Accuracy
##
## 0.840
## H_W_RF
## C4.5_W_RF
```

```
##
## 0.825
##
## CART_W_RF
## CHAID_W_RF
##
## 0.810
##
## Accuracy
##
## 0.84
##
## 0.855
##
## 0.82
##
## H_W_RF
## C4.5_W_RF
## CART_W_RF
##
## 0.80
##
## CHAID_W_RF
## H_RF
##
## H_RF
##
## log (M)+1
##
## 0.795
##
## C4.5_RF
##
## 2
##
## C4.5_RF
```

```
##
## log (M)+1
##
## 0.78
##
## 2
##
## CART_RF
##
## CART_RF
## CHAID_RF
##
## CHAID_RF
##
## 0.780
##
## 0.76
## 10
##
## 20
##
## 30
##
## 40
##
## 50
##
## 60
##
## 70
##
## 80
##
## 90
##
## 100
```

```
##
## 110
##
## 120
##
## 130
##
## 10
##
## 20
##
## Number of features
##
## 30
##
## 40
##
## 50
##
## 60
##
## 70
##
## 80
##
## 90
##
## 100
##
## 110
##
## 120
##
## 130
##
## Number of features
```

```
##
## FIGURE 5. Test Accuracy changes against the number of features in the subspace on
##
## high dimensional data. Our algorithm not only retains
## a small subspace size (Breiman's formula log2 (M ) + 1
## for determining the subspace size) to create accurate
## random forest models, but also effectively reduces
## the upper bound of the generalization error and
##
## improves classification performance. From the results of
## experiments on various high dimensional datasets, the
## random forest generated by our new method is superior
## to other classification methods. We can use the default
## log2 (M ) + 1 subspace size and generally guarantee
##
## The Computer Journal, Vol. ??,
##
## No. ??,
##
## ????
##
## 12
##
## Baoxun Xu, Joshua Zhexue Huang, Graham Williams, Yunming Ye
##
## TABLE 3. The comparison of results
## datasets
## Dataset
## Fbis
## Measures
## Acc
## Mic
## SVM
## 0.834 0.799
## KNN
## 0.78
```

```
## 0.752
## NB
## 0.776 0.74
## H RF
## 0.853 0.816
## C4.5 RF
## 0.836 0.806
## CART RF
## 0.829 0.797
## CHAID RF
## 0.842 0.805
## H W RF
## 0.856 0.825
## C4.5 W RF
## 0.841 0.809
## CART W RF
## 0.835 0.805
## CHAID W RF
## 0.839 0.815
## Dataset
## Wap
## Measures
## Acc
## Mic
## SVM
## 0.81
## 0.772
## KNN
## 0.752 0.622
## NB
## 0.797 0.742
## H RF
## 0.815 0.805
## C4.5 RF
## 0.797 0.795
## CART RF
```

```
## 0.793 0.793
## CHAID RF
## 0.805 0.805
## H W RF
## 0.815 0.805
## C4.5 W RF
## 0.805 0.795
## CART W RF
## 0.8
## 0.792
## CHAID W RF
## 0.811 0.795
##
## (best accuracy, Micro F1, and Macro F1 results) of the eleven methods on the 8
## Re0
## Mic
## 0.795
## 0.752
## 0.741
## 0.82
## 0.802
## 0.798
## 0.8
## 0.825
## 0.815
## 0.81
## 0.812
## Tr31
## Mac
## Acc
## Mic
## 0.663 0.955 0.907
## 0.622 0.905 0.82
## 0.559 0.925 0.832
## 0.735 0.965 0.925
## 0.732 0.962 0.902
```

```
## 0.73
## 0.958 0.892
## 0.732 0.96
## 0.9
## 0.735 0.965 0.925
## 0.732 0.962 0.911
## 0.73
## 0.96
## 0.902
## 0.73
## 0.96
## 0.905
##
## Mac
## 0.76
## 0.722
## 0.706
## 0.816
## 0.806
## 0.787
## 0.805
## 0.82
## 0.815
## 0.81
## 0.812
##
## Acc
## 0.804
## 0.779
## 0.784
## 0.845
## 0.836
## 0.826
## 0.832
## 0.855
## 0.845
```

```
## 0.839
## 0.842
##
## to always produce the best models, on a variety of
## measures, by using the hybrid weighted random forest
## algorithm.
## ACKNOWLEDGEMENTS
## This research is supported in part by NSFC under
## Grant NO.61073195, and Shenzhen New Industry Development Fund under Grant NO.CXB2
## REFERENCES
## [1] Breiman, L. (2001) Random forests. Machine learning,
## 45, 5-32.
## [2] Ho, T. (1998) Random subspace method for constructing decision forests. IEEE
## Analysis and Machine Intelligence, 20, 832-844.
## [3] Quinlan, J. (1993) C4.5: Programs for machine
## learning. Morgan Kaufmann.
## [4] Breiman, L. (1984) Classification and regression trees.
## Chapman & Hall/CRC.
## [5] Breiman, L. (1996) Bagging predictors.
## Machine
## learning, 24, 123-140.
## [6] Ho, T. (1995) Random decision forests. Proceedings
## of the Third International Conference on Document
## Analysis and Recognition, pp. 278-282. IEEE.
## [7] Dietterich, T. (2000) An experimental comparison of
## three methods for constructing ensembles of decision
## trees: Bagging, boosting, and randomization. Machine
## learning, 40, 139-157.
##
## Mac
## 0.756
## 0.752
## 0.619
## 0.82
## 0.802
## 0.798
```

```
## 0.8
## 0.822
## 0.812
## 0.805
## 0.815
## Mac
## 0.87
## 0.762
## 0.81
## 0.88
## 0.87
## 0.86
## 0.852
## 0.88
## 0.87
## 0.865
## 0.855
##
## Re1
## Mic
## 0.826
## 0.668
## 0.732
## 0.832
## 0.811
## 0.808
## 0.815
## 0.836
## 0.826
## 0.818
## 0.83
## la2s
## Acc
## Mic
## 0.89
## 0.832
```

```
## 0.841 0.805
## 0.896 0.815
## 0.89
## 0.84
## 0.878 0.83
## 0.882 0.832
## 0.88
## 0.83
## 0.896 0.848
## 0.886 0.835
## 0.887 0.835
## 0.887 0.833
## Acc
## 0.829
## 0.788
## 0.816
## 0.841
## 0.825
## 0.825
## 0.838
## 0.848
## 0.838
## 0.835
## 0.84
##
## Tr41
## Mic
## 0.915
## 0.813
## 0.856
## 0.926
## 0.92
## 0.891
## 0.903
## 0.926
## 0.922
```

```
## 0.91
## 0.915
## la1s
## Mac
## Acc
## Mic
## 0.807 0.875 0.82
## 0.786 0.827 0.798
## 0.79
## 0.87
## 0.802
## 0.82
## 0.862 0.825
## 0.81
## 0.855 0.82
## 0.81
## 0.84
## 0.815
## 0.803 0.845 0.816
## 0.825 0.875 0.836
## 0.816 0.866 0.825
## 0.812 0.87
## 0.825
## 0.81
## 0.865 0.825
## Mac
## 0.706
## 0.638
## 0.58
## 0.8
## 0.781
## 0.783
## 0.795
## 0.81
## 0.795
## 0.79
```

```
## 0.8
##
## Acc
## 0.95
## 0.915
## 0.935
## 0.953
## 0.948
## 0.917
## 0.926
## 0.953
## 0.95
## 0.935
## 0.942
##
## Mac
## 0.87
## 0.765
## 0.782
## 0.895
## 0.89
## 0.88
## 0.88
## 0.895
## 0.892
## 0.88
## 0.88
## Mac
## 0.803
## 0.761
## 0.775
## 0.805
## 0.798
## 0.792
## 0.795
## 0.82
```

```
## 0.81
## 0.81
## 0.805
##
## [8] Banfield, R., Hall, L., Bowyer, K., and Kegelmeyer, W.
## (2007) A comparison of decision tree ensemble creation
## techniques. IEEE Transactions on Pattern Analysis
## and Machine Intelligence, 29, 173-180.
##
## [9] Robnik-Sikonja,
## M. (2004) Improving random forests.
## Proceedings of the 15th European Conference on
## Machine Learning, pp. 359-370. Springer.
## [10] Ho, T. (1998) C4.5 decision forests. Proceedings of
## the Fourteenth International Conference on Pattern
## Recognition, pp. 545-549. IEEE.
## [11] Dietterrich, T. (1997) Machine learning research: four
## current direction. Artificial Intelligence Magzine, 18,
## 97-136.
## [12] Amaratunga, D., Cabrera, J., and Lee, Y. (2008)
## Enriched random forests. Bioinformatics, 24, 2010-
## 2014.
## [13] Ye, Y., Li, H., Deng, X., and Huang, J. (2008)
## Feature weighting random forest for detection of hidden
## web search interfaces. The Journal of Computational
## Linguistics and Chinese Language Processing, 13, 387-
## 404.
## [14] Xu, B., Huang, J., Williams, G., Wang, Q., and
## Ye, Y. (2012) Classifying very high-dimensional data
## with random forests built from small subspaces.
## International Journal of Data Warehousing and
## Mining, 8, 45-62.
## [15] Xu, B., Huang, J., Williams, G., Li, J., and Ye, Y.
## (2012) Hybrid random forests: Advantages of mixed
## trees in classifying text data. Proceedings of the 16th
## Pacific-Asia Conference on Knowledge Discovery and
```

Data Mining. Springer.
##
The Computer Journal, Vol. ??,
##
No. ??,
##
????
##
Hybrid weighted random forests for classifying very high-dimensional data
[16] Biggs, D., De Ville, B., and Suen, E. (1991) A method
of choosing multiway partitions for classification and
decision trees. Journal of Applied Statistics, 18, 49-62.
[17] Ture, M., Kurt, I., Turhan Kurum, A., and Ozdamar,
K. (2005) Comparing classification techniques for
predicting essential hypertension. Expert Systems with
Applications, 29, 583-588.
[18] Begum, N., M.A., F., and Ren, F. (2009) Automatic text summarization using s
International Journal of Innovative Computing, Information and Control, 5, 1987-1
[19] Chen, J., Huang, H., Tian, S., and Qu, Y. (2009)
Feature selection for text classification with naive
bayes. Expert Systems with Applications, 36, 5432-
5435.
[20] Tan, S. (2005) Neighbor-weighted k-nearest neighbor
for unbalanced text corpus.
Expert Systems with
Applications, 28, 667-671.
[21] Pearson, K. (1904) On the Theory of Contingency and
Its Relation to Association and Normal Correlation.
Cambridge University Press.
[22] Yang, Y. and Liu, X. (1999) A re-examination of
text categorization methods. Proceedings of the 22th
International Conference on Research and Development
in Information Retrieval, pp. 42-49. ACM.
[23] Han, E. and Karypis, G. (2000) Centroid-based
document classification: Analysis and experimental
results. Proceedings of the 4th European Conference on

```
## Principles of Data Mining and Knowledge Discovery,
## pp. 424-431. Springer.
## [24] TREC.
## (2011)
## Text
## retrieval
## conference,
## http://trec.nist.gov.
## [25] Lewis,
## D.
## (1999)
## Reuters-21578
## text
## categorization
## test
## collection
## distribution
## 1.0,
## http://www.research.att.com/ lewis.
## [26] Han, E., Boley, D., Gini, M., Gross, R., Hastings,
## K., Karypis, G., Kumar, V., Mobasher, B., and
## Moore, J. (1998) Webace: A web agent for document
## categorization and exploration. Proceedings of the 2nd
## International Conference on Autonomous Agents, pp.
## 408-415. ACM.
## [27] McCallum, A. and Nigam, K. (1998) A comparison of
## event models for naive bayes text classification. AAAI98 workshop on learning fo
## 48.
## [28] Witten, I., Frank, E., and Hall, M. (2011) Data Mining:
## Practical machine learning tools and techniques.
## Morgan Kaufmann.
##
## The Computer Journal, Vol. ??,
##
## No. ??,
##
```

```
## ????
##
## 13
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

21.8 Preparing the Corpus

We generally need to perform some pre-processing of the text data to prepare for the text analysis. Example transformations include converting the text to lower case, removing numbers and punctuation, removing stop words, stemming and identifying synonyms. The basic transforms are all available within .

```
getTransformations()
```

```
## [1] "removeNumbers"     "removePunctuation" "removeWords"
## [4] "stemDocument"      "stripWhitespace"
```

The function tm::tm_map() is used to apply one of these transformations across all documents within a corpus. Other transformations can be implemented using R functions and wrapped within tm::content_transformer() to create a function that can be passed through to tm::tm_map(). We will see an example of that in the next section.

In the following sections we will apply each of the transformations, one-by-one, to remove unwanted characters from the text.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

21.9 Simple Transforms

We start with some manual special transforms we may want to do. For example, we might want to replace `/'', used sometimes to separate alternative words, with a space. This will avoid the two words being run into one string of characters through the transformations. We might also replace` @'' and '' with a space, for the same reason.

To create a custom transformation we make use of tm::content_transformer() to create a function to achieve the transformation, and then apply it to the corpus using tm::tm_map().

```
toSpace <- content_transformer(function(x, pattern) gsub(pattern, " ", x))
docs <- tm_map(docs, toSpace, "/")
docs <- tm_map(docs, toSpace, "@")
docs <- tm_map(docs, toSpace, "\\|")
```

This can be done with a single call:

```
docs <- tm_map(docs, toSpace, "/|@|\\|")
```

Check the email address in the following.

```
inspect(docs[16])
```

```
## <<SimpleCorpus>>
## Metadata:  corpus specific: 1, document level (indexed): 0
## Content:  documents: 1
##
##
## Hybrid weighted random forests for\nclassifying very high-dimensional data\nBaoxu
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

21.10 Conversion to Lower Case

```
docs <- tm_map(docs, content_transformer(tolower))
```

```
inspect(docs[16])
```

```
## <<SimpleCorpus>>
## Metadata:  corpus specific: 1, document level (indexed): 0
## Content:  documents: 1
##
##
## hybrid weighted random forests for\nclassifying very high-dimensional data\nbaoxu
```

General character processing functions in R can be used to transform our corpus. A common requirement is to map the documents to lower case, using base::tolower(). As above, we need to wrap such functions with a tm::content_transformer():

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

21.11 Remove Numbers

```
docs <- tm_map(docs, removeNumbers)
```

```
viewDocs(docs, 16)
```

```
## hybrid weighted random forests for
## classifying very high-dimensional data
## baoxun xu , joshua zhexue huang , graham williams and
## yunming ye
##
##
## department of computer science, harbin institute of technology shenzhen graduate
## school, shenzhen , china
##
## shenzhen institutes of advanced technology, chinese academy of sciences, shenzhen
## , china
## email: amusing gmail.com
## random forests are a popular classification method based on an ensemble of a
## single type of decision trees from subspaces of data. in the literature, there
## are many different types of decision tree algorithms, including c., cart, and
## chaid. each type of decision tree algorithm may capture different information
## and structure. this paper proposes a hybrid weighted random forest algorithm,
## simultaneously using a feature weighting method and a hybrid forest method to
## classify very high dimensional data. the hybrid weighted random forest algorithm
## can effectively reduce subspace size and improve classification performance
## without increasing the error bound. we conduct a series of experiments on eight
## high dimensional datasets to compare our method with traditional random forest
## methods and other classification methods. the results show that our method
## consistently outperforms these traditional methods.
## keywords: random forests; hybrid weighted random forest; classification; decision
##
## .
##
## introduction
##
## random forests [, ] are a popular classification
## method which builds an ensemble of a single type
## of decision trees from different random subspaces of
## data. the decision trees are often either built using
## c. [] or cart [], but only one type within
```

```
## a single random forest. in recent years, random
## forests have attracted increasing attention due to
## () its competitive performance compared with other
## classification methods, especially for high-dimensional
## data, () algorithmic intuitiveness and simplicity, and
## () its most important capability - "ensemble" using
## bagging [] and stochastic discrimination [].
## several methods have been proposed to grow random
## forests from subspaces of data [, , , , , , ]. in
## these methods, the most popular forest construction
## procedure was proposed by breiman [] to first use
## bagging to generate training data subsets for building
## individual trees.
## a subspace of features is then
## randomly selected at each node to grow branches of
## a decision tree. the trees are then combined as an
## ensemble into a forest. as an ensemble learner, the
## performance of a random forest is highly dependent
## on two factors: the performance of each tree and the
## diversity of the trees in the forests []. breiman
## formulated the overall performance of a set of trees as
## the average strength and proved that the generalization
##
## error of a random forest is bounded by the ratio of the
## average correlation between trees divided by the square
## of the average strength of the trees.
## for very high dimensional data, such as text data,
## there are usually a large portion of features that are
## uninformative to the classes. during this forest building
## process, informative features would have the large
## chance to be missed, if we randomly select a small
## subspace (breiman suggested selecting log (m ) +
## features in a subspace, where m is the number of
## independent features in the data) from high dimensional
## data []. as a result, weak trees are created from these
## subspaces, the average strength of those trees is reduced
```

```
## and the error bound of the random forest is enlarged.
## therefore, when a large proportion of such "weak"
## trees are generated in a random forest, the forest has a
## large likelihood to make a wrong decision which mainly
## results from those "weak" trees' classification power.
## to address this problem, we aim to optimize decision
## trees of a random forest by two strategies. one
## straightforward strategy is to enhance the classification
## performance of individual trees by a feature weighting
## method for subspace sampling [, , ]. in this
## method, feature weights are computed with respect
## to the correlations of features to the class feature
## and regarded as the probabilities of the feature to
## be selected in subspaces. this method obviously
## increases the classification performance of individual
##
## the computer journal, vol. ??,
##
## no. ??,
##
## ????
##
##
##
## baoxun xu, joshua zhexue huang, graham williams, yunming ye
##
## trees because the subspaces will be biased to contain
## more informative features. however, the chance of more
## correlated trees is also increased since the features with
## large weights are likely to be repeatedly selected.
## the second strategy is more straightforward: use
## several different types of decision trees for each training
## data subset, to increase the diversity of the trees,
## and then select the optimal tree as the individual
## tree classifier in the random forest model. the work
## presented here extends the algorithm developed in [].
```

```
## specifically, we build three different types of tree
## classifiers (c., cart, and chaid [, ]) for each
## training data subset. we then evaluate the performance
## of the three classifiers and select the best tree. in
## this way, we build a hybrid random forest which may
## include different types of decision trees in the ensemble.
## the added diversity of the decision trees can effectively
## improve the accuracy of each tree in the forest, and
## hence the classification performance of the ensemble.
## however, when we use this method to build the best
## random forest model for classifying high dimensional
## data, we can not be sure of what subspace size is best.
## in this paper, we propose a hybrid weighted random
## forest algorithm by simultaneously using a new feature
## weighting method together with the hybrid random
## forest method to classify high dimensional data. in
## this new random forest algorithm, we calculate feature
## weights and use weighted sampling to randomly select
## features for subspaces at each node in building different
## types of trees classifiers (c., cart, and chaid) for
## each training data subset, and select the best tree as
## the individual tree in the final ensemble model.
## experiments were performed on  high dimensional
## text datasets with dimensions ranging from  to
## . we compared the performance of eight random
## forest methods and well-known classification methods:
## c. random forest, cart random forest, chaid
## random forest, hybrid random forest, c. weighted
## random forest, cart weighted random forest, chaid
## weighted random forest, hybrid weighted random
## forest, support vector machines [], naive bayes [],
## and k-nearest neighbors [].
## the experimental
## results show that our hybrid weighted random forest
## achieves improved classification performance over the
## ten competitive methods.
```

```
## the remainder of this paper is organized as follows.
## in section , we introduce a framework for building
## a hybrid weighted random forest, and describe a new
## random forest algorithm. section  summarizes four
## measures to evaluate random forest models. we present
## experimental results on  high dimensional text datasets
## in section . section  contains our conclusions.
##
## table . contingency table of input feature a and class
## feature y
## y = y . . .
## y = yj . . .
## y = yq total
## a = a
##
## ...
## j
## ...
## q
## *
## ..
## ..
## ..
## ..
## ..
## ..
## .
## .
## ...
## .
## .
## .
## .
## a = ai
## i
## ...
```

```
## ij
## ...
## iq
## i*
## ..
## ..
## ..
## ..
## ..
## ..
## ...
## .
## .
## .
## .
## .
## .
## a = ap
## p
## ...
## pj
## ...
## pq
## p*
## total
## *
## ...
## *j
## ...
## *q
##
##
## general framework for building hybrid random forests.
## by integrating these two methods, we propose a novel
## hybrid weighted random forest algorithm.
## ..
```

```
##
## let y be the class (or target) feature with q distinct
## class labels yj for j = , * * * , q. for the purposes of
## our discussion we consider a single categorical feature
## a in dataset d with p distinct category values. we
## denote the distinct values by ai for i = , * * * , p.
## numeric features can be discretized into p intervals with
## a supervised discretization method.
## assume d has val objects. the size of the subset of
## d satisfying the condition that a = ai and y = yj is
## denoted by ij . considering all combinations of the
## categorical values of a and the labels of y , we can
## obtain a contingency table [] of a against y as shown
## in table . the far right column contains the marginal
## totals for feature a:
##
## hybrid
## forests
##
## weighted
##
## random
##
## in this section, we first introduce a feature weighting
## method for subspace sampling. then we present a
##
## q
##
##
## i. =
##
## ij
##
## for i = , * * * , p
##
## ()
```

```
##
## j=
##
## and the bottom row is the marginal totals for class
## feature y :
## .j =
##
## p
##
##
## ij
##
## for j = , * * * , q
##
## ()
##
## i=
##
## the grand total (the total number of samples) is in
## the bottom right corner:
## =
##
## q
## p
##
##
## ij
##
## ()
##
## j= i=
##
## given a training dataset d and feature a we first
## compute the contingency table. the feature weights are
## then computed using the two methods to be discussed
## in the following subsection.
```

```
## ..
##
## .
##
## notation
##
## feature weighting method
##
## in this subsection, we give the details of the feature
## weighting method for subspace sampling in random
## forests. consider an m-dimensional feature space
## {a , a , . . . , am }. we present how to compute the
##
## the computer journal, vol. ??,
##
## no. ??,
##
## ????
##
## hybrid weighted random forests for classifying very high-dimensional data
## weights {w , w , . . . , wm } for every feature in the space.
## these weights are then used in the improved algorithm
## to grow each decision tree in the random forest.
## ... feature weight computation
## the weight of feature a represents the correlation
## between the values of feature a and the values of the
## class feature y . a larger weight will indicate that the
## class labels of objects in the training dataset are more
## correlated with the values of feature a, indicating that
## a is more informative to the class of objects. thus it
## is suggested that a has a stronger power in predicting
## the classes of new objects.
## in the following, we propose to use the chi-square
## statistic to compute feature weights because this
## method can quantify the correspondence between two
## categorical variables.
```

```
## given the contingency table of an input feature a and
## the class feature y of dataset d, the chi-square statistic
## of the two features is computed as:
## corr(a, y ) =
##
## q
## p
##
## (ij - tij )
## tij
## i= j=
##
## ()
##
## where ij is the observed frequency from the
## contingency table and tij is the expected frequency
## computed as
## i* x *j
## tij =
##
##
## ()
##
## the larger the measure corr(a, y ), the more
## informative the feature a is in predicting class y .
## ... normalized feature weight
## in practice, feature weights are normalized for feature
## subspace sampling. we use corr(a, y ) to measure the
## informativeness of these features and consider them
## as feature weights. however, to treat the weights as
## probabilities of features, we normalize the measures to
## ensure the sum of the normalized feature weights is
## equal to . let corr(ai , y ) (  i  m ) be the set
## of m feature measures. we compute the normalized
## weights as
##
```

```
## corr(ai , y )
## wi = n
## ()
## i= corr(ai , y )
## here, we use the square root to smooth the values of
## the measures. wi can be considered as the probability
## that feature ai is randomly sampled in a subspace. the
## more informative a feature is, the larger the weight and
## the higher the probability of the feature being selected.
##
## diversity is commonly obtained by using bagging and
## random subspace sampling. we introduce a further
## element of diversity by using different types of trees.
## considering an analogy with forestry, the different data subsets from bagging rep
## one is to use three types of decision tree algorithms to
## generate three different tree classifiers for each training data subset; the othe
## of each tree as the measure of tree importance. in this
## paper, we use the out-of-bag accuracy to assess the importance of a tree.
## following breiman [], we use bagging to generate
## a series of training data subsets from which we build
## trees. for each tree, the data subset used to grow
## the tree is called the "in-of-bag" (iob) data and the
## remaining data subset is called the "out-of-bag" (oob)
## data. since oob data is not used for building trees
## we can use this data to objectively evaluate each tree's
## accuracy and importance. the oob accuracy gives an
## unbiased estimate of the true accuracy of a model.
## given n instances in a training dataset d and a tree
## classifier hk (iobk ) built from the k'th training data
## subset iobk , we define the oob accuracy of the tree
## hk (iobk ), for di  d, as:
## n
## oobacck =
##
## framework for building a hybrid random
## forest
```

```
##
## as an ensemble learner, the performance of a random
## forest is highly dependent on two factors: the diversity
## among the trees and the accuracy of each tree [].
##
## i=
##
## i(hk (di ) = yi ; di  oobk )
## n
## i= i(di  oobk )
##
## ()
##
## where i(.) is an indicator function. the larger the
## oobacck , the better the classification quality of a tree.
## we use the out-of-bag data subset oobi to calculate
## the out-of-bag accuracies of the three types of trees
## (c., cart and chaid) with evaluation values e ,
## e and e respectively.
## fig.  illustrates the procedure for building a hybrid
## random forest model. firstly, a series of iob oob
## datasets are generated from the entire training dataset
## by bagging. then, three types of tree classifiers (c.,
## cart and chaid) are built using each iob dataset.
## the corresponding oob dataset is used to calculate the
## oob accuracies of the three tree classifiers. finally,
## we select the tree with the highest oob accuracy as
## the final tree classifier, which is included in the hybrid
## random forest.
## building a hybrid random forest model in this
## way will increase the diversity among the trees.
## the classification performance of each individual tree
## classifier is also maximized.
## ..
##
## ..
```

```
##
##
##
## decision tree algorithms
##
## the core of our approach is the diversity of decision
## tree algorithms in our random forest. different decision
## tree algorithms grow structurally different trees from
## the same training data. selecting a good decision tree
## algorithm to grow trees for a random forest is critical
##
## the computer journal, vol. ??,
##
## no. ??,
##
## ????
##
##
##
## baoxun xu, joshua zhexue huang, graham williams, yunming ye
## the difference lies in the way to split a node, such
## as the split functions and binary branches or multibranches. in this work we use
## tree algorithms to build a hybrid random forest.
##
## ..
##
## figure . the hybrid random forests framework.
##
## for the performance of the random forest. few studies
## have considered how different decision tree algorithms
## affect a random forest. we do so in this paper.
## the common decision tree algorithms are as follows:
## classification trees . (c.) is a supervised
## learning classification algorithm used to construct
## decision trees. given a set of pre-classified objects, each
## described by a vector of attribute values, we construct
```

```
## a mapping from attribute values to classes. c. uses
## a divide-and-conquer approach to grow decision trees.
## beginning with the entire dataset, a tree is constructed
## by considering each predictor variable for dividing the
## dataset. the best predictor is chosen at each node
## using a impurity or diversity measure. the goal is
## to produce subsets of the data which are homogeneous
## with respect to the target variable. c. selects the test
## that maximizes the information gain ratio (igr) [].
## classification and regression tree (cart) is
## a recursive partitioning method that can be used for
## both regression and classification. the main difference
## between c. and cart is the test selection and
## evaluation process.
## chi-squared automatic interaction detector
## (chaid) method is based on the chi-square test of
## association. a chaid decision tree is constructed
## by repeatedly splitting subsets of the space into two
## or more nodes. to determine the best split at any
## node, any allowable pair of categories of the predictor
## variables is merged until there is no statistically
## significant difference within the pair with respect to the
## target variable [, ].
## from these decision tree algorithms, we can see that
##
## hybrid weighted random forest algorithm
##
## in this subsection we present a hybrid weighted
## random forest algorithm by simultaneously using the
## feature weights and a hybrid method to classify high
## dimensional data. the benefits of our algorithm has
## two aspects: firstly, compared with hybrid forest
## method [], we can use a small subspace size to
## create accurate random forest models.
## secondly,
## compared with building a random forest using feature
```

```
## weighting [], we can use several different types of
## decision trees for each training data subset to increase
## the diversities of trees. the added diversity of the
## decision trees can effectively improve the classification
## performance of the ensemble model. the detailed steps
## are introduced in algorithm .
## input parameters to algorithm  include a training
## dataset d, the set of features a, the class feature y ,
## the number of trees in the random forest k and the
## size of subspaces m. the output is a random forest
## model m . lines - form the loop for building k
## decision trees. in the loop, line  samples the training
## data d by sampling with replacement to generate an
## in-of-bag data subset iobi for building a decision tree.
## line - build three types of tree classifiers (c.,
## cart, and chaid). in this procedure, line  calls
## the function createt reej () to build a tree classifier.
## line  calculates the out-of-bag accuracy of the tree
## classifier. after this procedure, line  selects the tree
## classifier with the maximum out-of-bag accuracy. k
## decision tree trees are thus generated to form a hybrid
## weighted random forest model m .
## generically, function createt reej () first creates a
## new node. then, it tests the stopping criteria to decide
## whether to return to the upper node or to split this
## node. if we choose to split this node, then the feature
## weighting method is used to randomly select m features
## as the subspace for node splitting. these features
## are used as candidates to generate the best split to
## partition the node. for each subset of the partition,
## createt reej () is called again to create a new node under
## the current node. if a leaf node is created, it returns to
## the parent node. this recursive process continues until
## a full tree is generated.
##
## the computer journal, vol. ??,
```

```
##
## no. ??,
##
## ????
##
## hybrid weighted random forests for classifying very high-dimensional data
## algorithm  new random forest algorithm
## : input:
## : - d : the training dataset,
## : - a : the features space {a , a , ..., am },
## : - y : the class features space {y , y , ..., yq },
## : - k : the number of trees,
## : - m : the size of subspaces.
## : output: a random forest m ;
## : method:
## : for i =  to k do
## :
## draw a bootstrap sample in-of-bag data subset
## iobi and out-of-bag data subset oobi from
## training dataset d;
## :
## for j =  to  do
## :
## hi,j (iobi ) = createt reej ();
## use out-of-bag data subset oobi to calculate
## :
## the out-of-bag accuracy oobacci, j of the tree
## classifier hi,j (iobi ) by equation();
## :
## end for
## :
## select hi (iobi ) with the highest out-of-bag
## accuracy oobacci as optimal tree i;
## : end for
## : combine
## the
```

```
## k
## tree
## classifiers
## h (iob ), h (iob ), ..., hk (iobk ) into a random
## forest m ;
## :
## :
## :
## :
## :
## :
## :
## :
## :
## :
## :
## :
## :
## :
##
## .
##
## function createtree()
## create a new node n ;
## if stopping criteria is met then
## return n as a leaf node;
## else
## for j =  to m do
## compute
## the
## informativeness
## measure
## corr(aj , y ) by equation ();
## end for
## compute feature weights {w , w , ..., wm } by
```

```
## equation ();
## use the feature weighting method to randomly
## select m features;
## use these m features as candidates to generate
## the best split for the node to be partitioned;
## call createtree() for each split;
## end if
## return n ;
## evaluation measures
##
## in this paper, we use five measures, i.e., strength,
## correlation, error bound c s , test accuracy, and f
## metric, to evaluate our random forest models. strength
## measures the collective performance of individual trees
## in a random forest and the correlation measures the
## diversity of the trees. the ratio of the correlation
## over the square of the strength c s indicates the
## generalization error bound of the random forest model.
## these three measures were introduced in []. the
## accuracy measures the performance of a random forest
## model on unseen test data. the f metric is a
##
##
##
## commonly used measure of classification performance.
## ..
##
## strength and correlation measures
##
## we follow breiman's method described in [] to
## calculate the strength, correlation and the ratio c s .
## following breiman's notation, we denote strength as
## s and correlation as . let hk (iobk ) be the kth
## tree classifier grown from the kth training data iobk
## sampled from d with replacement.
## assume the
```

```
## random forest model contains k trees. the out-of-bag
## proportion of votes for di  d on class j is
## k
## i(hk (di ) = j; di
##   iobk )
## q(di , j) = k=k
## ()
##   iobk )
## k= i(di
## this is the number of trees in the random forest
## which are trained without di and classify di into class
## j, divided by the number of training datasets not
## containing di .
## the strength s is computed as:
##
## (q(di , yi ) - maxj=yi q(di , j))
## n i=
## n
##
## s=
##
## ()
##
## where n is the number of objects in d and yi indicates
## the true class of di .
## the correlation  is computed as:
## n
##
##
##
## i= (q(di , yi ) - maxj=yi q(di , j)) - s
## n
## ()
##  =
##
##
```

```
## k
##
## (k
## k + (pk - pk ) )
## k= pk + p
## where
##
## n
## pk =
##
## i=
##
## i(hk (di ) = yi ; di
##   iobk )
## n
##   iobk )
## i= i(di
##
## ()
##
## and
## n
## pk =
##
## i=
##
## i(hk (di ) = j(di , y ); di
##   iobk )
## n
## i(d
##
##
## iob
## )
## i
## k
```

```
## i=
##
## ()
##
## where
## j(di , y ) = argmaxj=yi q(d, j)
##
## ()
##
## is the class that obtains the maximal number of votes
## among all classes but the true class.
## ..
##
## general error bound measure c s
##
## given the strength and correlation, the out-of-bag
## estimate of the c s measure can be computed.
## an important theoretical result in breiman's method
## is the upper bound of the generalization error of the
## random forest ensemble that is derived as
## p e  ( - s ) s
##
## ()
##
## where  is the mean value of correlations between all
## pairs of individual classifiers and s is the strength of
## the set of individual classifiers that is estimated as the
##
## the computer journal, vol. ??,
##
## no. ??,
##
## ????
##
##
##
```

```
## baoxun xu, joshua zhexue huang, graham williams, yunming ye
##
## average accuracy of individual classifiers on d with
## out-of-bag evaluation. this inequality shows that the
## generalization error of a random forest is affected by
## the strength of individual classifiers and their mutual
## correlations. therefore, breiman defined the c s ratio
## to measure a random forest as
## c s =  s
##
## ()
##
## the smaller the ratio, the better the performance of
## the random forest. as such, c s gives guidance for
## reducing the generalization error of random forests.
## ..
##
## test accuracy
##
## the test accuracy measures the classification performance of a random forest on t
## dt be a test data and yt be the class labels. given
## di  dt , the number of votes for di on class j is
## n (di , j) =
##
## k
##
##
## i(hk (di ) = j)
##
## ()
##
## table .
## summary statistic of  high-dimensional
## datasets
## name
## features
```

```
## instances
## classes % minority
## fbis
##
##
##
## .
## re
##
##
##
## .
## re
##
##
##
## .
## tr
##
##
##
## .
## wap
##
##
##
## .
## tr
## ,
##
##
## .
## las
## ,
##
##
```

```
## .
## las
## ,
##
##
## .
##
## it emphasizes the performance of a classifier on rare
## categories. define  and  as follows:
##
## i =
##
## t pi
## t pi
## , i =
## (t pi + f pi )
## (t pi + f ni )
##
## ()
##
## f  for each category i and the macro-averaged f
## are computed as:
##
## k=
##
## the test accuracy is calculated as
## f i =
##
## i(n (di , yi ) - maxj=yi n (di , j) > ) ()
## n i=
##
## i i
## , m acrof  =
## i + i
##
## q
```

```
## i=
##
## q
##
## f i
##
## ()
##
## n
##
## acc =
##
## where n is the number of objects in dt and yi indicates
## the true class of di .
## ..
##
## f metric
##
## to evaluate the performance of classification methods
## in dealing with an unbalanced class distribution, we use
## the f metric introduced by yang and liu []. this
## measure is equal to the harmonic mean of recall ()
## and precision (). the overall f score of the entire
## classification problem can be computed by a microaverage and a macro-average.
## micro-averaged f is computed globally over all
## classes, and emphasizes the performance of a classifier
## on common classes. define  and  as follows:
## q
##
## q
## t pi
## i= t pi
##  = q i=
## ,  = q
## ()
## i= (t pi + f pi )
```

```
## i= (t pi + f ni )
## where q is the number of classes. t pi (true positives)
## is the number of objects correctly predicted as class i,
## f pi (false positives) is the number of objects that are
## predicted to belong to class i but do not. the microaveraged f is computed as:
## m icrof  =
##
##
## +
##
## ()
##
## macro-averaged f is first computed locally over
## each class, and then the average over all classes is taken.
##
## the larger the microf and macrof values are, the
## higher the classification performance of the classifier.
## .
##
## experiments
##
## in this section, we present two experiments that
## demonstrate the effectiveness of the new random
## forest algorithm for classifying high dimensional data.
## high dimensional datasets with various sizes and
## characteristics were used in the experiments. the
## first experiment is designed to show how our proposed
## method can reduce the generalization error bound
## c s , and improve test accuracy when the size of
## the selected subspace is not too large. the second
## experiment is used to demonstrate the classification
## performance of our proposed method in comparison to
## other classification methods, i.e. svm, nb and knn.
## ..
##
## datasets
```

```
##
## in the experiments, we used eight real-world high
## dimensional datasets. these datasets were selected
## due to their diversities in the number of features, the
## number of instances, and the number of classes. their
## dimensionalities vary from  to ,. instances
## vary from  to  and the minority class rate varies
## from .% to .%. in each dataset, we randomly
## select % of instances as the training dataset, and
## the remaining data as the test dataset. detailed
## information of the eight datasets is listed in table .
## the fbis, re, re, tr, wap, tr, las
## and las datasets are classical text classification
## benchmark datasets which were carefully selected and
##
## the computer journal, vol. ??,
##
## no. ??,
##
## ????
##
## hybrid weighted random forests for classifying very high-dimensional data
## preprocessed by han and karypis []. dataset fbis
## was compiled from the foreign broadcast information
## service trec- []. the datasets re and re were
## selected from the reuters- text categorization test
## collection distribution . []. the datasets tr and
## tr were derived from trec- [], trec- [],
## and trec- []. dataset wap is from the webace
## project (wap) []. the datasets las and las were
## selected from the los angeles times for trec- [].
## the classes of these datasets were generated from the
## relevance judgment provided in these collections.
## ..
##
## performance comparisons between random forest methods
```

```
##
## the purpose of this experiment was to evaluate
## the effect of the hybrid weighted random forest
## method (h w rf) on strength, correlation, c s ,
## and test accuracy.
## the eight high dimensional
## datasets were analyzed and results were compared
## with seven other random forest methods, i.e., c.
## random forest (c. rf), cart random forest
## (cart rf), chaid random forest (chaid rf),
## hybrid random forest (h rf), c. weighted random
## forest (c. w rf), cart weighted random forest
## (cart w rf), chaid weighted random forest
## (chaid w rf). for each dataset, we ran each
## random forest algorithm against different sizes of the
## feature subspaces. since the number of features in these
## datasets was very large, we started with a subspace
## of  features and increased the subspace by  more
## features each time. for a given subspace size, we built
##  trees for each random forest model. in order to
## obtain a stable result, we built  random forest models
## for each subspace size, each dataset and each algorithm,
## and computed the average values of the four measures
## of strength, correlation, c s , and test accuracy as the
## final results for comparison. the performance of the
## eight random forest algorithms on the four measures
## for each of the  datasets is shown in figs. , , , and
## .
## fig.  plots the strength for the eight methods against
## different subspace sizes on each of the  datasets.
## in the same subspace, the higher the strength, the
## better the result. from the curves, we can see that
## the new algorithm (h w rf) consistently performs
## better than the seven other random forest algorithms.
## the advantages are more obvious for small subspaces.
## the new algorithm quickly achieved higher strength
```

```
## as the subspace size increases.
## the seven other
## random forest algorithms require larger subspaces to
## achieve a higher strength. these results indicate that
## the hybrid weighted random forest algorithm enables
## random forest models to achieve a higher strength
## for small subspace sizes compared to the seven other
## random forest algorithms.
## fig.  plots the curves for the correlations for the
## eight random forest methods on the  datasets. for
##
##
##
## small subspace sizes, h rf, c. rf, cart rf,
## and chaid rf produce higher correlations between
## the trees on all datasets. the correlation decreases
## as the subspace size increases. for the random forest
## models the lower the correlation between the trees
## then the better the final model.
## with our new
## random forest algorithm (h w rf) a low correlation
## level was achieved with very small subspaces in all
##  datasets. we also note that as the subspace size
## increased the correlation level increased as well. this is
## understandable because as the subspace size increases,
## the same informative features are more likely to be
## selected repeatedly in the subspaces, increasing the
## similarity of the decision trees. therefore, the feature
## weighting method for subspace selection works well for
## small subspaces, at least from the point of view of the
## correlation measure.
## fig.  shows the error bound indicator c s for the
## eight methods on the  datasets. from these figures
## we can observe that as the subspace size increases, c s
## consistently reduces. the behaviour indicates that a
## subspace size larger than log (m )+ benefits all eight
```

```
## algorithms. however, the new algorithm (h w rf)
## achieved a lower level of c s at subspace size of
## log (m ) +  than the seven other algorithms.
## fig.  plots the curves showing the accuracy of the
## eight random forest models on the test datasets from
## the  datasets. we can clearly see that the new random
## forest algorithm (h w rf) outperforms the seven
## other random forest algorithms in all eight data sets.
## it can be seen that the new method is more stable
## in classification performance than other methods. in
## all of these figures, it is observed that the highest test
## accuracy is often obtained with the default subspace size
## of log (m ) + . this implies that in practice, large
## size subspaces are not necessary to grow high-quality
## trees for random forests.
## ..
##
## performance comparisons
## classification methods
##
## with
##
## other
##
## we conducted a further experimental comparison
## against three other widely used text classification
## methods: support vector machines (svm), naive
## bayes (nb), and k-nearest neighbor (knn). the
## support vector machine used a linear kernel with a
## regularization parameter of ., which was often
## used in text categorization. for naive bayes, we
## adopted the multi-variate bernoulli event model that
## is frequently used in text classification []. for knearest neighbor (knn), we se
## neighbors to . in the experiments, we used weka's
## implementation for these three text classification
## methods []. we used a single subspace size of
```

```
## features in all eight datasets to run the random forest
## algorithms. for h rf, c. rf, cart rf, and
## chaid rf, we used a subspace size of  features in
## the first  datasets (i.e., fbis, re, re, tr, wap, and
##
## the computer journal, vol. ??,
##
## no. ??,
##
## ????
##
##
##
## baoxun xu, joshua zhexue huang, graham williams, yunming ye
## fbis
##
## re
##
## .
##
## .
##
## .
##
## .
##
## .
##
## strength
##
## strength
##
## .
##
## .
##
```

```
## .
##
## h_w_rf
##
## .
##
## c._w_rf
## cart_w_rf
##
## .
##
## h_w_rf
## c._w_rf
##
## .
##
## cart_w_rf
##
## chaid_w_rf
##
## chaid_w_rf
##
## .
##
## h_rf
##
## .
##
## h_rf
##
## c._rf
##
## c._rf
##
## .
##
```

```
## cart_rf
##
## .
##
## cart_rf
##
## chaid_rf
##
## chaid_rf
##
## .
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
```

```
##
##
##
##
##
##
##
##
##
##
##
## number of features
##
## number of features
##
## re
##
## .
##
##
##
## tr
##
## .
##
## .
##
## .
##
## .
## .
##
## .
##
```

```
## h_w_rf
## c._w_rf
##
## .
##
## cart_w_rf
##
## strength
##
## strength
##
## .
## .
## h_w_rf
## c._w_rf
##
## .
##
## cart_w_rf
##
## chaid_w_rf
##
## .
##
## chaid_w_rf
##
## h_rf
##
## h_rf
##
## .
##
## c._rf
##
## .
##
```

```
## c._rf
##
## cart_rf
##
## cart_rf
##
## .
##
## chaid_rf
##
## chaid_rf
##
## .
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
```

```
##
##
##
##
##
##
##
##
##
##
##
##
##
##
## number of features
##
## number of features
##
## wap
##
## tr
##
## .
##
## .
## .
##
## .
##
## .
##
## strength
##
```

```
## h_w_rf
##
## .
##
## c._w_rf
## cart_w_rf
##
## .
##
## strength
##
## .
##
## .
##
## .
## h_w_rf
## c._w_rf
##
## .
##
## cart_w_rf
## chaid_w_rf
##
## chaid_w_rf
## h_rf
##
## .
##
## h_rf
##
## .
##
## c._rf
##
## c._rf
```

```
##
## cart_rf
##
## cart_rf
##
## .
##
## .
##
## chaid_rf
##
## chaid_rf
##
## .
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
```

```
##
##
##
##
## las
##
##
##
##
##
##
##
##
##
##
## las
##
## .
##
## .
##
## .
##
## .
##
## .
##
## .
##
## .
## h_w_rf
##
## .
##
## c._w_rf
```

```
## cart_w_rf
## chaid_w_rf
##
## .
##
## strength
##
## strength
##
##
##
## number of features
##
## number of features
##
## .
##
##
##
## h_w_rf
##
## .
##
## c._w_rf
## cart_w_rf
##
## .
##
## chaid_w_rf
##
## h_rf
## c._rf
##
## .
##
## h_rf
```

```
##
## .
##
## c._rf
##
## cart_rf
## chaid_rf
##
## cart_rf
##
## .
##
## chaid_rf
##
## .
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
```

```
##
##
##
##
##
##
##
##
## number of features
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
## number of features
##
## figure . strength changes against the number of features in the subspace on the
```

```
##
## tr) to run the random forest algorithms, and used
## a subspace size of  features in the last  datasets
## (las and las) to run these random forest algorithms.
## for h w rf, c. w rf, cart w rf, and
## chaid w rf, we used breiman's subspace size of
##
## log (m ) +  to run these random forest algorithms.
## this number of features provided a consistent result as
## shown in fig. . in order to obtain stable results, we
## built  random forest models for each random forest
## algorithm and each dataset and present the average
##
## the computer journal, vol. ??,
##
## no. ??,
##
## ????
##
## hybrid weighted random forests for classifying very high-dimensional data
## fbis
##
##
##
## re
##
## .
##
## .
##
## .
##
## .
##
## correlation
##
```

```
## correlation
##
## .
##
## .
##
## .
##
## .
## h_w_rf
## c._w_rf
##
## .
##
## cart_w_rf
## chaid_w_rf
##
## .
##
## h_w_rf
##
## .
##
## c._w_rf
## cart_w_rf
##
## .
##
## chaid_w_rf
##
## h_rf
## c._rf
##
## .
##
## h_rf
```

```
##
## .
##
## c._rf
##
## cart_rf
## chaid_rf
##
## cart_rf
##
## .
##
## chaid_rf
##
## .
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
```

```
##
##
##
##
##
## number of features
##
##
##
##
##
##
##
##
##
##
##
## number of features
##
## re
##
## .
##
##
##
## tr
## .
##
## .
##
## .
##
## .
##
```

```
## .
## h_w_rf
## c._w_rf
##
## .
##
## cart_w_rf
## chaid_w_rf
##
## .
##
## correlation
##
## correlation
##
## .
##
## .
##
## h_w_rf
##
## .
##
## c._w_rf
## cart_w_rf
##
## .
##
## chaid_w_rf
## h_rf
##
## h_rf
## c._rf
##
## .
##
```

```
## c._rf
##
## .
##
## cart_rf
##
## cart_rf
##
## .
##
## chaid_rf
##
## chaid_rf
##
## .
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
```

```
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
## number of features
##
## number of features
##
## wap
##
## tr
##
## .
##
## .
##
## .
## .
##
## correlation
##
```

```
## .
##
## .
##
## h_w_rf
## c._w_rf
##
## .
##
## cart_w_rf
## chaid_w_rf
##
## .
##
## correlation
##
## .
##
## .
##
## h_w_rf
## c._w_rf
##
## .
##
## cart_w_rf
## chaid_w_rf
##
## .
##
## h_rf
##
## h_rf
##
## c._rf
##
```

```
## .
##
## c._rf
##
## cart_rf
##
## cart_rf
##
## .
##
## chaid_rf
##
## .
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
## chaid_rf
##
```

```
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
## number of features
##
## number of features
##
## las
##
## las
##
## .
##
## .
##
## .
##
## .
##
```

```
## .
##
## .
##
## h_w_rf
##
## .
##
## c._w_rf
## cart_w_rf
##
## .
##
## chaid_w_rf
##
## .
##
## correlation
##
## correlation
##
## .
##
## .
##
## .
##
## h_w_rf
## c._w_rf
##
## .
##
## cart_w_rf
## chaid_w_rf
##
## .
```

```
##
## h_rf
##
## h_rf
## c._rf
##
## .
##
## c._rf
##
## .
##
## cart_rf
##
## cart_rf
## chaid_rf
##
## chaid_rf
##
## .
##
## .
##
##
##
##
##
##
##
##
##
##
##
##
##
```

```
##
##
##
##
##
##
##
##
##
##
##
##
##
## number of features
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
```

```
##
##
##
##
## number of features
##
## figure . correlation changes against the number of features in the subspace on tl
##
## results, noting that the range of values are less than
## . and the hybrid trees are always more accurate.
## the comparison results of classification performance
## of eleven methods are shown in table .
## the
## performance is estimated using test accuracy (acc),
##
## micro f (mic), and macro f (mac). boldface
## denotes best results between eleven classification
## methods.
## while the improvement is often quite
## small, there is always an improvement demonstrated.
## we observe that our proposed method (h w rf)
##
## the computer journal, vol. ??,
##
## no. ??,
##
## ????
##
##
##
## baoxun xu, joshua zhexue huang, graham williams, yunming ye
## fbis
##
## .
##
## re
```

```
## .
##
## .
##
## log (m)+
##
## .
##
##
##
## .
##
## .
##
## .
##
## c._w_rf
##
## .
##
## .
##
##
##
## h_w_rf
##
## c s
##
## c s
##
##
##
## .
##
## cart_w_rf
##
```

```
## h_w_rf
## c._w_rf
##
## .
##
## cart_w_rf
##
## chaid_w_rf
##
## .
##
## chaid_w_rf
##
## .
##
## h_rf
##
## h_rf
##
## log (m)+
##
##
## c._rf
##
## .
##
## c._rf
##
## .
##
## cart_rf
##
## cart_rf
##
## chaid_rf
##
```

```
## .
##
## chaid_rf
##
## .
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
## number of features
##
##
##
##
```

```
##
##
##
##
##
## .
##
## .
##
## log (m)+
##
## .
##
## .
##
##
##
## c s
##
## h_w_rf
##
##
##
## .
##
## .
##
##
##
##
##
## tr
##
## .
##
## .
```

```
##
## c s
##
##
##
## number of features
##
## re
##
## .
##
##
##
## c._w_rf
##
## .
##
## .
##
## h_w_rf
## c._w_rf
##
## .
##
## cart_w_rf
##
## cart_w_rf
##
## chaid_w_rf
##
## .
##
## chaid_w_rf
##
## .
##
```

```
## h_rf
##
## h_rf
##
## c._rf
##
## log (m)+
##
## .
##
##
##
## c._rf
##
## .
##
## cart_rf
##
## cart_rf
##
## chaid_rf
##
## .
##
## chaid_rf
##
## .
##
##
##
##
##
##
##
##
##
```

```
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
## number of features
##
## number of features
##
## wap
```

```
##
## tr
##
##
##
## .
##
##
##
## log (m)+
##
## log (m)+
##
##
## .
##
##
##
##
##
## c._w_rf
##
## c s
##
##
##
## c s
##
## h_w_rf
##
##
##
##
##
## .
##
```

```
##
##
## h_w_rf
##
## .
##
## c._w_rf
## cart_w_rf
##
## cart_w_rf
## .
##
## chaid_w_rf
##
##
##
## chaid_w_rf
## h_rf
##
## h_rf
## c._rf
##
##
##
## c._rf
##
## .
##
## cart_rf
##
## cart_rf
## chaid_rf
##
## chaid_rf
##
##
```

```
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
## .
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
```

```
##
##
##
##
##
##
##
## number of features
##
## number of features
##
## las
##
## las
## .
##
## .
##
## .
## .
## .
## .
## .
## .
##
## c._w_rf
## cart_w_rf
##
## .
##
##
##
## h_rf
##
## .
##
```

```
## c._rf
## cart_w_rf
##
## .
##
## chaid_rf
##
##
##
##
##
##
##
##
## h_w_rf
##
## .
##
## c._w_rf
## cart_w_rf
##
## .
##
## chaid_w_rf
##
## log (m)+
##
##
##
## c s
##
## c s
##
##
##
```

```
## .
## h_w_rf
##
## .
##
##
##
##
##
##
##
##
##
##
##
##
##
##
## chaid_w_rf
##
## log (m)+
##
##
## .
##
## h_rf
## c._rf
##
## .
##
## cart_rf
## chaid_rf
##
```

```
## .
##
##
##
##
##
##
## number of features
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
## number of features
##
## figure . c s changes against the number of features in the subspace on the  high
##
```

```
## outperformed the other classification methods in all
## datasets.
##
## .
##
## conclusions
##
## in this paper, we presented a hybrid weighted random
## forest algorithm by simultaneously using a feature
## weighting method and a hybrid forest method to classify
## the computer journal, vol. ??,
##
## no. ??,
##
## ????
##
## hybrid weighted random forests for classifying very high-dimensional data
## fbis
##
## .
##
##
##
## re
##
## .
##
## .
## .
## .
##
## .
##
## .
## h_w_rf
## c._w_rf
```

```
##
## .
##
## cart_w_rf
##
## accuracy
##
## accuracy
##
## .
##
## chaid_w_rf
## h_rf
##
## .
##
## .
## h_w_rf
##
## .
##
## c._w_rf
## cart_w_rf
##
## .
##
## chaid_w_rf
## h_rf
##
## .
##
## c._rf
##
## c._rf
##
## log (m)+
```

```
##
## cart_rf
##
##
##
## .
##
## cart_rf
##
## log (m)+
##
## .
##
##
##
## chaid_rf
##
## chaid_rf
##
## .
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
```

```
##
##
##
##
##
##
##
##
##
## number of features
##
##
##
##
##
##
##
##
##
##
##
##
## number of features
##
## re
##
## tr
##
## .
##
## .
## .
```

```
##
## .
##
## log (m)+
##
## .
##
## .
##
##
##
## .
## h_w_rf
## c._w_rf
##
## .
##
## cart_w_rf
##
## .
##
## accuracy
##
## accuracy
##
## .
##
## chaid_w_rf
##
## .
##
## h_w_rf
##
## .
##
## c._w_rf
```

```
## cart_w_rf
##
## .
##
## chaid_w_rf
## h_rf
##
## h_rf
##
## .
##
## log (m)+
##
## .
##
## c._rf
##
## c._rf
##
##
##
## cart_rf
##
## cart_rf
##
## .
##
## .
##
## chaid_rf
##
## chaid_rf
##
## .
##
##
```

```
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
```

```
## number of features
##
## number of features
##
## wap
##
## .
##
##
##
## tr
##
## .
##
## .
##
## .
##
## .
##
## .
##
## .
##
## h_w_rf
## c._w_rf
##
## .
##
## cart_w_rf
##
## accuracy
##
## accuracy
##
## .
```

```
##
## .
##
## .
##
## h_w_rf
## c._w_rf
##
## .
##
## cart_w_rf
## chaid_w_rf
##
## chaid_w_rf
##
## .
##
## h_rf
##
## log (m)+
##
##
## .
##
## h_rf
##
## c._rf
##
## .
##
## cart_rf
##
## .
##
## c._rf
##
```

```
## log (m)+
##
##
## .
##
## cart_rf
## chaid_rf
##
## chaid_rf
##
## .
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
```

```
##
##
##
##
##
##
##
##
##
##
##
## number of features
##
## number of features
##
## las
##
## .
##
##
##
## las
##
## .
##
## .
## .
## .
##
## accuracy
##
## .
## h_w_rf
## c._w_rf
```

```
##
## .
##
## cart_w_rf
## chaid_w_rf
##
## .
##
## accuracy
##
## .
##
## .
##
## .
##
## h_w_rf
## c._w_rf
## cart_w_rf
##
## .
##
## chaid_w_rf
## h_rf
##
## h_rf
##
## log (m)+
##
## .
##
## c._rf
##
##
##
## c._rf
```

```
##
## log (m)+
##
## .
##
##
##
## cart_rf
##
## cart_rf
## chaid_rf
##
## chaid_rf
##
## .
##
## .
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
```

```
##
##
##
##
##
##
##
##
##
## number of features
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
## number of features
```

```
##
## figure . test accuracy changes against the number of features in the subspace on
##
## high dimensional data. our algorithm not only retains
## a small subspace size (breiman's formula log (m ) +
## for determining the subspace size) to create accurate
## random forest models, but also effectively reduces
## the upper bound of the generalization error and
##
## improves classification performance. from the results of
## experiments on various high dimensional datasets, the
## random forest generated by our new method is superior
## to other classification methods. we can use the default
## log (m ) +  subspace size and generally guarantee
##
## the computer journal, vol. ??,
##
## no. ??,
##
## ????
##
##
##
## baoxun xu, joshua zhexue huang, graham williams, yunming ye
##
## table . the comparison of results
## datasets
## dataset
## fbis
## measures
## acc
## mic
## svm
## . .
## knn
## .
```

```
## .
## nb
## . .
## h rf
## . .
## c. rf
## . .
## cart rf
## . .
## chaid rf
## . .
## h w rf
## . .
## c. w rf
## . .
## cart w rf
## . .
## chaid w rf
## . .
## dataset
## wap
## measures
## acc
## mic
## svm
## .
## .
## knn
## . .
## nb
## . .
## h rf
## . .
## c. rf
## . .
## cart rf
```

```
## . .
## chaid rf
## . .
## h w rf
## . .
## c. w rf
## . .
## cart w rf
## .
## .
## chaid w rf
## . .
##
## (best accuracy, micro f, and macro f results) of the eleven methods on the
## re
## mic
## .
## .
## .
## .
## .
## .
## .
## .
## .
## .
## .
## tr
## mac
## acc
## mic
## . . .
## . . .
## . . .
## . . .
## . . .
```

```
## .
## . .
## . .
## .
## . . .
## . . .
## .
## .
## .
## .
## .
## .
##
## mac
## .
## .
## .
## .
## .
## .
## .
## .
## .
## .
## .
##
## acc
## .
## .
## .
## .
## .
## .
## .
## .
## .
```

```
## .
## .
##
## to always produce the best models, on a variety of
## measures, by using the hybrid weighted random forest
## algorithm.
## acknowledgements
## this research is supported in part by nsfc under
## grant no., and shenzhen new industry development fund under grant no.cxba
## references
## [] breiman, l. () random forests. machine learning,
## , -.
## [] ho, t. () random subspace method for constructing decision forests. ieee trans
## analysis and machine intelligence, , -.
## [] quinlan, j. () c.: programs for machine
## learning. morgan kaufmann.
## [] breiman, l. () classification and regression trees.
## chapman & hall crc.
## [] breiman, l. () bagging predictors.
## machine
## learning, , -.
## [] ho, t. () random decision forests. proceedings
## of the third international conference on document
## analysis and recognition, pp. -. ieee.
## [] dietterich, t. () an experimental comparison of
## three methods for constructing ensembles of decision
## trees: bagging, boosting, and randomization. machine
## learning, , -.
##
## mac
## .
## .
## .
## .
## .
## .
```

```
## .
## .
## .
## .
## .
## mac
## .
## .
## .
## .
## .
## .
## .
## .
## .
## .
##
## re
## mic
## .
## .
## .
## .
## .
## .
## .
## .
## .
## .
## las
## acc
## mic
## .
## .
```

```
## . .
## . .
## .
## .
## . .
## . .
## .
## .
## . .
## . .
## . .
## . .
## acc
## .
## .
## .
## .
## .
## .
## .
## .
## .
## .
## .
##
## tr
## mic
## .
## .
## .
## .
## .
## .
## .
## .
## .
```

```
## .
## .
## las
## mac
## acc
## mic
## . . .
## . . .
## .
## .
## .
## .
## . .
## .
## . .
## .
## .
## .
## . . .
## . . .
## . . .
## . .
## .
## .
## . .
## mac
## .
## .
## .
## .
## .
## .
## .
## .
## .
## .
```

```
## .
##
## acc
## .
## .
## .
## .
## .
## .
## .
## .
## .
## .
## .
##
## mac
## .
## .
## .
## .
## .
## .
## .
## .
## .
## .
## .
## .
## mac
## .
## .
## .
## .
## .
## .
## .
## .
## .
```

.
.
.
##
[] banfield, r., hall, l., bowyer, k., and kegelmeyer, w.
() a comparison of decision tree ensemble creation
techniques. ieee transactions on pattern analysis
and machine intelligence, , -.
##
[] robnik-sikonja,
m. () improving random forests.
proceedings of the th european conference on
machine learning, pp. -. springer.
[] ho, t. () c. decision forests. proceedings of
the fourteenth international conference on pattern
recognition, pp. -. ieee.
[] dietterrich, t. () machine learning research: four
current direction. artificial intelligence magzine, ,
-.
[] amaratunga, d., cabrera, j., and lee, y. ()
enriched random forests. bioinformatics, , -
.
[] ye, y., li, h., deng, x., and huang, j. ()
feature weighting random forest for detection of hidden
web search interfaces. the journal of computational
linguistics and chinese language processing, , -
.
[] xu, b., huang, j., williams, g., wang, q., and
ye, y. () classifying very high-dimensional data
with random forests built from small subspaces.
international journal of data warehousing and
mining, , -.
[] xu, b., huang, j., williams, g., li, j., and ye, y.
() hybrid random forests: advantages of mixed
trees in classifying text data. proceedings of the th
pacific-asia conference on knowledge discovery and

```
## data mining. springer.
##
## the computer journal, vol. ??,
##
## no. ??,
##
## ????
##
## hybrid weighted random forests for classifying very high-dimensional data
## [] biggs, d., de ville, b., and suen, e. () a method
## of choosing multiway partitions for classification and
## decision trees. journal of applied statistics, , -.
## [] ture, m., kurt, i., turhan kurum, a., and ozdamar,
## k. () comparing classification techniques for
## predicting essential hypertension. expert systems with
## applications, , -.
## [] begum, n., m.a., f., and ren, f. () automatic text summarization using suppor
## international journal of innovative computing, information and control, , -.
## [] chen, j., huang, h., tian, s., and qu, y. ()
## feature selection for text classification with naive
## bayes. expert systems with applications, , -
## .
## [] tan, s. () neighbor-weighted k-nearest neighbor
## for unbalanced text corpus.
## expert systems with
## applications, , -.
## [] pearson, k. () on the theory of contingency and
## its relation to association and normal correlation.
## cambridge university press.
## [] yang, y. and liu, x. () a re-examination of
## text categorization methods. proceedings of the th
## international conference on research and development
## in information retrieval, pp. -. acm.
## [] han, e. and karypis, g. () centroid-based
## document classification: analysis and experimental
## results. proceedings of the th european conference on
```

```
## principles of data mining and knowledge discovery,
## pp. -. springer.
## [] trec.
## ()
## text
## retrieval
## conference,
## http:  trec.nist.gov.
## [] lewis,
## d.
## ()
## reuters-
## text
## categorization
## test
## collection
## distribution
## .,
## http:  www.research.att.com  lewis.
## [] han, e., boley, d., gini, m., gross, r., hastings,
## k., karypis, g., kumar, v., mobasher, b., and
## moore, j. () webace: a web agent for document
## categorization and exploration. proceedings of the nd
## international conference on autonomous agents, pp.
## -. acm.
## [] mccallum, a. and nigam, k. () a comparison of
## event models for naive bayes text classification. aaai workshop on learning for t
## .
## [] witten, i., frank, e., and hall, m. () data mining:
## practical machine learning tools and techniques.
## morgan kaufmann.
##
## the computer journal, vol. ??,
##
## no. ??,
```

```
##
## ????
```

Numbers may or may not be relevant to our analyses. This transform can remove numbers simply.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

21.12 Remove Punctuation

```
docs <- tm_map(docs, removePunctuation)
```

```
viewDocs(docs, 16)
```

```
## hybrid weighted random forests for
## classifying very highdimensional data
## baoxun xu  joshua zhexue huang  graham williams and
## yunming ye
##
##
## department of computer science harbin institute of technology shenzhen graduate
## school shenzhen  china
##
## shenzhen institutes of advanced technology chinese academy of sciences shenzhen
##  china
## email amusing gmailcom
## random forests are a popular classification method based on an ensemble of a
## single type of decision trees from subspaces of data in the literature there
## are many different types of decision tree algorithms including c cart and
## chaid each type of decision tree algorithm may capture different information
## and structure this paper proposes a hybrid weighted random forest algorithm
## simultaneously using a feature weighting method and a hybrid forest method to
## classify very high dimensional data the hybrid weighted random forest algorithm
## can effectively reduce subspace size and improve classification performance
## without increasing the error bound we conduct a series of experiments on eight
## high dimensional datasets to compare our method with traditional random forest
## methods and other classification methods the results show that our method
## consistently outperforms these traditional methods
## keywords random forests hybrid weighted random forest classification decision tre
##
##
##
## introduction
##
## random forests   are a popular classification
## method which builds an ensemble of a single type
## of decision trees from different random subspaces of
## data the decision trees are often either built using
## c  or cart  but only one type within
```

```
## a single random forest in recent years random
## forests have attracted increasing attention due to
##  its competitive performance compared with other
## classification methods especially for highdimensional
## data  algorithmic intuitiveness and simplicity and
##  its most important capability  ensemble using
## bagging  and stochastic discrimination
## several methods have been proposed to grow random
## forests from subspaces of data        in
## these methods the most popular forest construction
## procedure was proposed by breiman  to first use
## bagging to generate training data subsets for building
## individual trees
## a subspace of features is then
## randomly selected at each node to grow branches of
## a decision tree the trees are then combined as an
## ensemble into a forest as an ensemble learner the
## performance of a random forest is highly dependent
## on two factors the performance of each tree and the
## diversity of the trees in the forests  breiman
## formulated the overall performance of a set of trees as
## the average strength and proved that the generalization
##
## error of a random forest is bounded by the ratio of the
## average correlation between trees divided by the square
## of the average strength of the trees
## for very high dimensional data such as text data
## there are usually a large portion of features that are
## uninformative to the classes during this forest building
## process informative features would have the large
## chance to be missed if we randomly select a small
## subspace breiman suggested selecting log m
## features in a subspace where m is the number of
## independent features in the data from high dimensional
## data  as a result weak trees are created from these
## subspaces the average strength of those trees is reduced
```

```
## and the error bound of the random forest is enlarged
## therefore when a large proportion of such weak
## trees are generated in a random forest the forest has a
## large likelihood to make a wrong decision which mainly
## results from those weak trees classification power
## to address this problem we aim to optimize decision
## trees of a random forest by two strategies one
## straightforward strategy is to enhance the classification
## performance of individual trees by a feature weighting
## method for subspace sampling    in this
## method feature weights are computed with respect
## to the correlations of features to the class feature
## and regarded as the probabilities of the feature to
## be selected in subspaces this method obviously
## increases the classification performance of individual
##
## the computer journal vol
##
## no
##
##
##
##
##
## baoxun xu joshua zhexue huang graham williams yunming ye
##
## trees because the subspaces will be biased to contain
## more informative features however the chance of more
## correlated trees is also increased since the features with
## large weights are likely to be repeatedly selected
## the second strategy is more straightforward use
## several different types of decision trees for each training
## data subset to increase the diversity of the trees
## and then select the optimal tree as the individual
## tree classifier in the random forest model the work
## presented here extends the algorithm developed in
```

```
## specifically we build three different types of tree
## classifiers c cart and chaid   for each
## training data subset we then evaluate the performance
## of the three classifiers and select the best tree in
## this way we build a hybrid random forest which may
## include different types of decision trees in the ensemble
## the added diversity of the decision trees can effectively
## improve the accuracy of each tree in the forest and
## hence the classification performance of the ensemble
## however when we use this method to build the best
## random forest model for classifying high dimensional
## data we can not be sure of what subspace size is best
## in this paper we propose a hybrid weighted random
## forest algorithm by simultaneously using a new feature
## weighting method together with the hybrid random
## forest method to classify high dimensional data in
## this new random forest algorithm we calculate feature
## weights and use weighted sampling to randomly select
## features for subspaces at each node in building different
## types of trees classifiers c cart and chaid for
## each training data subset and select the best tree as
## the individual tree in the final ensemble model
## experiments were performed on  high dimensional
## text datasets with dimensions ranging from  to
##  we compared the performance of eight random
## forest methods and wellknown classification methods
## c random forest cart random forest chaid
## random forest hybrid random forest c weighted
## random forest cart weighted random forest chaid
## weighted random forest hybrid weighted random
## forest support vector machines  naive bayes
## and knearest neighbors
## the experimental
## results show that our hybrid weighted random forest
## achieves improved classification performance over the
## ten competitive methods
```

```
## the remainder of this paper is organized as follows
## in section  we introduce a framework for building
## a hybrid weighted random forest and describe a new
## random forest algorithm section  summarizes four
## measures to evaluate random forest models we present
## experimental results on  high dimensional text datasets
## in section  section  contains our conclusions
##
## table  contingency table of input feature a and class
## feature y
## y  y
## y  yj
## y  yq total
## a  a
##
##
## j
##
## q
##
##
##
##
##
##
##
##
##
##
##
##
## a  ai
## i
##
```

```
## ij
##
## iq
## i
##
##
##
##
##
##
##
##
##
##
## a  ap
## p
##
## pj
##
## pq
## p
## total
##
##
## j
##
## q
##
##
## general framework for building hybrid random forests
## by integrating these two methods we propose a novel
## hybrid weighted random forest algorithm
##
```

```
##
## let y be the class or target feature with q distinct
## class labels yj for j       q for the purposes of
## our discussion we consider a single categorical feature
## a in dataset d with p distinct category values we
## denote the distinct values by ai for i       p
## numeric features can be discretized into p intervals with
## a supervised discretization method
## assume d has val objects the size of the subset of
## d satisfying the condition that a  ai and y  yj is
## denoted by ij  considering all combinations of the
## categorical values of a and the labels of y  we can
## obtain a contingency table  of a against y as shown
## in table  the far right column contains the marginal
## totals for feature a
##
## hybrid
## forests
##
## weighted
##
## random
##
## in this section we first introduce a feature weighting
## method for subspace sampling then we present a
##
## q
##
##
## i
##
## ij
##
## for i       p
##
##
```

```
##
## j
##
## and the bottom row is the marginal totals for class
## feature y
## j
##
## p
##
##
## ij
##
## for j       q
##
##
##
## i
##
## the grand total the total number of samples is in
## the bottom right corner
##
##
## q
## p
##
##
## ij
##
##
##
## j i
##
## given a training dataset d and feature a we first
## compute the contingency table the feature weights are
## then computed using the two methods to be discussed
## in the following subsection
```

```
##
##
##
##
## notation
##
## feature weighting method
##
## in this subsection we give the details of the feature
## weighting method for subspace sampling in random
## forests consider an mdimensional feature space
## a  a      am  we present how to compute the
##
## the computer journal vol
##
## no
##
##
##
## hybrid weighted random forests for classifying very highdimensional data
## weights w  w      wm  for every feature in the space
## these weights are then used in the improved algorithm
## to grow each decision tree in the random forest
##  feature weight computation
## the weight of feature a represents the correlation
## between the values of feature a and the values of the
## class feature y  a larger weight will indicate that the
## class labels of objects in the training dataset are more
## correlated with the values of feature a indicating that
## a is more informative to the class of objects thus it
## is suggested that a has a stronger power in predicting
## the classes of new objects
## in the following we propose to use the chisquare
## statistic to compute feature weights because this
## method can quantify the correspondence between two
## categorical variables
```

```
## given the contingency table of an input feature a and
## the class feature y of dataset d the chisquare statistic
## of the two features is computed as
## corra y
##
## q
## p
##
## ij  tij
## tij
## i j
##
##
##
## where ij is the observed frequency from the
## contingency table and tij is the expected frequency
## computed as
## i x j
## tij
##
##
##
##
## the larger the measure corra y  the more
## informative the feature a is in predicting class y
##  normalized feature weight
## in practice feature weights are normalized for feature
## subspace sampling we use corra y  to measure the
## informativeness of these features and consider them
## as feature weights however to treat the weights as
## probabilities of features we normalize the measures to
## ensure the sum of the normalized feature weights is
## equal to  let corrai  y    i  m  be the set
## of m feature measures we compute the normalized
## weights as
##
```

```
## corrai  y
## wi  n
##
## i corrai  y
## here we use the square root to smooth the values of
## the measures wi can be considered as the probability
## that feature ai is randomly sampled in a subspace the
## more informative a feature is the larger the weight and
## the higher the probability of the feature being selected
##
## diversity is commonly obtained by using bagging and
## random subspace sampling we introduce a further
## element of diversity by using different types of trees
## considering an analogy with forestry the different data subsets from bagging rep
## one is to use three types of decision tree algorithms to
## generate three different tree classifiers for each training data subset the othe
## of each tree as the measure of tree importance in this
## paper we use the outofbag accuracy to assess the importance of a tree
## following breiman  we use bagging to generate
## a series of training data subsets from which we build
## trees for each tree the data subset used to grow
## the tree is called the inofbag iob data and the
## remaining data subset is called the outofbag oob
## data since oob data is not used for building trees
## we can use this data to objectively evaluate each trees
## accuracy and importance the oob accuracy gives an
## unbiased estimate of the true accuracy of a model
## given n instances in a training dataset d and a tree
## classifier hk iobk  built from the kth training data
## subset iobk  we define the oob accuracy of the tree
## hk iobk  for di  d as
## n
## oobacck
##
## framework for building a hybrid random
## forest
```

```
##
## as an ensemble learner the performance of a random
## forest is highly dependent on two factors the diversity
## among the trees and the accuracy of each tree
##
## i
##
## ihk di   yi  di  oobk
## n
## i idi  oobk
##
##
##
## where i is an indicator function the larger the
## oobacck  the better the classification quality of a tree
## we use the outofbag data subset oobi to calculate
## the outofbag accuracies of the three types of trees
## c cart and chaid with evaluation values e
## e and e respectively
## fig  illustrates the procedure for building a hybrid
## random forest model firstly a series of iob oob
## datasets are generated from the entire training dataset
## by bagging then three types of tree classifiers c
## cart and chaid are built using each iob dataset
## the corresponding oob dataset is used to calculate the
## oob accuracies of the three tree classifiers finally
## we select the tree with the highest oob accuracy as
## the final tree classifier which is included in the hybrid
## random forest
## building a hybrid random forest model in this
## way will increase the diversity among the trees
## the classification performance of each individual tree
## classifier is also maximized
##
##
##
```

```
##
##
##
## decision tree algorithms
##
## the core of our approach is the diversity of decision
## tree algorithms in our random forest different decision
## tree algorithms grow structurally different trees from
## the same training data selecting a good decision tree
## algorithm to grow trees for a random forest is critical
##
## the computer journal vol
##
## no
##
##
##
##
##
## baoxun xu joshua zhexue huang graham williams yunming ye
## the difference lies in the way to split a node such
## as the split functions and binary branches or multibranches in this work we use t
## tree algorithms to build a hybrid random forest
##
##
##
## figure  the hybrid random forests framework
##
## for the performance of the random forest few studies
## have considered how different decision tree algorithms
## affect a random forest we do so in this paper
## the common decision tree algorithms are as follows
## classification trees  c is a supervised
## learning classification algorithm used to construct
## decision trees given a set of preclassified objects each
## described by a vector of attribute values we construct
```

```
## a mapping from attribute values to classes c uses
## a divideandconquer approach to grow decision trees
## beginning with the entire dataset a tree is constructed
## by considering each predictor variable for dividing the
## dataset the best predictor is chosen at each node
## using a impurity or diversity measure the goal is
## to produce subsets of the data which are homogeneous
## with respect to the target variable c selects the test
## that maximizes the information gain ratio igr
## classification and regression tree cart is
## a recursive partitioning method that can be used for
## both regression and classification the main difference
## between c and cart is the test selection and
## evaluation process
## chisquared automatic interaction detector
## chaid method is based on the chisquare test of
## association a chaid decision tree is constructed
## by repeatedly splitting subsets of the space into two
## or more nodes to determine the best split at any
## node any allowable pair of categories of the predictor
## variables is merged until there is no statistically
## significant difference within the pair with respect to the
## target variable
## from these decision tree algorithms we can see that
##
## hybrid weighted random forest algorithm
##
## in this subsection we present a hybrid weighted
## random forest algorithm by simultaneously using the
## feature weights and a hybrid method to classify high
## dimensional data the benefits of our algorithm has
## two aspects firstly compared with hybrid forest
## method  we can use a small subspace size to
## create accurate random forest models
## secondly
## compared with building a random forest using feature
```

```
## weighting  we can use several different types of
## decision trees for each training data subset to increase
## the diversities of trees the added diversity of the
## decision trees can effectively improve the classification
## performance of the ensemble model the detailed steps
## are introduced in algorithm
## input parameters to algorithm  include a training
## dataset d the set of features a the class feature y
## the number of trees in the random forest k and the
## size of subspaces m the output is a random forest
## model m  lines  form the loop for building k
## decision trees in the loop line  samples the training
## data d by sampling with replacement to generate an
## inofbag data subset iobi for building a decision tree
## line  build three types of tree classifiers c
## cart and chaid in this procedure line  calls
## the function createt reej  to build a tree classifier
## line  calculates the outofbag accuracy of the tree
## classifier after this procedure line  selects the tree
## classifier with the maximum outofbag accuracy k
## decision tree trees are thus generated to form a hybrid
## weighted random forest model m
## generically function createt reej  first creates a
## new node then it tests the stopping criteria to decide
## whether to return to the upper node or to split this
## node if we choose to split this node then the feature
## weighting method is used to randomly select m features
## as the subspace for node splitting these features
## are used as candidates to generate the best split to
## partition the node for each subset of the partition
## createt reej  is called again to create a new node under
## the current node if a leaf node is created it returns to
## the parent node this recursive process continues until
## a full tree is generated
##
## the computer journal vol
```

```
##
## no
##
##
##
## hybrid weighted random forests for classifying very highdimensional data
## algorithm  new random forest algorithm
##  input
##   d  the training dataset
##   a  the features space a  a   am
##   y  the class features space y  y   yq
##   k  the number of trees
##   m  the size of subspaces
##  output a random forest m
##  method
##  for i   to k do
##
## draw a bootstrap sample inofbag data subset
## iobi and outofbag data subset oobi from
## training dataset d
##
## for j   to  do
##
## hij iobi   createt reej
## use outofbag data subset oobi to calculate
##
## the outofbag accuracy oobacci j of the tree
## classifier hij iobi  by equation
##
## end for
##
## select hi iobi  with the highest outofbag
## accuracy oobacci as optimal tree i
##  end for
##  combine
## the
```

```
## k
## tree
## classifiers
## h iob  h iob   hk iobk  into a random
## forest m
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
## function createtree
## create a new node n
## if stopping criteria is met then
## return n as a leaf node
## else
## for j   to m do
## compute
## the
## informativeness
## measure
## corraj  y  by equation
## end for
## compute feature weights w  w   wm  by
```

```
## equation
## use the feature weighting method to randomly
## select m features
## use these m features as candidates to generate
## the best split for the node to be partitioned
## call createtree for each split
## end if
## return n
## evaluation measures
##
## in this paper we use five measures ie strength
## correlation error bound c s  test accuracy and f
## metric to evaluate our random forest models strength
## measures the collective performance of individual trees
## in a random forest and the correlation measures the
## diversity of the trees the ratio of the correlation
## over the square of the strength c s indicates the
## generalization error bound of the random forest model
## these three measures were introduced in  the
## accuracy measures the performance of a random forest
## model on unseen test data the f metric is a
##
##
##
## commonly used measure of classification performance
##
##
## strength and correlation measures
##
## we follow breimans method described in  to
## calculate the strength correlation and the ratio c s
## following breimans notation we denote strength as
## s and correlation as  let hk iobk  be the kth
## tree classifier grown from the kth training data iobk
## sampled from d with replacement
## assume the
```

```
## random forest model contains k trees the outofbag
## proportion of votes for di  d on class j is
## k
## ihk di   j di
##   iobk
## qdi  j  kk
##
##   iobk
## k idi
## this is the number of trees in the random forest
## which are trained without di and classify di into class
## j divided by the number of training datasets not
## containing di
## the strength s is computed as
##
## qdi  yi   maxjyi qdi  j
## n i
## n
##
## s
##
##
##
## where n is the number of objects in d and yi indicates
## the true class of di
## the correlation  is computed as
## n
##
##
##
## i qdi  yi   maxjyi qdi  j  s
## n
##
##
##
##
```

```
## k
##
## k
## k  pk  pk
## k pk  p
## where
##
## n
## pk
##
## i
##
## ihk di   yi  di
##   iobk
## n
##   iobk
## i idi
##
##
##
## and
## n
## pk
##
## i
##
## ihk di   jdi  y  di
##   iobk
## n
## id
##
##
## iob
##
## i
## k
```

```
## i
##
##
##
## where
## jdi  y   argmaxjyi qd j
##
##
##
## is the class that obtains the maximal number of votes
## among all classes but the true class
##
##
## general error bound measure c s
##
## given the strength and correlation the outofbag
## estimate of the c s measure can be computed
## an important theoretical result in breimans method
## is the upper bound of the generalization error of the
## random forest ensemble that is derived as
## p e    s  s
##
##
##
## where  is the mean value of correlations between all
## pairs of individual classifiers and s is the strength of
## the set of individual classifiers that is estimated as the
##
## the computer journal vol
##
## no
##
##
##
##
##
```

```
## baoxun xu joshua zhexue huang graham williams yunming ye
##
## average accuracy of individual classifiers on d with
## outofbag evaluation this inequality shows that the
## generalization error of a random forest is affected by
## the strength of individual classifiers and their mutual
## correlations therefore breiman defined the c s ratio
## to measure a random forest as
## c s   s
##
##
##
## the smaller the ratio the better the performance of
## the random forest as such c s gives guidance for
## reducing the generalization error of random forests
##
##
## test accuracy
##
## the test accuracy measures the classification performance of a random forest on t
## dt be a test data and yt be the class labels given
## di  dt  the number of votes for di on class j is
## n di  j
##
## k
##
##
## ihk di   j
##
##
##
## table
## summary statistic of  highdimensional
## datasets
## name
## features
```

```
## instances
## classes  minority
## fbis
##
##
##
##
## re
##
##
##
##
## re
##
##
##
##
## tr
##
##
##
##
## wap
##
##
##
##
## tr
##
##
##
##
## las
##
##
##
```

```
##
## las
##
##
##
##
## it emphasizes the performance of a classifier on rare
## categories define  and  as follows
##
## i
##
## t pi
## t pi
##  i
## t pi  f pi
## t pi  f ni
##
##
##
## f  for each category i and the macroaveraged f
## are computed as
##
## k
##
## the test accuracy is calculated as
## f i
##
## in di  yi   maxjyi n di  j
## n i
##
## i i
##  m acrof
## i  i
##
## q
```

```
## i
##
## q
##
## f i
##
##
##
## n
##
## acc
##
## where n is the number of objects in dt and yi indicates
## the true class of di
##
##
## f metric
##
## to evaluate the performance of classification methods
## in dealing with an unbalanced class distribution we use
## the f metric introduced by yang and liu  this
## measure is equal to the harmonic mean of recall
## and precision  the overall f score of the entire
## classification problem can be computed by a microaverage and a macroaverage
## microaveraged f is computed globally over all
## classes and emphasizes the performance of a classifier
## on common classes define  and  as follows
## q
##
## q
## t pi
## i t pi
##   q i
##    q
##
## i t pi  f pi
```

```
## i t pi  f ni
## where q is the number of classes t pi true positives
## is the number of objects correctly predicted as class i
## f pi false positives is the number of objects that are
## predicted to belong to class i but do not the microaveraged f is computed as
## m icrof
##
##
##
##
##
##
## macroaveraged f is first computed locally over
## each class and then the average over all classes is taken
##
## the larger the microf and macrof values are the
## higher the classification performance of the classifier
##
##
## experiments
##
## in this section we present two experiments that
## demonstrate the effectiveness of the new random
## forest algorithm for classifying high dimensional data
## high dimensional datasets with various sizes and
## characteristics were used in the experiments the
## first experiment is designed to show how our proposed
## method can reduce the generalization error bound
## c s  and improve test accuracy when the size of
## the selected subspace is not too large the second
## experiment is used to demonstrate the classification
## performance of our proposed method in comparison to
## other classification methods ie svm nb and knn
##
##
## datasets
```

```
##
## in the experiments we used eight realworld high
## dimensional datasets these datasets were selected
## due to their diversities in the number of features the
## number of instances and the number of classes their
## dimensionalities vary from  to  instances
## vary from  to  and the minority class rate varies
## from  to  in each dataset we randomly
## select  of instances as the training dataset and
## the remaining data as the test dataset detailed
## information of the eight datasets is listed in table
## the fbis re re tr wap tr las
## and las datasets are classical text classification
## benchmark datasets which were carefully selected and
##
## the computer journal vol
##
## no
##
##
##
## hybrid weighted random forests for classifying very highdimensional data
## preprocessed by han and karypis  dataset fbis
## was compiled from the foreign broadcast information
## service trec  the datasets re and re were
## selected from the reuters text categorization test
## collection distribution   the datasets tr and
## tr were derived from trec  trec
## and trec  dataset wap is from the webace
## project wap  the datasets las and las were
## selected from the los angeles times for trec
## the classes of these datasets were generated from the
## relevance judgment provided in these collections
##
##
## performance comparisons between random forest methods
```

```
##
## the purpose of this experiment was to evaluate
## the effect of the hybrid weighted random forest
## method h w rf on strength correlation c s
## and test accuracy
## the eight high dimensional
## datasets were analyzed and results were compared
## with seven other random forest methods ie c
## random forest c rf cart random forest
## cart rf chaid random forest chaid rf
## hybrid random forest h rf c weighted random
## forest c w rf cart weighted random forest
## cart w rf chaid weighted random forest
## chaid w rf for each dataset we ran each
## random forest algorithm against different sizes of the
## feature subspaces since the number of features in these
## datasets was very large we started with a subspace
## of  features and increased the subspace by  more
## features each time for a given subspace size we built
##  trees for each random forest model in order to
## obtain a stable result we built  random forest models
## for each subspace size each dataset and each algorithm
## and computed the average values of the four measures
## of strength correlation c s  and test accuracy as the
## final results for comparison the performance of the
## eight random forest algorithms on the four measures
## for each of the  datasets is shown in figs    and
##
## fig  plots the strength for the eight methods against
## different subspace sizes on each of the  datasets
## in the same subspace the higher the strength the
## better the result from the curves we can see that
## the new algorithm h w rf consistently performs
## better than the seven other random forest algorithms
## the advantages are more obvious for small subspaces
## the new algorithm quickly achieved higher strength
```

```
## as the subspace size increases
## the seven other
## random forest algorithms require larger subspaces to
## achieve a higher strength these results indicate that
## the hybrid weighted random forest algorithm enables
## random forest models to achieve a higher strength
## for small subspace sizes compared to the seven other
## random forest algorithms
## fig  plots the curves for the correlations for the
## eight random forest methods on the  datasets for
##
##
##
## small subspace sizes h rf c rf cart rf
## and chaid rf produce higher correlations between
## the trees on all datasets the correlation decreases
## as the subspace size increases for the random forest
## models the lower the correlation between the trees
## then the better the final model
## with our new
## random forest algorithm h w rf a low correlation
## level was achieved with very small subspaces in all
##  datasets we also note that as the subspace size
## increased the correlation level increased as well this is
## understandable because as the subspace size increases
## the same informative features are more likely to be
## selected repeatedly in the subspaces increasing the
## similarity of the decision trees therefore the feature
## weighting method for subspace selection works well for
## small subspaces at least from the point of view of the
## correlation measure
## fig  shows the error bound indicator c s for the
## eight methods on the  datasets from these figures
## we can observe that as the subspace size increases c s
## consistently reduces the behaviour indicates that a
## subspace size larger than log m  benefits all eight
```

```
## algorithms however the new algorithm h w rf
## achieved a lower level of c s at subspace size of
## log m    than the seven other algorithms
## fig  plots the curves showing the accuracy of the
## eight random forest models on the test datasets from
## the  datasets we can clearly see that the new random
## forest algorithm h w rf outperforms the seven
## other random forest algorithms in all eight data sets
## it can be seen that the new method is more stable
## in classification performance than other methods in
## all of these figures it is observed that the highest test
## accuracy is often obtained with the default subspace size
## of log m    this implies that in practice large
## size subspaces are not necessary to grow highquality
## trees for random forests
##
##
## performance comparisons
## classification methods
##
## with
##
## other
##
## we conducted a further experimental comparison
## against three other widely used text classification
## methods support vector machines svm naive
## bayes nb and knearest neighbor knn the
## support vector machine used a linear kernel with a
## regularization parameter of  which was often
## used in text categorization for naive bayes we
## adopted the multivariate bernoulli event model that
## is frequently used in text classification  for knearest neighbor knn we set the n
## neighbors to  in the experiments we used wekas
## implementation for these three text classification
## methods  we used a single subspace size of
```

```
## features in all eight datasets to run the random forest
## algorithms for h rf c rf cart rf and
## chaid rf we used a subspace size of  features in
## the first  datasets ie fbis re re tr wap and
##
## the computer journal vol
##
## no
##
##
##
##
##
## baoxun xu joshua zhexue huang graham williams yunming ye
## fbis
##
## re
##
##
##
##
##
##
##
##
##
##
## strength
##
## strength
##
##
##
##
##
```

```
##
##
## hwrf
##
##
##
## cwrf
## cartwrf
##
##
##
## hwrf
## cwrf
##
##
##
## cartwrf
##
## chaidwrf
##
## chaidwrf
##
##
##
## hrf
##
##
##
## hrf
##
## crf
##
## crf
##
##
##
```

```
## cartrf
##
##
##
## cartrf
##
## chaidrf
##
## chaidrf
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
```

```
##
##
##
##
##
##
##
##
##
##
##
## number of features
##
## number of features
##
## re
##
##
##
##
##
## tr
##
##
##
##
##
##
##
##
##
##
##
##
```

```
## hwrf
## cwrf
##
##
##
## cartwrf
##
## strength
##
## strength
##
##
##
## hwrf
## cwrf
##
##
##
## cartwrf
##
## chaidwrf
##
##
##
## chaidwrf
##
## hrf
##
## hrf
##
##
##
## crf
##
##
##
```

```
## crf
##
## cartrf
##
## cartrf
##
##
##
## chaidrf
##
## chaidrf
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
```

```
##
##
##
##
##
##
##
##
##
##
##
##
##
##
## number of features
##
## number of features
##
## wap
##
## tr
##
##
##
##
##
##
##
##
##
##
## strength
##
```

```
## hwrf
##
##
##
## cwrf
## cartwrf
##
##
##
## strength
##
##
##
##
##
##
## hwrf
## cwrf
##
##
##
## cartwrf
## chaidwrf
##
## chaidwrf
## hrf
##
##
##
## hrf
##
##
##
## crf
##
## crf
```

```
##
## cartrf
##
## cartrf
##
##
##
##
##
## chaidrf
##
## chaidrf
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
```

```
##
##
##
##
## las
##
##
##
##
##
##
##
##
##
## las
##
##
##
##
##
##
##
##
##
##
##
##
## hwrf
##
##
##
## cwrf
```

```
## cartwrf
## chaidwrf
##
##
##
## strength
##
## strength
##
##
##
## number of features
##
## number of features
##
##
##
##
##
## hwrf
##
##
##
## cwrf
## cartwrf
##
##
##
## chaidwrf
##
## hrf
## crf
##
##
##
## hrf
```

```
##
##
##
## crf
##
## cartrf
## chaidrf
##
## cartrf
##
##
##
## chaidrf
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
```

```
##
##
##
##
##
##
##
##
## number of features
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
## number of features
##
## figure  strength changes against the number of features in the subspace on the  l
```

```
##
## tr to run the random forest algorithms and used
## a subspace size of  features in the last  datasets
## las and las to run these random forest algorithms
## for h w rf c w rf cart w rf and
## chaid w rf we used breimans subspace size of
##
## log m    to run these random forest algorithms
## this number of features provided a consistent result as
## shown in fig  in order to obtain stable results we
## built  random forest models for each random forest
## algorithm and each dataset and present the average
##
## the computer journal vol
##
## no
##
##
##
## hybrid weighted random forests for classifying very highdimensional data
## fbis
##
##
##
## re
##
##
##
##
##
##
##
##
##
## correlation
##
```

```
## correlation
##
##
##
##
##
##
##
##
## hwrf
## cwrf
##
##
##
## cartwrf
## chaidwrf
##
##
##
## hwrf
##
##
##
## cwrf
## cartwrf
##
##
##
## chaidwrf
##
## hrf
## crf
##
##
##
## hrf
```

```
##
##
##
## crf
##
## cartrf
## chaidrf
##
## cartrf
##
##
##
## chaidrf
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
```

```
##
##
##
##
##
## number of features
##
##
##
##
##
##
##
##
##
##
##
## number of features
##
## re
##
##
##
##
##
## tr
##
##
##
##
##
##
##
##
```

```
##
## hwrf
## cwrf
##
##
##
## cartwrf
## chaidwrf
##
##
##
## correlation
##
## correlation
##
##
##
##
##
## hwrf
##
##
##
## cwrf
## cartwrf
##
##
##
## chaidwrf
## hrf
##
## hrf
## crf
##
##
##
```

```
## crf
##
##
##
## cartrf
##
## cartrf
##
##
##
## chaidrf
##
## chaidrf
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
```

```
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
## number of features
##
## number of features
##
## wap
##
## tr
##
##
##
##
##
##
##
##
## correlation
##
```

```
##
##
##
##
## hwrf
## cwrf
##
##
##
## cartwrf
## chaidwrf
##
##
##
## correlation
##
##
##
##
##
## hwrf
## cwrf
##
##
##
## cartwrf
## chaidwrf
##
##
##
## hrf
##
## hrf
##
## crf
##
```

```
##
##
## crf
##
## cartrf
##
## cartrf
##
##
##
## chaidrf
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
## chaidrf
##
```

```
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
## number of features
##
## number of features
##
## las
##
## las
##
##
##
##
##
##
##
##
##
```

```
##
##
##
##
## hwrf
##
##
##
## cwrf
## cartwrf
##
##
##
## chaidwrf
##
##
##
## correlation
##
## correlation
##
##
##
##
##
##
## hwrf
## cwrf
##
##
##
## cartwrf
## chaidwrf
##
##
```

```
##
## hrf
##
## hrf
## crf
##
##
##
## crf
##
##
##
## cartrf
##
## cartrf
## chaidrf
##
## chaidrf
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
```

```
##
##
##
##
##
##
##
##
##
##
##
##
##
## number of features
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
```

```
##
##
##
##
## number of features
##
## figure  correlation changes against the number of features in the subspace on the
##
## results noting that the range of values are less than
##  and the hybrid trees are always more accurate
## the comparison results of classification performance
## of eleven methods are shown in table
## the
## performance is estimated using test accuracy acc
##
## micro f mic and macro f mac boldface
## denotes best results between eleven classification
## methods
## while the improvement is often quite
## small there is always an improvement demonstrated
## we observe that our proposed method h w rf
##
## the computer journal vol
##
## no
##
##
##
##
##
## baoxun xu joshua zhexue huang graham williams yunming ye
## fbis
##
##
##
## re
```

```
##
##
##
##
## log m
##
##
##
##
##
##
##
##
##
## cwrf
##
##
##
##
##
##
##
## hwrf
##
## c s
##
## c s
##
##
##
##
##
## cartwrf
##
```

```
## hwrf
## cwrf
##
##
##
## cartwrf
##
## chaidwrf
##
##
##
## chaidwrf
##
##
##
## hrf
##
## hrf
##
## log m
##
##
## crf
##
##
##
## crf
##
##
##
## cartrf
##
## cartrf
##
## chaidrf
##
```

```
##
##
## chaidrf
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
## number of features
##
##
##
##
```

```
##
##
##
##
##
##
##
##
## log m
##
##
##
##
##
##
##
## c s
##
## hwrf
##
##
##
##
##
##
##
##
##
##
## tr
##
##
##
##
```

```
##
## c s
##
##
##
## number of features
##
## re
##
##
##
##
##
## cwrf
##
##
##
##
##
## hwrf
## cwrf
##
##
##
## cartwrf
##
## cartwrf
##
## chaidwrf
##
##
##
## chaidwrf
##
##
##
```

```
## hrf
##
## hrf
##
## crf
##
## log m
##
##
##
##
## crf
##
##
##
## cartrf
##
## cartrf
##
## chaidrf
##
##
##
## chaidrf
##
##
##
##
##
##
##
##
##
##
##
```

```
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
## number of features
##
## number of features
##
## wap
```

```
##
## tr
##
##
##
##
##
##
## log m
##
## log m
##
##
##
##
##
##
##
## cwrf
##
## c s
##
##
##
## c s
##
## hwrf
##
##
##
##
##
##
##
```

```
##
##
## hwrf
##
##
##
## cwrf
## cartwrf
##
## cartwrf
##
##
## chaidwrf
##
##
##
## chaidwrf
## hrf
##
## hrf
## crf
##
##
##
## crf
##
##
##
## cartrf
##
## cartrf
## chaidrf
##
## chaidrf
##
##
```

```
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
```

```
##
##
##
##
##
##
## number of features
##
## number of features
##
## las
##
## las
##
##
##
##
##
##
##
##
##
## cwrf
## cartwrf
##
##
##
##
## hrf
##
##
```

```
## crf
## cartwrf
##
##
##
## chaidrf
##
##
##
##
##
##
##
##
##
## hwrf
##
##
##
## cwrf
## cartwrf
##
##
##
## chaidwrf
##
## log m
##
##
##
## c s
##
## c s
##
##
##
```

```
##
## hwrf
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
## chaidwrf
##
## log m
##
##
##
## hrf
## crf
##
##
## cartrf
## chaidrf
##
```

```
##
##
##
##
##
##
##
## number of features
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
## number of features
##
## figure  c s changes against the number of features in the subspace on the  high d
##
```

```
## outperformed the other classification methods in all
## datasets
##
##
##
## conclusions
##
## in this paper we presented a hybrid weighted random
## forest algorithm by simultaneously using a feature
## weighting method and a hybrid forest method to classify
## the computer journal vol
##
## no
##
##
##
## hybrid weighted random forests for classifying very highdimensional data
## fbis
##
##
##
##
##
## re
##
##
##
##
##
##
##
##
##
##
## hwrf
## cwrf
```

```
##
##
##
## cartwrf
##
## accuracy
##
## accuracy
##
##
##
## chaidwrf
## hrf
##
##
##
##
## hwrf
##
##
##
## cwrf
## cartwrf
##
##
##
## chaidwrf
## hrf
##
##
##
## crf
##
## crf
##
## log m
```

```
##
## cartrf
##
##
##
##
##
## cartrf
##
## log m
##
##
##
##
##
## chaidrf
##
## chaidrf
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
```

```
##
##
##
##
##
##
##
##
##
## number of features
##
##
##
##
##
##
##
##
##
##
##
##
## number of features
##
## re
##
## tr
##
##
##
##
```

```
##
##
##
## log m
##
##
##
##
##
##
##
## hwrf
## cwrf
##
##
##
## cartwrf
##
##
##
## accuracy
##
## accuracy
##
##
##
## chaidwrf
##
##
##
## hwrf
##
##
##
## cwrf
```

```
## cartwrf
##
##
##
## chaidwrf
## hrf
##
## hrf
##
##
##
## log m
##
##
##
## crf
##
## crf
##
##
##
## cartrf
##
## cartrf
##
##
##
##
## chaidrf
##
## chaidrf
##
##
##
##
```

```
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
```

```
## number of features
##
## number of features
##
## wap
##
##
##
##
##
## tr
##
##
##
##
##
##
##
##
##
##
##
##
## hwrf
## cwrf
##
##
##
## cartwrf
##
## accuracy
##
## accuracy
##
##
```

```
##
##
##
##
##
## hwrf
## cwrf
##
##
##
## cartwrf
## chaidwrf
##
## chaidwrf
##
##
##
## hrf
##
## log m
##
##
##
##
## hrf
##
## crf
##
##
##
## cartrf
##
##
##
## crf
##
```

```
## log m
##
##
##
##
## cartrf
## chaidrf
##
## chaidrf
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
```

```
##
##
##
##
##
##
##
##
##
##
##
## number of features
##
## number of features
##
## las
##
##
##
##
##
## las
##
##
##
##
##
##
##
## accuracy
##
##
## hwrf
## cwrf
```

```
##
##
##
## cartwrf
## chaidwrf
##
##
##
## accuracy
##
##
##
##
##
##
##
## hwrf
## cwrf
## cartwrf
##
##
##
## chaidwrf
## hrf
##
## hrf
##
## log m
##
##
##
## crf
##
##
##
## crf
```

```
##
## log m
##
##
##
##
## cartrf
##
## cartrf
## chaidrf
##
## chaidrf
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
```

```
##
##
##
##
##
##
##
##
##
## number of features
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
## number of features
```

```
##
## figure  test accuracy changes against the number of features in the subspace on t
##
## high dimensional data our algorithm not only retains
## a small subspace size breimans formula log m
## for determining the subspace size to create accurate
## random forest models but also effectively reduces
## the upper bound of the generalization error and
##
## improves classification performance from the results of
## experiments on various high dimensional datasets the
## random forest generated by our new method is superior
## to other classification methods we can use the default
## log m    subspace size and generally guarantee
##
## the computer journal vol
##
## no
##
##
##
##
##
## baoxun xu joshua zhexue huang graham williams yunming ye
##
## table  the comparison of results
## datasets
## dataset
## fbis
## measures
## acc
## mic
## svm
##
## knn
##
```

```
##
## nb
##
## h rf
##
## c rf
##
## cart rf
##
## chaid rf
##
## h w rf
##
## c w rf
##
## cart w rf
##
## chaid w rf
##
## dataset
## wap
## measures
## acc
## mic
## svm
##
##
## knn
##
## nb
##
## h rf
##
## c rf
##
## cart rf
```

```
##
## chaid rf
##
## h w rf
##
## c w rf
##
## cart w rf
##
##
## chaid w rf
##
##
## best accuracy micro f and macro f results of the eleven methods on the
## re
## mic
##
##
##
##
##
##
##
##
##
##
##
## tr
## mac
## acc
## mic
##
##
##
##
##
```

```
##
##
##
##
##
##
##
##
##
##
##
## mac
##
##
##
##
##
##
##
##
##
##
##
## acc
##
##
##
##
##
##
##
##
##
##
```

```
##
##
##
## to always produce the best models on a variety of
## measures by using the hybrid weighted random forest
## algorithm
## acknowledgements
## this research is supported in part by nsfc under
## grant no and shenzhen new industry development fund under grant nocxba
## references
##  breiman l  random forests machine learning
##
##  ho t  random subspace method for constructing decision forests ieee transactions
## analysis and machine intelligence
##  quinlan j  c programs for machine
## learning morgan kaufmann
##  breiman l  classification and regression trees
## chapman  hall crc
##  breiman l  bagging predictors
## machine
## learning
##  ho t  random decision forests proceedings
## of the third international conference on document
## analysis and recognition pp  ieee
##  dietterich t  an experimental comparison of
## three methods for constructing ensembles of decision
## trees bagging boosting and randomization machine
## learning
##
## mac
##
##
##
##
##
##
```

```
##
##
##
##
##
## mac
##
##
##
##
##
##
##
##
##
##
## re
## mic
##
##
##
##
##
##
##
##
##
##
## las
## acc
## mic
##
##
```

```
##
##
##
##
##
##
##
##
##
##
## acc
##
##
##
##
##
##
##
##
##
##
## tr
## mic
##
##
##
##
##
##
##
##
##
```

```
##
##
## las
## mac
## acc
## mic
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
## mac
##
##
##
##
##
##
##
##
##
##
```

```
##
##
## acc
##
##
##
##
##
##
##
##
##
##
## mac
##
##
##
##
##
##
##
##
##
##
##
## mac
##
##
##
##
##
##
##
##
##
```

```
##
##
##
##
##  banfield r hall l bowyer k and kegelmeyer w
##  a comparison of decision tree ensemble creation
## techniques ieee transactions on pattern analysis
## and machine intelligence
##
##  robniksikonja
## m  improving random forests
## proceedings of the th european conference on
## machine learning pp  springer
##  ho t  c decision forests proceedings of
## the fourteenth international conference on pattern
## recognition pp  ieee
##  dietterrich t  machine learning research four
## current direction artificial intelligence magzine
##
##  amaratunga d cabrera j and lee y
## enriched random forests bioinformatics
##
##  ye y li h deng x and huang j
## feature weighting random forest for detection of hidden
## web search interfaces the journal of computational
## linguistics and chinese language processing
##
##  xu b huang j williams g wang q and
## ye y  classifying very highdimensional data
## with random forests built from small subspaces
## international journal of data warehousing and
## mining
##  xu b huang j williams g li j and ye y
##  hybrid random forests advantages of mixed
## trees in classifying text data proceedings of the th
## pacificasia conference on knowledge discovery and
```

```
## data mining springer
##
## the computer journal vol
##
## no
##
##
##
## hybrid weighted random forests for classifying very highdimensional data
##  biggs d de ville b and suen e  a method
## of choosing multiway partitions for classification and
## decision trees journal of applied statistics
##  ture m kurt i turhan kurum a and ozdamar
## k  comparing classification techniques for
## predicting essential hypertension expert systems with
## applications
##  begum n ma f and ren f  automatic text summarization using support vector machi
## international journal of innovative computing information and control
##  chen j huang h tian s and qu y
## feature selection for text classification with naive
## bayes expert systems with applications
##
##  tan s  neighborweighted knearest neighbor
## for unbalanced text corpus
## expert systems with
## applications
##  pearson k  on the theory of contingency and
## its relation to association and normal correlation
## cambridge university press
##  yang y and liu x  a reexamination of
## text categorization methods proceedings of the th
## international conference on research and development
## in information retrieval pp  acm
##  han e and karypis g  centroidbased
## document classification analysis and experimental
## results proceedings of the th european conference on
```

```
## principles of data mining and knowledge discovery
## pp  springer
##  trec
##
## text
## retrieval
## conference
## http  trecnistgov
##  lewis
## d
##
## reuters
## text
## categorization
## test
## collection
## distribution
##
## http  wwwresearchattcom  lewis
##  han e boley d gini m gross r hastings
## k karypis g kumar v mobasher b and
## moore j  webace a web agent for document
## categorization and exploration proceedings of the nd
## international conference on autonomous agents pp
##  acm
##  mccallum a and nigam k  a comparison of
## event models for naive bayes text classification aaai workshop on learning for te
##
##  witten i frank e and hall m  data mining
## practical machine learning tools and techniques
## morgan kaufmann
##
## the computer journal vol
##
## no
```

Punctuation can provide gramatical context which supports understanding. Often for initial analyses we ignore the punctuation. Later we will use punctuation to support the extraction of meaning.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

21.13 Remove English Stop Words

```
docs <- tm_map(docs, removeWords, stopwords("english"))
```

```
inspect(docs[16])
```

```
## <<SimpleCorpus>>
## Metadata:  corpus specific: 1, document level (indexed): 0
## Content:  documents: 1
##
##
## hybrid weighted random forests \nclassifying  highdimensional data\nbaoxun xu  j
```

Stop words are common words found in a language. Words like *for*, *very*, *and*, *of*, *are*, etc, are common stop words. Notice they have been removed from the above text.

We can list the stop words:

```
length(stopwords("english"))
```

```
## [1] 174
```

```
stopwords("english")
```

```
##   [1] "i"          "me"         "my"         "myself"     "we"
##   [6] "our"        "ours"       "ourselves"  "you"        "your"
##  [11] "yours"      "yourself"   "yourselves" "he"         "him"
##  [16] "his"        "himself"    "she"        "her"        "hers"
##  [21] "herself"    "it"         "its"        "itself"     "they"
##  [26] "them"       "their"      "theirs"     "themselves" "what"
##  [31] "which"      "who"        "whom"       "this"       "that"
##  [36] "these"      "those"      "am"         "is"         "are"
##  [41] "was"        "were"       "be"         "been"       "being"
##  [46] "have"       "has"        "had"        "having"     "do"
##  [51] "does"       "did"        "doing"      "would"      "should"
##  [56] "could"      "ought"      "i'm"        "you're"     "he's"
##  [61] "she's"      "it's"       "we're"      "they're"    "i've"
##  [66] "you've"     "we've"      "they've"    "i'd"        "you'd"
##  [71] "he'd"       "she'd"      "we'd"       "they'd"     "i'll"
##  [76] "you'll"     "he'll"      "she'll"     "we'll"      "they'll"
##  [81] "isn't"      "aren't"     "wasn't"     "weren't"    "hasn't"
##  [86] "haven't"    "hadn't"     "doesn't"    "don't"      "didn't"
##  [91] "won't"      "wouldn't"   "shan't"     "shouldn't"  "can't"
##  [96] "cannot"     "couldn't"   "mustn't"    "let's"      "that's"
## [101] "who's"      "what's"     "here's"     "there's"    "when's"
## [106] "where's"    "why's"      "how's"      "a"          "an"
## [111] "the"        "and"        "but"        "if"         "or"
## [116] "because"    "as"         "until"      "while"      "of"
## [121] "at"         "by"         "for"        "with"       "about"
## [126] "against"    "between"    "into"       "through"    "during"
## [131] "before"     "after"      "above"      "below"      "to"
## [136] "from"       "up"         "down"       "in"         "out"
## [141] "on"         "off"        "over"       "under"      "again"
## [146] "further"    "then"       "once"       "here"       "there"
## [151] "when"       "where"      "why"        "how"        "all"
## [156] "any"        "both"       "each"       "few"        "more"
## [161] "most"       "other"      "some"       "such"       "no"
## [166] "nor"        "not"        "only"       "own"        "same"
## [171] "so"         "than"       "too"        "very"
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

21.14 Remove Own Stop Words

```
docs <- tm_map(docs, removeWords, c("department", "email"))
```

```
viewDocs(docs, 16)
```

```
## hybrid weighted random forests
## classifying  highdimensional data
## baoxun xu  joshua zhexue huang  graham williams
## yunming ye
##
##
##   computer science harbin institute  technology shenzhen graduate
## school shenzhen  china
##
## shenzhen institutes  advanced technology chinese academy  sciences shenzhen
##  china
##  amusing gmailcom
## random forests   popular classification method based   ensemble
## single type  decision trees  subspaces  data   literature
##  many different types  decision tree algorithms including c cart
## chaid  type  decision tree algorithm may capture different information
##  structure  paper proposes  hybrid weighted random forest algorithm
## simultaneously using  feature weighting method   hybrid forest method
## classify  high dimensional data  hybrid weighted random forest algorithm
## can effectively reduce subspace size  improve classification performance
## without increasing  error bound  conduct  series  experiments  eight
## high dimensional datasets  compare  method  traditional random forest
## methods   classification methods  results show   method
## consistently outperforms  traditional methods
## keywords random forests hybrid weighted random forest classification decision tre
##
##
##
## introduction
##
## random forests     popular classification
## method  builds  ensemble   single type
##  decision trees  different random subspaces
## data  decision trees  often either built using
## c   cart    one type within
```

```
##  single random forest  recent years random
## forests  attracted increasing attention due
##   competitive performance compared
## classification methods especially  highdimensional
## data  algorithmic intuitiveness  simplicity
##    important capability  ensemble using
## bagging   stochastic discrimination
## several methods   proposed  grow random
## forests  subspaces  data
##  methods   popular forest construction
## procedure  proposed  breiman   first use
## bagging  generate training data subsets  building
## individual trees
##  subspace  features
## randomly selected   node  grow branches
##  decision tree  trees   combined
## ensemble   forest   ensemble learner
## performance   random forest  highly dependent
##  two factors  performance   tree
## diversity   trees   forests  breiman
## formulated  overall performance   set  trees
##  average strength  proved   generalization
##
## error   random forest  bounded   ratio
## average correlation  trees divided   square
##   average strength   trees
##   high dimensional data   text data
##   usually  large portion  features
## uninformative   classes   forest building
## process informative features    large
## chance   missed   randomly select  small
## subspace breiman suggested selecting log m
## features   subspace  m   number
## independent features   data  high dimensional
## data    result weak trees  created
## subspaces  average strength   trees  reduced
```

```
##   error bound   random forest  enlarged
## therefore   large proportion   weak
## trees  generated   random forest  forest
## large likelihood  make  wrong decision  mainly
## results   weak trees classification power
##  address  problem  aim  optimize decision
## trees   random forest  two strategies one
## straightforward strategy   enhance  classification
## performance  individual trees   feature weighting
## method  subspace sampling
## method feature weights  computed  respect
##   correlations  features   class feature
##  regarded   probabilities   feature
##  selected  subspaces  method obviously
## increases  classification performance  individual
##
##  computer journal vol
##
##
##
##
##
##
##
## baoxun xu joshua zhexue huang graham williams yunming ye
##
## trees   subspaces will  biased  contain
##  informative features however  chance
## correlated trees  also increased since  features
## large weights  likely   repeatedly selected
##  second strategy   straightforward use
## several different types  decision trees   training
## data subset  increase  diversity   trees
##   select  optimal tree   individual
## tree classifier   random forest model  work
## presented  extends  algorithm developed
```

```
## specifically  build three different types  tree
## classifiers c cart  chaid
## training data subset   evaluate  performance
##   three classifiers  select  best tree
##  way  build  hybrid random forest  may
## include different types  decision trees   ensemble
##  added diversity   decision trees can effectively
## improve  accuracy   tree   forest
## hence  classification performance   ensemble
## however   use  method  build  best
## random forest model  classifying high dimensional
## data  can   sure   subspace size  best
##   paper  propose  hybrid weighted random
## forest algorithm  simultaneously using  new feature
## weighting method together   hybrid random
## forest method  classify high dimensional data
##  new random forest algorithm  calculate feature
## weights  use weighted sampling  randomly select
## features  subspaces   node  building different
## types  trees classifiers c cart  chaid
##  training data subset  select  best tree
##  individual tree   final ensemble model
## experiments  performed   high dimensional
## text datasets  dimensions ranging
##   compared  performance  eight random
## forest methods  wellknown classification methods
## c random forest cart random forest chaid
## random forest hybrid random forest c weighted
## random forest cart weighted random forest chaid
## weighted random forest hybrid weighted random
## forest support vector machines  naive bayes
##  knearest neighbors
##  experimental
## results show   hybrid weighted random forest
## achieves improved classification performance
## ten competitive methods
```

```
##  remainder   paper  organized  follows
##  section   introduce  framework  building
##  hybrid weighted random forest  describe  new
## random forest algorithm section  summarizes four
## measures  evaluate random forest models  present
## experimental results   high dimensional text datasets
##  section  section  contains  conclusions
##
## table  contingency table  input feature   class
## feature y
## y  y
## y  yj
## y  yq total
##
##
##
## j
##
## q
##
##
##
##
##
##
##
##
##
##
##
##
##
##   ai
##
##
```

```
## ij
##
## iq
##
##
##
##
##
##
##
##
##
##
##
##
##
##   ap
## p
##
## pj
##
## pq
## p
## total
##
##
## j
##
## q
##
##
## general framework  building hybrid random forests
##  integrating  two methods  propose  novel
## hybrid weighted random forest algorithm
##
```

```
##
## let y   class  target feature  q distinct
## class labels yj  j       q   purposes
##  discussion  consider  single categorical feature
##   dataset d  p distinct category values
## denote  distinct values  ai         p
## numeric features can  discretized  p intervals
##  supervised discretization method
## assume d  val objects  size   subset
## d satisfying  condition    ai  y  yj
## denoted  ij  considering  combinations
## categorical values     labels  y   can
## obtain  contingency table     y  shown
##  table   far right column contains  marginal
## totals  feature
##
## hybrid
## forests
##
## weighted
##
## random
##
##   section  first introduce  feature weighting
## method  subspace sampling   present
##
## q
##
##
##
##
## ij
##
##         p
##
##
```

```
##
## j
##
##   bottom row   marginal totals  class
## feature y
## j
##
## p
##
##
## ij
##
##  j       q
##
##
##
##
##
##  grand total  total number  samples
##  bottom right corner
##
##
## q
## p
##
##
## ij
##
##
##
## j
##
## given  training dataset d  feature   first
## compute  contingency table  feature weights
##  computed using  two methods   discussed
##   following subsection
```

```
##
##
##
##
## notation
##
## feature weighting method
##
##   subsection  give  details   feature
## weighting method  subspace sampling  random
## forests consider  mdimensional feature space
##            present   compute
##
##  computer journal vol
##
##
##
##
##
## hybrid weighted random forests  classifying  highdimensional data
## weights w  w      wm   every feature   space
##  weights   used   improved algorithm
##  grow  decision tree   random forest
##  feature weight computation
##  weight  feature  represents  correlation
##   values  feature    values
## class feature y   larger weight will indicate
## class labels  objects   training dataset
## correlated   values  feature  indicating
##    informative   class  objects thus
##  suggested     stronger power  predicting
##  classes  new objects
##   following  propose  use  chisquare
## statistic  compute feature weights
## method can quantify  correspondence  two
## categorical variables
```

```
## given  contingency table   input feature
##  class feature y  dataset d  chisquare statistic
##   two features  computed
## corra y
##
## q
## p
##
## ij  tij
## tij
##  j
##
##
##
##  ij   observed frequency
## contingency table  tij   expected frequency
## computed
##  x j
## tij
##
##
##
##
##  larger  measure corra y
## informative  feature    predicting class y
##  normalized feature weight
##  practice feature weights  normalized  feature
## subspace sampling  use corra y   measure
## informativeness   features  consider
##  feature weights however  treat  weights
## probabilities  features  normalize  measures
## ensure  sum   normalized feature weights
## equal   let corrai  y      m    set
##  m feature measures  compute  normalized
## weights
##
```

```
## corrai  y
## wi  n
##
##  corrai  y
##   use  square root  smooth  values
##  measures wi can  considered   probability
##  feature ai  randomly sampled   subspace
##  informative  feature   larger  weight
##  higher  probability   feature  selected
##
## diversity  commonly obtained  using bagging
## random subspace sampling  introduce
## element  diversity  using different types  trees
## considering  analogy  forestry  different data subsets  bagging represent  soil s
## one   use three types  decision tree algorithms
## generate three different tree classifiers   training data subset     evaluate  ac
##   tree   measure  tree importance
## paper  use  outofbag accuracy  assess  importance   tree
## following breiman   use bagging  generate
##  series  training data subsets    build
## trees   tree  data subset used  grow
##  tree  called  inofbag iob data
## remaining data subset  called  outofbag oob
## data since oob data   used  building trees
##  can use  data  objectively evaluate  trees
## accuracy  importance  oob accuracy gives
## unbiased estimate   true accuracy   model
## given n instances   training dataset d   tree
## classifier hk iobk  built   kth training data
## subset iobk   define  oob accuracy   tree
## hk iobk   di  d
## n
## oobacck
##
## framework  building  hybrid random
## forest
```

```
##
##   ensemble learner  performance   random
## forest  highly dependent  two factors  diversity
## among  trees   accuracy   tree
##
##
##
## ihk di   yi  di  oobk
## n
##  idi  oobk
##
##
##
##     indicator function  larger
## oobacck   better  classification quality   tree
##  use  outofbag data subset oobi  calculate
##  outofbag accuracies   three types  trees
## c cart  chaid  evaluation values e
## e  e respectively
## fig  illustrates  procedure  building  hybrid
## random forest model firstly  series  iob oob
## datasets  generated   entire training dataset
##  bagging  three types  tree classifiers c
## cart  chaid  built using  iob dataset
##  corresponding oob dataset  used  calculate
## oob accuracies   three tree classifiers finally
##  select  tree   highest oob accuracy
##  final tree classifier   included   hybrid
## random forest
## building  hybrid random forest model
## way will increase  diversity among  trees
##  classification performance   individual tree
## classifier  also maximized
##
##
##
```

```
##
##
##
## decision tree algorithms
##
##  core   approach   diversity  decision
## tree algorithms   random forest different decision
## tree algorithms grow structurally different trees
##   training data selecting  good decision tree
## algorithm  grow trees   random forest  critical
##
##  computer journal vol
##
##
##
##
##
##
##
## baoxun xu joshua zhexue huang graham williams yunming ye
##  difference lies   way  split  node
##   split functions  binary branches  multibranches   work  use  different decisio
## tree algorithms  build  hybrid random forest
##
##
##
## figure   hybrid random forests framework
##
##   performance   random forest  studies
##  considered  different decision tree algorithms
## affect  random forest      paper
##  common decision tree algorithms   follows
## classification trees  c   supervised
## learning classification algorithm used  construct
## decision trees given  set  preclassified objects
## described   vector  attribute values  construct
```

```
##  mapping  attribute values  classes c uses
##  divideandconquer approach  grow decision trees
## beginning   entire dataset  tree  constructed
##  considering  predictor variable  dividing
## dataset  best predictor  chosen   node
## using  impurity  diversity measure  goal
##  produce subsets   data   homogeneous
##  respect   target variable c selects  test
##  maximizes  information gain ratio igr
## classification  regression tree cart
##  recursive partitioning method  can  used
##  regression  classification  main difference
##  c  cart   test selection
## evaluation process
## chisquared automatic interaction detector
## chaid method  based   chisquare test
## association  chaid decision tree  constructed
##  repeatedly splitting subsets   space  two
##   nodes  determine  best split
## node  allowable pair  categories   predictor
## variables  merged     statistically
## significant difference within  pair  respect
## target variable
##   decision tree algorithms  can see
##
## hybrid weighted random forest algorithm
##
##   subsection  present  hybrid weighted
## random forest algorithm  simultaneously using
## feature weights   hybrid method  classify high
## dimensional data  benefits   algorithm
## two aspects firstly compared  hybrid forest
## method   can use  small subspace size
## create accurate random forest models
## secondly
## compared  building  random forest using feature
```

```
## weighting   can use several different types
## decision trees   training data subset  increase
##  diversities  trees  added diversity
## decision trees can effectively improve  classification
## performance   ensemble model  detailed steps
##  introduced  algorithm
## input parameters  algorithm  include  training
## dataset d  set  features   class feature y
##  number  trees   random forest k
## size  subspaces m  output   random forest
## model m  lines  form  loop  building k
## decision trees   loop line  samples  training
## data d  sampling  replacement  generate
## inofbag data subset iobi  building  decision tree
## line  build three types  tree classifiers c
## cart  chaid   procedure line  calls
##  function createt reej   build  tree classifier
## line  calculates  outofbag accuracy   tree
## classifier   procedure line  selects  tree
## classifier   maximum outofbag accuracy k
## decision tree trees  thus generated  form  hybrid
## weighted random forest model m
## generically function createt reej  first creates
## new node   tests  stopping criteria  decide
## whether  return   upper node   split
## node   choose  split  node   feature
## weighting method  used  randomly select m features
##   subspace  node splitting  features
##  used  candidates  generate  best split
## partition  node   subset   partition
## createt reej   called   create  new node
##  current node   leaf node  created  returns
##  parent node  recursive process continues
##  full tree  generated
##
##  computer journal vol
```

```
##
##
##
##
##
## hybrid weighted random forests  classifying  highdimensional data
## algorithm  new random forest algorithm
##  input
##   d   training dataset
##      features space
##   y   class features space y  y   yq
##   k   number  trees
##   m   size  subspaces
##  output  random forest m
##  method
##       k
##
## draw  bootstrap sample inofbag data subset
## iobi  outofbag data subset oobi
## training dataset d
##
##  j
##
## hij iobi   createt reej
## use outofbag data subset oobi  calculate
##
##  outofbag accuracy oobacci j   tree
## classifier hij iobi   equation
##
## end
##
## select hi iobi    highest outofbag
## accuracy oobacci  optimal tree
##  end
##  combine
##
```

```
## k
## tree
## classifiers
## h iob  h iob   hk iobk    random
## forest m
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
## function createtree
## create  new node n
##  stopping criteria  met
## return n   leaf node
## else
##  j    m
## compute
##
## informativeness
## measure
## corraj  y   equation
## end
## compute feature weights w  w   wm
```

```
## equation
## use  feature weighting method  randomly
## select m features
## use  m features  candidates  generate
##  best split   node   partitioned
## call createtree   split
## end
## return n
## evaluation measures
##
##   paper  use five measures ie strength
## correlation error bound c s  test accuracy  f
## metric  evaluate  random forest models strength
## measures  collective performance  individual trees
##   random forest   correlation measures
## diversity   trees  ratio   correlation
##   square   strength c s indicates
## generalization error bound   random forest model
##  three measures  introduced
## accuracy measures  performance   random forest
## model  unseen test data  f metric
##
##
##
## commonly used measure  classification performance
##
##
## strength  correlation measures
##
##  follow breimans method described
## calculate  strength correlation   ratio c s
## following breimans notation  denote strength
## s  correlation   let hk iobk    kth
## tree classifier grown   kth training data iobk
## sampled  d  replacement
## assume
```

```
## random forest model contains k trees  outofbag
## proportion  votes  di  d  class j
## k
## ihk di   j di
##   iobk
## qdi  j  kk
##
##   iobk
## k idi
##    number  trees   random forest
##   trained without di  classify di  class
## j divided   number  training datasets
## containing di
##  strength s  computed
##
## qdi  yi   maxjyi qdi  j
## n
## n
##
## s
##
##
##
##  n   number  objects  d  yi indicates
##  true class  di
##  correlation   computed
## n
##
##
##
##  qdi  yi   maxjyi qdi  j  s
## n
##
##
##
##
```

```
## k
##
## k
## k  pk  pk
## k pk  p
##
##
## n
## pk
##
##
##
## ihk di   yi  di
##   iobk
## n
##   iobk
##  idi
##
##
##
##
## n
## pk
##
##
##
## ihk di   jdi  y  di
##   iobk
## n
## id
##
##
## iob
##
##
## k
```

```
##
##
##
##
##
## jdi  y   argmaxjyi qd j
##
##
##
##   class  obtains  maximal number  votes
## among  classes   true class
##
##
## general error bound measure c s
##
## given  strength  correlation  outofbag
## estimate   c s measure can  computed
##  important theoretical result  breimans method
##   upper bound   generalization error
## random forest ensemble   derived
## p e    s  s
##
##
##
##     mean value  correlations
## pairs  individual classifiers  s   strength
##  set  individual classifiers   estimated
##
##  computer journal vol
##
##
##
##
##
##
```

```
## baoxun xu joshua zhexue huang graham williams yunming ye
##
## average accuracy  individual classifiers  d
## outofbag evaluation  inequality shows
## generalization error   random forest  affected
##  strength  individual classifiers   mutual
## correlations therefore breiman defined  c s ratio
##  measure  random forest
## c s   s
##
##
##
##  smaller  ratio  better  performance
##  random forest   c s gives guidance
## reducing  generalization error  random forests
##
##
## test accuracy
##
##  test accuracy measures  classification performance   random forest   test data s
## dt   test data  yt   class labels given
## di  dt   number  votes  di  class j
## n di  j
##
## k
##
##
## ihk di   j
##
##
##
## table
## summary statistic   highdimensional
## datasets
## name
## features
```

```
## instances
## classes  minority
## fbis
##
##
##
##
## re
##
##
##
##
## re
##
##
##
##
## tr
##
##
##
##
## wap
##
##
##
##
## tr
##
##
##
##
## las
##
##
##
```

```
##
## las
##
##
##
##
##
##  emphasizes  performance   classifier  rare
## categories define     follows
##
##
##
## t pi
## t pi
##
## t pi  f pi
## t pi  f ni
##
##
##
## f    category    macroaveraged f
##  computed
##
## k
##
##  test accuracy  calculated
## f
##
##  di  yi   maxjyi n di  j
## n
##
##
##  m acrof
##
##
## q
```

```
##
##
## q
##
## f
##
##
##
## n
##
## acc
##
##  n   number  objects  dt  yi indicates
##  true class  di
##
##
## f metric
##
##  evaluate  performance  classification methods
##  dealing   unbalanced class distribution  use
##  f metric introduced  yang  liu
## measure  equal   harmonic mean  recall
##  precision   overall f score   entire
## classification problem can  computed   microaverage   macroaverage
## microaveraged f  computed globally
## classes  emphasizes  performance   classifier
##  common classes define     follows
## q
##
## q
## t pi
##  t pi
##   q
##    q
##
##  t pi  f pi
```

```
##  t pi  f ni
##  q   number  classes t pi true positives
##   number  objects correctly predicted  class
## f pi false positives   number  objects
## predicted  belong  class      microaveraged f  computed
## m icrof
##
##
##
##
##
##
## macroaveraged f  first computed locally
##  class    average   classes  taken
##
##  larger  microf  macrof values
## higher  classification performance   classifier
##
##
## experiments
##
##   section  present two experiments
## demonstrate  effectiveness   new random
## forest algorithm  classifying high dimensional data
## high dimensional datasets  various sizes
## characteristics  used   experiments
## first experiment  designed  show   proposed
## method can reduce  generalization error bound
## c s   improve test accuracy   size
##  selected subspace    large  second
## experiment  used  demonstrate  classification
## performance   proposed method  comparison
##  classification methods ie svm nb  knn
##
##
## datasets
```

```
##
##   experiments  used eight realworld high
## dimensional datasets  datasets  selected
## due   diversities   number  features
## number  instances   number  classes
## dimensionalities vary     instances
## vary       minority class rate varies
##       dataset  randomly
## select   instances   training dataset
##  remaining data   test dataset detailed
## information   eight datasets  listed  table
##  fbis re re tr wap tr las
##  las datasets  classical text classification
## benchmark datasets   carefully selected
##
##  computer journal vol
##
##
##
##
##
## hybrid weighted random forests  classifying  highdimensional data
## preprocessed  han  karypis  dataset fbis
##  compiled   foreign broadcast information
## service trec   datasets re  re
## selected   reuters text categorization test
## collection distribution    datasets tr
## tr  derived  trec  trec
##  trec  dataset wap    webace
## project wap   datasets las  las
## selected   los angeles times  trec
##  classes   datasets  generated
## relevance judgment provided   collections
##
##
## performance comparisons  random forest methods
```

```
##
##  purpose   experiment   evaluate
##  effect   hybrid weighted random forest
## method h w rf  strength correlation c s
##  test accuracy
##  eight high dimensional
## datasets  analyzed  results  compared
##  seven  random forest methods ie c
## random forest c rf cart random forest
## cart rf chaid random forest chaid rf
## hybrid random forest h rf c weighted random
## forest c w rf cart weighted random forest
## cart w rf chaid weighted random forest
## chaid w rf   dataset  ran
## random forest algorithm  different sizes
## feature subspaces since  number  features
## datasets   large  started   subspace
##   features  increased  subspace
## features  time   given subspace size  built
##  trees   random forest model  order
## obtain  stable result  built  random forest models
##   subspace size  dataset   algorithm
##  computed  average values   four measures
##  strength correlation c s   test accuracy
## final results  comparison  performance
## eight random forest algorithms   four measures
##      datasets  shown  figs
##
## fig  plots  strength   eight methods
## different subspace sizes      datasets
##    subspace  higher  strength
## better  result   curves  can see
##  new algorithm h w rf consistently performs
## better   seven  random forest algorithms
##  advantages   obvious  small subspaces
##  new algorithm quickly achieved higher strength
```

```
##   subspace size increases
##  seven
## random forest algorithms require larger subspaces
## achieve  higher strength  results indicate
##  hybrid weighted random forest algorithm enables
## random forest models  achieve  higher strength
##  small subspace sizes compared   seven
## random forest algorithms
## fig  plots  curves   correlations
## eight random forest methods    datasets
##
##
##
## small subspace sizes h rf c rf cart rf
##  chaid rf produce higher correlations
##  trees   datasets  correlation decreases
##   subspace size increases   random forest
## models  lower  correlation   trees
##   better  final model
##   new
## random forest algorithm h w rf  low correlation
## level  achieved   small subspaces
##  datasets  also note    subspace size
## increased  correlation level increased  well
## understandable    subspace size increases
##   informative features   likely
## selected repeatedly   subspaces increasing
## similarity   decision trees therefore  feature
## weighting method  subspace selection works well
## small subspaces  least   point  view
## correlation measure
## fig  shows  error bound indicator c s
## eight methods    datasets   figures
##  can observe    subspace size increases c s
## consistently reduces  behaviour indicates
## subspace size larger  log m  benefits  eight
```

```
## algorithms however  new algorithm h w rf
## achieved  lower level  c s  subspace size
## log m      seven  algorithms
## fig  plots  curves showing  accuracy
## eight random forest models   test datasets
##   datasets  can clearly see   new random
## forest algorithm h w rf outperforms  seven
##  random forest algorithms   eight data sets
##  can  seen   new method   stable
##  classification performance   methods
##    figures   observed   highest test
## accuracy  often obtained   default subspace size
##  log m     implies   practice large
## size subspaces   necessary  grow highquality
## trees  random forests
##
##
## performance comparisons
## classification methods
##
##
##
##
##
##  conducted   experimental comparison
##  three  widely used text classification
## methods support vector machines svm naive
## bayes nb  knearest neighbor knn
## support vector machine used  linear kernel
## regularization parameter     often
## used  text categorization  naive bayes
## adopted  multivariate bernoulli event model
##  frequently used  text classification   knearest neighbor knn  set  number k
## neighbors     experiments  used wekas
## implementation   three text classification
## methods   used  single subspace size
```

```
## features   eight datasets  run  random forest
## algorithms  h rf c rf cart rf
## chaid rf  used  subspace size   features
##  first  datasets ie fbis re re tr wap
##
##  computer journal vol
##
##
##
##
##
##
##
## baoxun xu joshua zhexue huang graham williams yunming ye
## fbis
##
## re
##
##
##
##
##
##
##
##
##
##
## strength
##
## strength
##
##
##
##
##
```

```
##
##
## hwrf
##
##
##
## cwrf
## cartwrf
##
##
##
## hwrf
## cwrf
##
##
##
## cartwrf
##
## chaidwrf
##
## chaidwrf
##
##
##
## hrf
##
##
##
## hrf
##
## crf
##
## crf
##
##
##
```

```
## cartrf
##
##
##
## cartrf
##
## chaidrf
##
## chaidrf
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
```

```
##
##
##
##
##
##
##
##
##
##
##
## number  features
##
## number  features
##
## re
##
##
##
##
##
## tr
##
##
##
##
##
##
##
##
##
##
##
##
```

```
## hwrf
## cwrf
##
##
##
## cartwrf
##
## strength
##
## strength
##
##
##
## hwrf
## cwrf
##
##
##
## cartwrf
##
## chaidwrf
##
##
##
## chaidwrf
##
## hrf
##
## hrf
##
##
##
## crf
##
##
##
```

```
## crf
##
## cartrf
##
## cartrf
##
##
##
## chaidrf
##
## chaidrf
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
```

```
##
##
##
##
##
##
##
##
##
##
##
##
##
##
## number  features
##
## number  features
##
## wap
##
## tr
##
##
##
##
##
##
##
##
##
##
## strength
##
```

```
## hwrf
##
##
##
## cwrf
## cartwrf
##
##
##
## strength
##
##
##
##
##
##
## hwrf
## cwrf
##
##
##
## cartwrf
## chaidwrf
##
## chaidwrf
## hrf
##
##
##
## hrf
##
##
##
## crf
##
## crf
```

```
##
## cartrf
##
## cartrf
##
##
##
##
##
## chaidrf
##
## chaidrf
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
```

```
##
##
##
##
## las
##
##
##
##
##
##
##
##
##
## las
##
##
##
##
##
##
##
##
##
##
##
##
## hwrf
##
##
##
## cwrf
```

```
## cartwrf
## chaidwrf
##
##
##
## strength
##
## strength
##
##
##
## number  features
##
## number  features
##
##
##
##
##
## hwrf
##
##
##
## cwrf
## cartwrf
##
##
##
## chaidwrf
##
## hrf
## crf
##
##
##
## hrf
```

```
##
##
##
## crf
##
## cartrf
## chaidrf
##
## cartrf
##
##
##
## chaidrf
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
```

```
##
##
##
##
##
##
##
##
## number  features
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
## number  features
##
## figure  strength changes   number  features   subspace    high dimensional datase
```

```
##
## tr  run  random forest algorithms  used
##  subspace size   features   last  datasets
## las  las  run  random forest algorithms
##  h w rf c w rf cart w rf
## chaid w rf  used breimans subspace size
##
## log m     run  random forest algorithms
##  number  features provided  consistent result
## shown  fig   order  obtain stable results
## built  random forest models   random forest
## algorithm   dataset  present  average
##
##  computer journal vol
##
##
##
##
##
## hybrid weighted random forests  classifying  highdimensional data
## fbis
##
##
##
## re
##
##
##
##
##
##
##
##
##
## correlation
##
```

```
## correlation
##
##
##
##
##
##
##
##
## hwrf
## cwrf
##
##
##
## cartwrf
## chaidwrf
##
##
##
## hwrf
##
##
##
## cwrf
## cartwrf
##
##
##
## chaidwrf
##
## hrf
## crf
##
##
##
## hrf
```

```
##
##
##
## crf
##
## cartrf
## chaidrf
##
## cartrf
##
##
##
## chaidrf
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
```

```
##
##
##
##
##
## number  features
##
##
##
##
##
##
##
##
##
##
##
## number  features
##
## re
##
##
##
##
##
## tr
##
##
##
##
##
##
##
##
```

```
##
## hwrf
## cwrf
##
##
##
## cartwrf
## chaidwrf
##
##
##
## correlation
##
## correlation
##
##
##
##
##
## hwrf
##
##
##
## cwrf
## cartwrf
##
##
##
## chaidwrf
## hrf
##
## hrf
## crf
##
##
##
```

```
## crf
##
##
##
## cartrf
##
## cartrf
##
##
##
## chaidrf
##
## chaidrf
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
```

```
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
## number  features
##
## number  features
##
## wap
##
## tr
##
##
##
##
##
##
##
##
## correlation
##
```

```
##
##
##
##
## hwrf
## cwrf
##
##
##
## cartwrf
## chaidwrf
##
##
##
## correlation
##
##
##
##
##
## hwrf
## cwrf
##
##
##
## cartwrf
## chaidwrf
##
##
##
## hrf
##
## hrf
##
## crf
##
```

```
##
##
## crf
##
## cartrf
##
## cartrf
##
##
##
## chaidrf
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
## chaidrf
##
```

```
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
## number  features
##
## number  features
##
## las
##
## las
##
##
##
##
##
##
##
##
```

```
##
##
##
##
## hwrf
##
##
##
## cwrf
## cartwrf
##
##
##
## chaidwrf
##
##
##
## correlation
##
## correlation
##
##
##
##
##
##
## hwrf
## cwrf
##
##
##
## cartwrf
## chaidwrf
##
##
```

```
##
## hrf
##
## hrf
## crf
##
##
##
## crf
##
##
##
## cartrf
##
## cartrf
## chaidrf
##
## chaidrf
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
```

```
##
##
##
##
##
##
##
##
##
##
##
##
##
## number  features
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
```

```
##
##
##
##
## number  features
##
## figure  correlation changes   number  features   subspace    high dimensional dat
##
## results noting   range  values  less
##    hybrid trees  always  accurate
##  comparison results  classification performance
##  eleven methods  shown  table
##
## performance  estimated using test accuracy acc
##
## micro f mic  macro f mac boldface
## denotes best results  eleven classification
## methods
##   improvement  often quite
## small   always  improvement demonstrated
##  observe   proposed method h w rf
##
##  computer journal vol
##
##
##
##
##
##
##
## baoxun xu joshua zhexue huang graham williams yunming ye
## fbis
##
##
##
## re
```

```
##
##
##
##
## log m
##
##
##
##
##
##
##
##
##
## cwrf
##
##
##
##
##
##
##
## hwrf
##
## c s
##
## c s
##
##
##
##
##
## cartwrf
##
```

```
## hwrf
## cwrf
##
##
##
## cartwrf
##
## chaidwrf
##
##
##
## chaidwrf
##
##
##
## hrf
##
## hrf
##
## log m
##
##
## crf
##
##
##
## crf
##
##
##
## cartrf
##
## cartrf
##
## chaidrf
##
```

```
##
##
## chaidrf
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
## number  features
##
##
##
##
```

```
##
##
##
##
##
##
##
##
## log m
##
##
##
##
##
##
##
## c s
##
## hwrf
##
##
##
##
##
##
##
##
##
##
## tr
##
##
##
##
```

```
##
## c s
##
##
##
## number  features
##
## re
##
##
##
##
##
## cwrf
##
##
##
##
##
## hwrf
## cwrf
##
##
##
## cartwrf
##
## cartwrf
##
## chaidwrf
##
##
##
## chaidwrf
##
##
##
```

```
## hrf
##
## hrf
##
## crf
##
## log m
##
##
##
##
## crf
##
##
##
## cartrf
##
## cartrf
##
## chaidrf
##
##
##
## chaidrf
##
##
##
##
##
##
##
##
##
##
##
```

```
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
## number  features
##
## number  features
##
## wap
```

```
##
## tr
##
##
##
##
##
##
## log m
##
## log m
##
##
##
##
##
##
##
## cwrf
##
## c s
##
##
##
## c s
##
## hwrf
##
##
##
##
##
##
##
```

```
##
##
## hwrf
##
##
##
## cwrf
## cartwrf
##
## cartwrf
##
##
## chaidwrf
##
##
##
## chaidwrf
## hrf
##
## hrf
## crf
##
##
##
## crf
##
##
##
## cartrf
##
## cartrf
## chaidrf
##
## chaidrf
##
##
```

```
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
```

```
##
##
##
##
##
##
##
## number  features
##
## number  features
##
## las
##
## las
##
##
##
##
##
##
##
##
##
##
## cwrf
## cartwrf
##
##
##
##
##
## hrf
##
##
##
```

```
## crf
## cartwrf
##
##
##
## chaidrf
##
##
##
##
##
##
##
##
##
## hwrf
##
##
##
## cwrf
## cartwrf
##
##
##
## chaidwrf
##
## log m
##
##
##
## c s
##
## c s
##
##
##
```

```
##
## hwrf
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
## chaidwrf
##
## log m
##
##
##
## hrf
## crf
##
##
## cartrf
## chaidrf
##
```

```
##
##
##
##
##
##
##
## number  features
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
## number  features
##
## figure  c s changes   number  features   subspace    high dimensional datasets
##
```

```
## outperformed   classification methods
## datasets
##
##
##
## conclusions
##
##   paper  presented  hybrid weighted random
## forest algorithm  simultaneously using  feature
## weighting method   hybrid forest method  classify
##  computer journal vol
##
##
##
##
##
## hybrid weighted random forests  classifying  highdimensional data
## fbis
##
##
##
##
##
## re
##
##
##
##
##
##
##
##
##
##
## hwrf
## cwrf
```

```
##
##
##
## cartwrf
##
## accuracy
##
## accuracy
##
##
##
## chaidwrf
## hrf
##
##
##
##
## hwrf
##
##
##
## cwrf
## cartwrf
##
##
##
## chaidwrf
## hrf
##
##
##
## crf
##
## crf
##
## log m
```

```
##
## cartrf
##
##
##
##
##
## cartrf
##
## log m
##
##
##
##
##
## chaidrf
##
## chaidrf
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
```

```
##
##
##
##
##
##
##
##
##
## number  features
##
##
##
##
##
##
##
##
##
##
##
##
## number  features
##
## re
##
## tr
##
##
##
##
##
```

```
##
##
##
## log m
##
##
##
##
##
##
##
## hwrf
## cwrf
##
##
##
## cartwrf
##
##
##
## accuracy
##
## accuracy
##
##
##
## chaidwrf
##
##
##
## hwrf
##
##
##
## cwrf
```

```
## cartwrf
##
##
##
## chaidwrf
## hrf
##
## hrf
##
##
##
## log m
##
##
##
## crf
##
## crf
##
##
##
## cartrf
##
## cartrf
##
##
##
##
## chaidrf
##
## chaidrf
##
##
##
##
```

```
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
```

```
## number  features
##
## number  features
##
## wap
##
##
##
##
##
## tr
##
##
##
##
##
##
##
##
##
##
##
##
## hwrf
## cwrf
##
##
##
## cartwrf
##
## accuracy
##
## accuracy
##
##
```

```
##
##
##
##
##
## hwrf
## cwrf
##
##
##
## cartwrf
## chaidwrf
##
## chaidwrf
##
##
##
## hrf
##
## log m
##
##
##
##
## hrf
##
## crf
##
##
##
## cartrf
##
##
##
## crf
##
```

```
## log m
##
##
##
##
## cartrf
## chaidrf
##
## chaidrf
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
```

```
##
##
##
##
##
##
##
##
##
##
##
## number  features
##
## number  features
##
## las
##
##
##
##
##
## las
##
##
##
##
##
##
##
## accuracy
##
##
## hwrf
## cwrf
```

```
##
##
##
## cartwrf
## chaidwrf
##
##
##
## accuracy
##
##
##
##
##
##
##
## hwrf
## cwrf
## cartwrf
##
##
##
## chaidwrf
## hrf
##
## hrf
##
## log m
##
##
##
## crf
##
##
##
## crf
```

```
##
## log m
##
##
##
##
## cartrf
##
## cartrf
## chaidrf
##
## chaidrf
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
```

```
##
##
##
##
##
##
##
##
##
## number  features
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
## number  features
```

```
##
## figure  test accuracy changes   number  features   subspace    high dimensional d
##
## high dimensional data  algorithm   retains
##  small subspace size breimans formula log m
##  determining  subspace size  create accurate
## random forest models  also effectively reduces
##  upper bound   generalization error
##
## improves classification performance   results
## experiments  various high dimensional datasets
## random forest generated   new method  superior
##   classification methods  can use  default
## log m    subspace size  generally guarantee
##
##  computer journal vol
##
##
##
##
##
##
## baoxun xu joshua zhexue huang graham williams yunming ye
##
## table   comparison  results
## datasets
## dataset
## fbis
## measures
## acc
## mic
## svm
##
## knn
##
```

```
##
## nb
##
## h rf
##
## c rf
##
## cart rf
##
## chaid rf
##
## h w rf
##
## c w rf
##
## cart w rf
##
## chaid w rf
##
## dataset
## wap
## measures
## acc
## mic
## svm
##
##
## knn
##
## nb
##
## h rf
##
## c rf
##
## cart rf
```

```
##
## chaid rf
##
## h w rf
##
## c w rf
##
## cart w rf
##
##
## chaid w rf
##
##
## best accuracy micro f  macro f results   eleven methods
## re
## mic
##
##
##
##
##
##
##
##
##
##
## tr
## mac
## acc
## mic
##
##
##
##
##
```

```
##
##
##
##
##
##
##
##
##
##
##
## mac
##
##
##
##
##
##
##
##
##
##
##
## acc
##
##
##
##
##
##
##
##
##
##
```

```
##
##
##
##  always produce  best models   variety
## measures  using  hybrid weighted random forest
## algorithm
## acknowledgements
##  research  supported  part  nsfc
## grant   shenzhen new industry development fund  grant nocxba
## references
##  breiman l  random forests machine learning
##
##  ho t  random subspace method  constructing decision forests ieee transactions  p
## analysis  machine intelligence
##  quinlan j  c programs  machine
## learning morgan kaufmann
##  breiman l  classification  regression trees
## chapman  hall crc
##  breiman l  bagging predictors
## machine
## learning
##  ho t  random decision forests proceedings
##   third international conference  document
## analysis  recognition pp  ieee
##  dietterich t   experimental comparison
## three methods  constructing ensembles  decision
## trees bagging boosting  randomization machine
## learning
##
## mac
##
##
##
##
##
##
```

```
##
##
##
##
##
## mac
##
##
##
##
##
##
##
##
##
##
## re
## mic
##
##
##
##
##
##
##
##
##
##
## las
## acc
## mic
##
##
```

```
##
##
##
##
##
##
##
##
##
##
## acc
##
##
##
##
##
##
##
##
##
##
## tr
## mic
##
##
##
##
##
##
##
##
##
```

```
##
##
## las
## mac
## acc
## mic
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
## mac
##
##
##
##
##
##
##
##
##
##
```

```
##
##
## acc
##
##
##
##
##
##
##
##
##
##
## mac
##
##
##
##
##
##
##
##
##
##
##
## mac
##
##
##
##
##
##
##
##
##
```

```
##
##
##
##
##  banfield r hall l bowyer k  kegelmeyer w
##   comparison  decision tree ensemble creation
## techniques ieee transactions  pattern analysis
##  machine intelligence
##
##  robniksikonja
## m  improving random forests
## proceedings   th european conference
## machine learning pp  springer
##  ho t  c decision forests proceedings
##  fourteenth international conference  pattern
## recognition pp  ieee
##  dietterrich t  machine learning research four
## current direction artificial intelligence magzine
##
##  amaratunga d cabrera j  lee y
## enriched random forests bioinformatics
##
##  ye y li h deng x  huang j
## feature weighting random forest  detection  hidden
## web search interfaces  journal  computational
## linguistics  chinese language processing
##
##  xu b huang j williams g wang q
## ye y  classifying  highdimensional data
##  random forests built  small subspaces
## international journal  data warehousing
## mining
##  xu b huang j williams g li j  ye y
##  hybrid random forests advantages  mixed
## trees  classifying text data proceedings   th
## pacificasia conference  knowledge discovery
```

```
## data mining springer
##
##  computer journal vol
##
##
##
##
##
## hybrid weighted random forests  classifying  highdimensional data
##  biggs d de ville b  suen e   method
##  choosing multiway partitions  classification
## decision trees journal  applied statistics
##  ture m kurt  turhan kurum   ozdamar
## k  comparing classification techniques
## predicting essential hypertension expert systems
## applications
##  begum n ma f  ren f  automatic text summarization using support vector machine
## international journal  innovative computing information  control
##  chen j huang h tian s  qu y
## feature selection  text classification  naive
## bayes expert systems  applications
##
##  tan s  neighborweighted knearest neighbor
##  unbalanced text corpus
## expert systems
## applications
##  pearson k    theory  contingency
##  relation  association  normal correlation
## cambridge university press
##  yang y  liu x   reexamination
## text categorization methods proceedings   th
## international conference  research  development
##  information retrieval pp  acm
##  han e  karypis g  centroidbased
## document classification analysis  experimental
## results proceedings   th european conference
```

```
## principles  data mining  knowledge discovery
## pp  springer
##  trec
##
## text
## retrieval
## conference
## http  trecnistgov
##  lewis
## d
##
## reuters
## text
## categorization
## test
## collection
## distribution
##
## http  wwwresearchattcom  lewis
##  han e boley d gini m gross r hastings
## k karypis g kumar v mobasher b
## moore j  webace  web agent  document
## categorization  exploration proceedings   nd
## international conference  autonomous agents pp
##  acm
##  mccallum   nigam k   comparison
## event models  naive bayes text classification aaai workshop  learning  text cate
##
##  witten  frank e  hall m  data mining
## practical machine learning tools  techniques
## morgan kaufmann
##
##  computer journal vol
##
##
```

Previously we used the English stopwords provided by . We could instead or in addition remove our own stop words as we have done above. We have chosen here two words, simply for illustration. The choice might depend on the domain of discourse, and might not become apparent until we've done some analysis.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

21.15 Strip Whitespace

```
docs <- tm_map(docs, stripWhitespace)
```

```
viewDocs(docs, 16)
```

```
## hybrid weighted random forests classifying highdimensional data baoxun xu joshua
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

21.16 Specific Transformations

We might also have some specific transformations we would like to perform. The examples here may or may not be useful, depending on how we want to analyse the documents. This is really for illustration using the part of the document we are looking at here, rather than suggesting this specific transform adds value.

```
toString <- content_transformer(function(x, from, to) gsub(from, to, x))
docs <- tm_map(docs, toString, "harbin institute technology", "HIT")
docs <- tm_map(docs, toString, "shenzhen institutes advanced technology", "SIAT")
docs <- tm_map(docs, toString, "chinese academy sciences", "CAS")
```

```
inspect(docs[16])
```

```
## <<SimpleCorpus>>
## Metadata:  corpus specific: 1, document level (indexed): 0
## Content:  documents: 1
##
##
## hybrid weighted random forests classifying highdimensional data baoxun xu joshua
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

21.17 Stemming

```
docs <- tm_map(docs, stemDocument)
```

```
viewDocs(docs, 16)
```

```
## hybrid weight random forest classifi highdimension data baoxun xu joshua zhexu hu
```

Stemming uses an algorithm that removes common word endings for English words, such as `es'', ` ed'' and 's''.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

21.18 Creating a Document Term Matrix

A document term matrix is simply a matrix with documents as the rows and terms as the columns and a count of the frequency of words as the cells of the matrix. We use tm::DocumentTermMatrix() to create the matrix:

```
dtm <- DocumentTermMatrix(docs)


dtm
```

```
## <<DocumentTermMatrix (documents: 46, terms: 6508)>>
## Non-/sparse entries: 30059/269309
## Sparsity           : 90%
## Maximal term length: 56
## Weighting          : term frequency (tf)
```

We can inspect the document term matrix using tm::inspect(). Here, to avoid too much output, we select a subset of inspect.

```
inspect(dtm[1:5, 1000:1005])
```

```
## <<DocumentTermMatrix (documents: 5, terms: 6)>>
## Non-/sparse entries: 9/21
## Sparsity           : 70%
## Maximal term length: 8
## Weighting          : term frequency (tf)
## Sample             :
##             Terms
## Docs         path patholog patient pbs per perhap
##   acnn96.txt    0        0       0   0   0      0
##   adm02.txt     0        0       0   0   0      0
##   ai02.txt      4        6       9   4   4      1
##   ai03.txt      0        0      11  11   0      0
##   ai97.txt      0        0       3   0   0      0
```

The document term matrix is in fact quite sparse (that is, mostly empty) and so it is actually stored in a much more compact representation internally. We can still get the row and column counts.

```
class(dtm)
```

```
## [1] "DocumentTermMatrix"    "simple_triplet_matrix"
```

```
dim(dtm)
```

```
## [1]   46 6508
```

The transpose is created using tm::TermDocumentMatrix():

```
tdm <- TermDocumentMatrix(docs)
tdm
```

```
## <<TermDocumentMatrix (terms: 6508, documents: 46)>>
## Non-/sparse entries: 30059/269309
## Sparsity           : 90%
## Maximal term length: 56
## Weighting          : term frequency (tf)
```

We will use the document term matrix for the remainder of the chapter.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

21.19 Exploring the Document Term Matrix

We can obtain the term frequencies as a vector by converting the document term matrix into a matrix and summing the column counts:

```
freq <- colSums(as.matrix(dtm))
length(freq)
```

```
## [1] 6508
```

By ordering the frequencies we can list the most frequent terms and the least frequent terms:

```
ord <- order(freq)

# Least frequent terms.
freq[head(ord)]
```

```
##      acnntex        dmitl  microsystem       ventur         adra attributeori
##            1            1            1            1            1            1
```

Notice these terms appear just once and are probably not really terms that are of interest to us. Indeed they are likely to be spurious terms introduced through the translation of the original document from PDF to text.

```
# Most frequent terms.
freq[tail(ord)]
```

```
##     can dataset pattern     use    mine    data
##     709     776     887    1366    1446    3101
```

These terms are much more likely to be of interest to us. Not surprising, given the choice of documents in the corpus, the most frequent terms are: .

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

21.20 Distribution of Term Frequencies

```
# Frequency of frequencies.
head(table(freq), 15)
```

```
## freq
##    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15
## 2381 1031  504  308  209  190  135  130   81   82   65   62   55   52   52
```

```
tail(table(freq), 15)
```

```
## freq
##  483  544  547  555  578  609  611  616  703  709  776  887 1366 1446 3101
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
```

So we can see here that there are terms that occur just once.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

21.21 Conversion to Matrix and Save to CSV

We can convert the document term matrix to a simple matrix for writing to a CSV file, for example, for loading the data into other software if we need to do so. To write to CSV we first convert the data structure into a simple matrix:

```
m <- as.matrix(dtm)
dim(m)
```

```
## [1]   46 6508
```

For very large corpus the size of the matrix can exceed R's calculation limits. This will manifest itself as a integer overflow error with a message like:

If this occurs, then consider removing sparse terms from the document term matrix, as we discuss shortly.

Once converted into a standard matrix the usual utils::write.csv() can be used to write the data to file.

```
write.csv(m, file="dtm.csv")
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

21.22 Removing Sparse Terms

We are often not interested in infrequent terms in our documents. Such sparse'' terms can be removed from the document term matrix quite easily using tm::removeSparseTerms():

```
dim(dtm)
```

```
## [1]   46 6508
```

```
dtms <- removeSparseTerms(dtm, 0.1)
dim(dtms)
```

```
## [1] 46  6
```

This has removed most terms!

```
inspect(dtms)
```

```
## <<DocumentTermMatrix (documents: 46, terms: 6)>>
## Non-/sparse entries: 257/19
## Sparsity           : 7%
## Maximal term length: 7
## Weighting          : term frequency (tf)
## Sample             :
##                                Terms
## Docs                            data graham inform time use william
##   adm02.txt                      189      1     12   45  31       1
##   his03.txt                      176     17      1    7   7      25
##   ijdwm2012.txt                  110      2     65    4  65       2
##   karisk.txt                      83      1     24    6  55       3
##   kddmodel.txt                   127      1     23    1  44       3
##   pakdd07.txt                    161      1      6    2  26       3
##   RJournal_2009-2_Williams.txt  122      2      8    9  62      10
##   seal98.txt                     171      2     18   13  31       2
##   story.txt                      205     11      4   18  38      35
##   templates.txt                   67      1     55   25  34       6
```

We can see the effect by looking at the terms we have left:

```
freq <- colSums(as.matrix(dtms))
freq
```

```
##    data  graham  inform    time     use william
##    3101     108     467     483    1366     236
```

```
table(freq)
```

```
## freq
##  108  236  467  483 1366 3101
##    1    1    1    1    1    1
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

21.23 Identifying Frequent Items and Associations

One thing we often to first do is to get an idea of the most frequent terms in the corpus. We use tm::findFreqTerms() to do this. Here we limit the output to those terms that occur at least 1,000 times:

```
findFreqTerms(dtm, lowfreq=1000)
```

```
## [1] "data" "mine" "use"
```

So that only lists a few. We can get more of them by reducing the threshold:

```
findFreqTerms(dtm, lowfreq=100)
```

```
##   [1] "accuraci"    "acsi"        "advers"      "algorithm"   "also"
##   [6] "approach"    "associ"      "australia"   "australian"  "build"
##  [11] "call"        "classif"     "common"      "comput"      "consist"
##  [16] "csiro"       "data"        "databas"     "dataset"     "develop"
##  [21] "discoveri"   "effici"      "explor"      "graham"      "howev"
##  [26] "implement"   "import"      "inform"      "kdd"         "knowledg"
##  [31] "larg"        "may"         "mine"        "network"     "neural"
##  [36] "page"        "particular"  "perform"     "problem"     "provid"
##  [41] "requir"      "research"    "rule"        "scienc"      "search"
##  [46] "significf"   "structur"    "system"      "techniqu"    "technolog"
##  [51] "time"        "tool"        "train"       "univers"     "use"
##  [56] "weight"      "william"     "within"      "work"        "allow"
##  [61] "analysi"     "appli"       "applic"      "area"        "attribut"
##  [66] "base"        "can"         "case"        "chang"       "class"
##  [71] "classifi"    "cluster"     "collect"     "combin"      "compar"
##  [76] "condit"      "confer"      "contain"     "decis"       "defin"
##  [81] "describ"     "differ"      "discov"      "discuss"     "distanc"
##  [86] "distribut"   "domain"      "estim"       "event"       "exampl"
##  [91] "exist"       "featur"      "find"        "first"       "follow"
##  [96] "form"        "function"    "general"     "generat"     "given"
## [101] "group"       "hybrid"      "identifi"    "includ"      "increas"
## [106] "interest"    "intern"      "interv"      "lead"        "learn"
## [111] "level"       "like"        "link"        "make"        "mani"
## [116] "mean"        "measur"      "method"      "model"       "multipl"
## [121] "need"        "new"         "number"      "object"      "observ"
## [126] "occur"       "often"       "one"         "paper"       "pattern"
## [131] "period"      "point"       "predict"     "present"     "proceed"
## [136] "process"     "propos"      "record"      "refer"       "regress"
## [141] "relat"       "report"      "repres"      "result"      "sampl"
## [146] "section"     "select"      "sequenc"     "set"         "similar"
## [151] "singl"       "stage"       "state"       "statist"     "step"
## [156] "studi"       "subset"      "support"     "target"      "task"
## [161] "tempor"      "three"       "transact"    "tree"        "two"
## [166] "type"        "understand"  "user"        "valu"        "variabl"
## [171] "well"        "will"        "year"        "age"         "avail"
```

```
## [176] "averag"      "care"        "claim"       "consid"      "cost"
## [181] "day"         "detect"      "drug"        "effect"      "episod"
## [186] "error"       "expect"      "expert"      "fig"         "figur"
## [191] "health"      "hospit"      "http"        "indic"       "individu"
## [196] "intellig"    "journal"     "machin"      "medic"       "node"
## [201] "packag"      "patient"     "popul"       "servic"      "show"
## [206] "small"       "sourc"       "tabl"        "test"        "total"
## [211] "unexpect"    "angioedema"  "current"     "evalu"       "high"
## [216] "interesting" "order"       "ratio"       "reaction"    "risk"
## [221] "unit"        "usual"       "visual"      "entiti"      "experi"
## [226] "hot"         "insur"       "map"         "nugget"      "open"
## [231] "polici"      "size"        "spot"        "random"      "vector"
## [236] "outlier"     "pmml"        "rank"        "rnn"         "window"
## [241] "adr"         "oper"        "forest"      "subspac"     "rattl"
## [246] "utar"
```

We can also find associations with a word, specifying a correlation limit.

```r
findAssocs(dtm, "data", corlimit=0.6)
```

```
## $data
##         mine       induct     challeng         know       answer         need
##         0.90         0.72         0.70         0.65         0.64         0.63
## statistician      general      foundat        major         mani        boost
##         0.63         0.62         0.62         0.61         0.61         0.61
##         come
##         0.60
```

If two words always appear together then the correlation would be 1.0 and if they never appear together the correlation would be 0.0. Thus the correlation is a measure of how closely associated the words are in the corpus.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

21.24 Correlations Plots

```
plot(dtm,
     terms=findFreqTerms(dtm, lowfreq=100)[1:50],
     corThreshold=0.5)
```

(Hansen et al. 2018) from the BioConductor repository for R (`bioconductor.org`) is used to plot the network graph that displays the correlation between chosen words in the corpus. Here we choose 50 of the more frequent words as the nodes and include links between words when they have at least a correlation of 0.5.

By default (without providing terms and a correlation threshold) the plot function chooses a random 20 terms with a threshold of 0.7.

References

Hansen, Kasper Daniel, Jeff Gentry, Li Long, Robert Gentleman, Seth Falcon, Florian Hahne, and Deepayan Sarkar. 2018. *Rgraphviz: Provides Plotting Capabilities for r Graph Objects*.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

21.25 Correlations Plot—Options

```
plot(dtm,
     terms=findFreqTerms(dtm, lowfreq=100)[1:50],
     corThreshold=0.5)
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

21.26 Plotting Word Frequencies

We can generate the frequency count of all words in a corpus:

```
freq <- sort(colSums(as.matrix(dtm)), decreasing=TRUE)
head(freq, 14)
```

```
##      data      mine       use   pattern   dataset       can     model   cluster
##      3101      1446      1366       887       776       709       703       616
## algorithm      rule    featur       set      tree    method
##       611       609       578       555       547       544
```

```
wf   <- data.frame(word=names(freq), freq=freq)
head(wf)
```

```
##            word freq
## data       data 3101
## mine       mine 1446
## use         use 1366
## pattern pattern  887
## dataset dataset  776
## can         can  709
```

We can then plot the frequency of those words that occur at least 500 times in the corpus:

```
subset(wf, freq>500)                                                    %>%
  ggplot(aes(word, freq))                                                 +
  geom_bar(stat="identity")                                               +
  theme(axis.text.x=element_text(angle=45, hjust=1))
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

21.27 Word Clouds

We can generate a word cloud as an effective alternative to providing a quick visual overview of the frequency of words in a corpus.

The (Fellows 2018) package provides the required function.

```
set.seed(123)
wordcloud(names(freq), freq, min.freq=40)
```

Notice the use of base::set.seed() only so that we can obtain the same layout each time—otherwise a random layout is chosen, which is not usually an issue.

References

Fellows, Ian. 2018. *Wordcloud: Word Clouds*. https://CRAN.R-project.org/package=wordcloud.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

21.28 Reducing Clutter With Max Words

To increase or reduce the number of words displayed we can tune the value of *max.words*=. Here we have limited the display to the 100 most frequent words.

```
set.seed(142)
wordcloud(names(freq), freq, max.words=100)
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

21.29 Reducing Clutter With Min Freq

A more common approach to increase or reduce the number of words displayed is by tuning the value of *min.freq*=. Here we have limited the display to those words that occur at least 100 times.

```
set.seed(142)
wordcloud(names(freq), freq, min.freq=100)
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

21.30 Adding Some Colour

We can also add some colour to the display. Here we make use of RColorBrewer::brewer.pal() from (Neuwirth 2014) to generate a palette of colours to use.

```
set.seed(142)
wordcloud(names(freq), freq, min.freq=100, colors=brewer.pal(6, "Dark2"))
```

References

Neuwirth, Erich. 2014. *RColorBrewer: ColorBrewer Palettes*. https://CRAN.R-project.org/package=RColorBrewer.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

21.31 Varying the Scaling

We can change the range of font sizes used in the plot using the *scale*= option. By default the most frequent words have a scale of 4 and the least have a scale of 0.5. Here we illustrate the effect of increasing the scale range.

```
set.seed(142)
wordcloud(names(freq), freq, min.freq=100, scale=c(5, .1), colors=brewer.pal(6, "Da
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

21.32 Rotating Words

We can change the proportion of words that are rotated by 90 degrees from the default 10% to, say, 20% using *rot.per=0.2*.

```
set.seed(142)
dark2 <- brewer.pal(6, "Dark2")
wordcloud(names(freq), freq, min.freq=100, rot.per=0.2, colors=dark2)
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

21.33 Quantitative Analysis of Text

The (Rinker 2020a) package provides an extensive suite of functions to support the quantitative analysis of text.

We can obtain simple summaries of a list of words, and to do so we will illustrate with the terms from our Term Document Matrix `tdm`. We first extract the shorter terms from each of our documents into one long word list. To do so we convert `tdm` into a matrix, extract the column names (the terms) and retain those shorter than 20 characters.

```
words <- dtm                                                    %>%
  as.matrix                                                     %>%
  colnames                                                      %>%
  (function(x) x[nchar(x) < 20])
```

We can then summarise the word list. Notice, in particular, the use of qdap::dist_tab() from to generate frequencies and percentages.

```
length(words)
```

```
## [1] 6456
```

```
head(words, 15)
```

```
##  [1] "abstract"  "academi"   "accur"     "accuraci"  "acnntex"   "acsi"
##  [7] "act"       "address"   "adjust"    "adopt"     "advanc"    "advantag"
## [13] "advers"    "affect"    "algorithm"
```

```
summary(nchar(words))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   3.000   5.000   6.000   6.644   8.000  19.000
```

```r
table(nchar(words))
```

```
##
##    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17   18
##  579  867 1044 1114  935  651  397  268  200  138   79   63   34   28   22   21
##   19
##   16
```

```r
dist_tab(nchar(words))
```

```
##    interval freq cum.freq percent cum.percent
## 1         3  579      579    8.97        8.97
## 2         4  867     1446   13.43       22.40
## 3         5 1044     2490   16.17       38.57
## 4         6 1114     3604   17.26       55.82
## 5         7  935     4539   14.48       70.31
## 6         8  651     5190   10.08       80.39
## 7         9  397     5587    6.15       86.54
## 8        10  268     5855    4.15       90.69
## 9        11  200     6055    3.10       93.79
## 10       12  138     6193    2.14       95.93
## 11       13   79     6272    1.22       97.15
## 12       14   63     6335    0.98       98.13
## 13       15   34     6369    0.53       98.65
## 14       16   28     6397    0.43       99.09
## 15       17   22     6419    0.34       99.43
## 16       18   21     6440    0.33       99.75
## 17       19   16     6456    0.25      100.00
```

References

———. 2020a. *Qdap: Bridging the Gap Between Qualitative Data and Quantitative Analysis*. http://trinker.github.io/qdap/.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

21.34 Word Length Counts

A simple plot is then effective in showing the distribution of the word lengths. Here we create a single column data frame that is passed on to ggplot2::ggplot() to generate a histogram, with a vertical line to show the mean length of words.

```
data.frame(nletters=nchar(words))                                    %>%
  ggplot(aes(x=nletters))                                             +
  geom_histogram(binwidth=1)                                          +
  geom_vline(xintercept=mean(nchar(words)),
             colour="green", size=1, alpha=.5)                        +
  labs(x="Number of Letters", y="Number of Words")
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of*

Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

21.35 Letter Frequency

Next we want to review the frequency of letters across all of the words in the discourse. Some data preparation will transform the vector of words into a list of letters, which we then construct a frequency count for, and pass this on to be plotted.

We again use a pipeline to string together the operations on the data. Starting from the vector of words stored in `word` we split the words into characters using stringr::str_split() from (Wickham 2019b), removing the first string (an empty string) from each of the results (using BiocGenerics::sapply()). Reducing the result into a simple vector, using BiocGenerics::unlist(), we then generate a data frame recording the letter frequencies, using qdap::dist_tab() from . We can then plot the letter proportions.

References

———. 2019b. *Stringr: Simple, Consistent Wrappers for Common String Operations*. https://CRAN.R-project.org/package=stringr.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

21.36 Letter and Position Heatmap

```
## Warning: Ignoring unknown aesthetics: fill
```

The qdap::qheat() function from provides an effective visualisation of tabular data. Here we transform the list of words into a position count of each letter, and constructing a table of the proportions that is passed on to qdap::qheat() to do the plotting.

```
words                                                                  %>%
  lapply(function(x) sapply(letters, gregexpr, x, fixed=TRUE))         %>%
  unlist                                                               %>%
  (function(x) x[x!=-1])                                               %>%
  (function(x) setNames(x, gsub("\\d", "", names(x))))                 %>%
  (function(x) apply(table(data.frame(letter=toupper(names(x)),
                                      position=unname(x))),
                     1, function(y) y/length(x)))                      %>%
  qheat(high="green", low="yellow", by.column=NULL,
        values=TRUE, digits=3, plot=FALSE)                              +
  labs(y="Letter", x="Position") +
  theme(axis.text.x=element_text(angle=0))                              +
  guides(fill=guide_legend(title="Proportion"))
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

21.37 Miscellaneous Functions

We can generate gender from a name list, using the (**R-genderdata?**) package

```
devtools::install_github("lmullen/gender-data-pkg")
```

```
name2sex(qcv(graham, frank, leslie, james, jacqui, jack, kerry, kerrie))
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

21.38 Word Distances

Continuous bag of words (CBOW). Word2Vec associates each word in a vocabulary with a unique vector of real numbers of length d. Words that have a similar syntactic context appear closer together within the vector space. The syntactic context is based on a set of words within a specific window size.

```
install.packages("tmcn.word2vec", repos="http://R-Forge.R-project.org")
library(tmcn.word2vec)
model <- word2vec(system.file("examples", "rfaq.txt", package = "tmcn.word2vec"))
distance(model$model_file, "the")
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

21.39 Review Preparing the Corpus

Here in one sequence is collected the code to perform a text mining project. Notice that we would not necessarily do all of these steps so pick and choose as is appropriate to your situation.

```
# Locate and load the Corpus.

cname <- file.path(".", "corpus", "txt")
docs <- Corpus(DirSource(cname))

docs
summary(docs)
inspect(docs[1])

# Transforms

toSpace <- content_transformer(function(x, pattern) gsub(pattern, " ", x))
docs <- tm_map(docs, toSpace, "/|@|\\|")

docs <- tm_map(docs, content_transformer(tolower))
docs <- tm_map(docs, removeNumbers)
docs <- tm_map(docs, removePunctuation)
docs <- tm_map(docs, removeWords, stopwords("english"))
docs <- tm_map(docs, removeWords, c("own", "stop", "words"))
docs <- tm_map(docs, stripWhitespace)

toString <- content_transformer(function(x, from, to) gsub(from, to, x))
docs <- tm_map(docs, toString, "specific transform", "ST")
docs <- tm_map(docs, toString, "other specific transform", "OST")

docs <- tm_map(docs, stemDocument)
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

21.40 Review Analysing the Corpus

```
# Document term matrix.

dtm <- DocumentTermMatrix(docs)
inspect(dtm[1:5, 1000:1005])

# Explore the corpus.

findFreqTerms(dtm, lowfreq=100)
findAssocs(dtm, "data", corlimit=0.6)

freq <- sort(colSums(as.matrix(dtm)), decreasing=TRUE)
wf   <- data.frame(word=names(freq), freq=freq)

p <- ggplot(subset(wf, freq>500), aes(word, freq))
p <- p + geom_bar(stat="identity")
p <- p + theme(axis.text.x=element_text(angle=45, hjust=1))

# Generate a word cloud

wordcloud(names(freq), freq, min.freq=100, colors=brewer.pal(6, "Dark2"))
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

21.41 LDA

Topic Models such as Latent Dirichlet Allocation has been popular for text mining in last 15 years. Applied with varying degrees of success. Text is fed into LDA to extract the topics underlying the text document. Examples are the AP corpus and the Science Corpus 1880-2002 (Blei and Lafferty 2009). PERHAPS USEFUL IN BOOK?

When is LDA applicable - it will fail on some data and need to choose number of topics to find and how many documents are needed. HOw do we know the topics learned are correct topics.

Two fundemental papers - independelty discovered: Blei, Ng, Jordan - NIPS 2001 with 11k citations. Other paper is Pritchard, Stephens, and Donnelly in Genetics June 200 14K citations - models are exactly the same except for minor differences: except topics versus population structures.

No theoretic analysis as such. How to guarantee correct topics and how efficient is the learning procedure?

Observations:

LDA won't work on many short tweets or very few long documents.

We should not liberally over-fit the LDA with too many redundant topics…

Limiting factors:

We should use as many documents as we can and short documents less than 10 words won't work even if there are many of them. Need sufficiently long documents.

Small Dirichlet paramenter helps especially if we overfit. See Long Nguen's keynote at PAKDD 2015 in Vietnam.

number of documents the most important factor

document length plays a useful role too

avoid overfitting as you get too many topics and don't really learn anything as the humn needs to cull the topics.

New work detects new topics as they emerge.

```r
data(cora.documents)
data(cora.vocab)
theme_set(theme_bw())
set.seed(8675309)
K <- 10 ## Num clusters
result <- lda.collapsed.gibbs.sampler(cora.documents,
                                      K,  ## Num clusters
                                      cora.vocab,
                                      25,  ## Num iterations
                                      0.1,
                                      0.1,
                                      compute.log.likelihood=TRUE)
## Get the top words in the cluster
top.words <- top.topic.words(result$topics, 5, by.score=TRUE)
## Number of documents to display
N <- 10

topic.proportions <- t(result$document_sums) / colSums(result$document_sums)

topic.proportions <-
   topic.proportions[sample(1:dim(topic.proportions)[1], N),]

topic.proportions[is.na(topic.proportions)] <-  1 / K

colnames(topic.proportions) <- apply(top.words, 2, paste, collapse=" ")

topic.proportions.df <- melt(cbind(data.frame(topic.proportions),
                                   document=factor(1:N)),
                             variable.name="topic",
                             id.vars = "document")

ggplot(topic.proportions.df, aes(x=topic, y=value, fill=topic)) +
    geom_bar(stat="identity") +
    theme(axis.text.x = element_text(angle=45, hjust=1, size=7),
          legend.position="none") +
```

```
    coord_flip() +
    facet_wrap(~ document, ncol=5)
```

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

21.42 Further Reading and Acknowledgements

Other resources include:

- The Journal of Statistical Software article, is a good start http://www.jstatsoft.org/v25/i05/paper
- Bilisoly (2008) presents methods and algorithms for text mining using Perl.

Thanks also to Tony Nolan for suggestions of some of the examples used in this chapter.

Some of the examples were motivated by http://trinkerrstuff.wordpress.com/2014/10/31/exploration-of-letter-make-up-of-english-words/.

References

Bilisoly, Roger. 2008. *Practical Text Mining with Perl*. Wiley Series on Methods and Applications in Data Mining. Wiley. http://books.google.com.au/books?id=YkMFVbsrdzkC.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

22 Computer Vision

20210605 What does it mean to actually see something? Since at least the 1980's we have been getting our computers more competent at collecting an image through a digitally connected camera and processing the image in some way. It has been pretty straight forward to identify colour in images and generally identify boundaries within the image. Through complex algorithmic approaches we were able to basically identify objects within the image, identifying, chairs, and balls and so in.

With the advent of deep neural networks, that is networks of neurons with very many layers and nodes, we've replaced the algorithmic approaches (which us humans writing the algorithms) with automated feature learning and algorithm generating approaches that we little understand. Yet, building these models from massive collections of data utilising considerable amounts of compute, the models can now often exhibit better than human performance.

In the future we might also hope that we will discover how this is achieved and so gain actual knowledgeable insights into intelligence.

In this chapter we explore the current state-of-the-art in computer vision with the aim to explain it all simply. We will often highlight how you can explore the technology yourself through the MLHub repository. Reference will frequently be made to the companion sections of the Machine Learning Hub Desktop Survival Guide sections on Computer Vision. Lear about computer vision by doing it, and find a few useful tools that might actually become part of your daily toolkit.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

22.1 Resnet Models

THIS SECTION IS UNDER DEVELOPMENT. PLEASE COME BACK LATER

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

22.2 Document Classification

THIS SECTION IS UNDER DEVELOPMENT. PLEASE COME BACK LATER

A useful training dataset or demo data can be found from https://www.cs.cmu.edu/~aharley/rvl-cdip/

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

22.3 Face Recognition

THIS SECTION IS UNDER DEVELOPMENT. PLEASE COME BACK LATER

https://www.wired.com/story/schools-adopt-face-recognition-name-fighting-covid/

The Azure Face cloud service, through the MLHub repository, supports facial detection and also can identify similar faces.

The MLHub repository provides the *facematch* package which demonstrates the open source capabilities of facial matching.

Azure's Computer Vision cloud service, through the MLHub repository, also supports the identification of faces and of celebrities.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

22.4 OCR Optical Character Recognition

THIS SECTION IS UNDER DEVELOPMENT. PLEASE COME BACK LATER

A useful training dataset or demo data can be found from https://www.cs.cmu.edu/~aharley/rvl-cdip/

Azure's Computer Vision cloud service, through the MLHub repository, offers ocr as does the open source *easyocr* ocr command, though the latter tends to not as good as the former.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

23 Natural Language

20210605

BERT = Bidirectional Encoder Representation from Transformations.

GPT-N = Generative Pre-trained Transformer.

DistilBART = Distilled Bidirectional and Auto-Regressive Transformed.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

23.1 Natural Language Processing

THIS SECTION IS UNDER DEVELOPMENT. PLEASE COME BACK LATER

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

23.2 Natural Language Understanding

THIS SECTION IS UNDER DEVELOPMENT. PLEASE COME BACK LATER

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

24 Planning

20210610

Planning is to generate a course of action, enabling an agent to reach given goals. We have states of the world, actions to transist states, and a set of goals. A Plan is action sequence leading to a goal state. Policy maps current state to action to execute. It is a prolem because the space is large.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

25 Privacy

20200514 **Privacy is a fundamental human right** according to the Australian Government's Office of the Australian Information Commissioner. Microsoft's CEO Satya Nadella and President Brad Smith have also stated the same principle on many occasions.

Without privacy we live in a state of fear, fear that anything we say may be overheard, or anything we do may lead to retribution or other inappropriate consequences. It severely constrains us, even though we do nothing wrong or illegal.

Privacy is thus essential to our freedom.

Unfortunately, Governments and Business in reality see it differently.

Citizens need to take care of their own privacy, and today with some care our technology can help us.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

25.1 Cloud Privacy

20210507 In general, unless you ensure your data is encrypted in the cloud, your data is accessible by the cloud host, government, and criminal entities. We should store our data encrypted at all times. Unfortunately, there appear to be legislative attempts to access our data at the point where it is not encrypted (when we use or view the data) which can then also be compromised by criminal entities. Be vigilant.

For a while US cloud compute companies fought in the US courts against action by the US government to access data held within data centres overseas. Whilst this was underway the US government introduced the CLOUD Act (2018) to clarify that the US has the right to request this data where ever it is. The cloud compute companies accepted this, despite the risk it posed to their business model.

> *The Clarifying Lawful Overseas Use of Data Act or CLOUD Act (H.R. 4943) is a United States federal law enacted in 2018 by the passing of the Consolidated Appropriations Act, 2018, PL 115-141, Division V. The CLOUD Act primarily amends the Stored Communications Act (SCA) of 1986 to allow federal law enforcement to* ***compel U.S.-based technology companies via warrant or subpoena to provide requested data stored on servers regardless of whether the data are stored in the U.S. or on foreign soil.*** *(Wikipedia 20210507)*

Ensuring that all of your data in the cloud is encrypted is important.

However, this is not enough. At some stage you need to decrypt your data to make use of it. This is where the Australian Government stepped in to force technology companies to hand over data once it is decrypted by the user. This came into effect in 2018 with the Assistance and Access Act 2018. Apparently, the Government can demand a tech company include a mechanism for access to decrypted data in their otherwise encrypted apps, in secret. Whilst the encryption itself is not interfered with, once decrypted on the user's device, a backdoor can provide access to the otherwise private data. Today, any number of apps in Australia may have such a backdoor and we may be unaware of it. It is also likely that if there are backdoors then they will be exploitable, eventually, by criminals too. The Act explicitly prohibits a backdoor to the encrypted data itself, but not, apparently, a backdoor to access the data once it is decrypted.

> *Under the bill, designated communications providers (which include … **anyone who develops software** likely to be used by a carriage service or an electronic service with end-users in Australia …) can be **ordered to assist in intercepting information** relevant to a case, either by means of an existing capability if possible (Technical Assistance Order, TAO), or being ordered to develop, test, add, or remove equipment for a new interception capability (Technical Capability Order, TCO). (Wikipedia, 20210507)*

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

25.2 Computer Vision Privacy

Face recognition is becoming widely used in government and business for a variety of tasks including border control, crowd control, shopper identification, and so on. A significant issue for such models is to deal with the bias of the models often resulting from the bias inherent in the dataset of images provided for the training of the model.

The US National Institute of Standards and Techology (NIST) has considered issues of bias and effectiveness of face recognition models. For example, during the COVID-19 pandemic NIST investigated the impact of masks on identification but automatically masking images of millions of faces and testing the effectiveness.

See https://www.nist.gov/programs-projects/face-recognition-vendor-test-frvt

https://www.wired.com/story/schools-adopt-face-recognition-name-fighting-covid/

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

25.3 Differential Privacy

Differential privacy developed in 2006 by Cynthia Dwork is now widely used across industry so that data can be shared with limited impact on privacy.

The concept is to share aggregate data without disclosing individual data, by statistically perturbing the data in such a way so as not to lose the anlaytical value of the data.

Consider if we shared observations of 100 people (e.g., outcomes of a medical test), and in aggregate note that 70 test positive and 30 test negative. Later we might find the test results for the other 99 patients. Then we can determine the outcome for the remaining patient based on the supplied aggregate data! This is referred to as a *differential attack* and is difficult to protect against as we do not in general know how much other data may become available.

Differential privacy adds random noise to the aggregated data so instead of 70 patients testing positive it might be 68 or 69 or 71 or 72 that are reported as positive. It is roughly 70% and so any analysis remains generally valid whilst not being precise. The random perturbation introduced provides a level of privacy. Even knowing the test outcomes for 99 patients, the final patients test outcome can not be ascertained.

Mathematically it can be determined for a dataset how much noise is enough for specific requirements of differential privacy.

Applications include location-based services, social networks, and recommendation systems, with a body of research focused on differential privacy in graphs.

Machine learning in this context will train on aggregated dataset and obfuscated using differential privacy noise. Or trained on centrally-located data obfuscated using LDP noise. Performance degradation is $O(log(n)n^-2epsilon^-2)$ and $Omega(n^-2epsilon^-2)$ (Bassil et al 2014) Epsilon is the privacy budget. Larger datasets can deliver better privacy. Distributed ML with DP asks each contributor to add DP noice to the gradient cacluations.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of*

Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

26 Literate Data Science

20200602 A data scientist's role is to tell the stories supported by the data.

The narrative that we tell is one of our key deliverables and as such we need our narrative to be well supported by the data. In telling the narrative the analysis needs to be transparent, repeatable, and reproducible. We also capture and share our activities for quality assurance and for peer review. We will find ourselves repeating our work on other datasets in other scenarios and with other organisations. Documenting what we do helps when we come back to the code at a later time. Others will also want to reproduce our work and we should do all we can to facilitate that process. In short, we need to clearly communicate what we do so that we and others can understand and can continue the journey.

A general rule of thumb tells us that we should spend about a quarter of our time capturing what we have done—documenting our projects. Even more important is to capture this as we are doing the work rather than the chore of writing it up later. This does present an overhead and risks interrupting the flow of our work but the investment pays off longer term. Tools can be utilised to support the capture of our work with minimal interruption to our work flow.

To support the narrative and to encourage our efforts to be transparent, repeatable and reproducible we introduce the concept of literate programming (Knuth 1984). The concept is to intermix our narrative with the underlying analyses of the data (our code) within the one document. By introducing the concept here we aim to provide a solid foundation for the data scientist. We won't always have the time or the patience to deliver a carefully crafted narrative telling the story derived from the data but we should strive to do so.

We will use knitr (Xie 2021) to support literate data science. This package combines the document typesetting power of the free and open source LaTeX software with the statistical power of R. Literate data science is also well supported by RStudio which is able to process the source document into a beautifully formatted PDF. This book is itself produced using knitr (Xie 2021).

In addition to these packages we also need to install the LaTeX software. LaTeX is a typesetting markup language which combined with knitr (Xie 2021) allows us to intermix R code with our narrative and to program certain parts of the narrative using R. LaTeX is free and open source

software and instructions for installing are available from the LaTeX Project.

See the GNU/Linux Desktop Survival Guide for using markdown instead of LaTeX.

References

Knuth, Donald E. 1984. “Literate Programming.” *The Computer Journal (British Computer Society)* 27 (2): 97–111. http://www.literateprogramming.com/knuthweb.pdf.

———. 2021. *Knitr: A General-Purpose Package for Dynamic Report Generation in r*. https://yihui.org/knitr/.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*

Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

26.1 KnitR Setup

20200602

Packages used in this chapter include diagram (Soetaert 2020), dplyr (Wickham et al. 2021), ggplot2 (Wickham et al. 2020), magrittr (Bache and Wickham 2020), xtable (Dahl et al. 2019), Hmisc (Harrell 2021), and rattle (G. Williams 2020).

Packages are loaded into the currently running R session from your local library directories on disk. Missing packages can be installed using utils::install.packages() within R. On Ubuntu, for example, R packages can also be installed using `$ wajig install r-cran-<pkgname>`.

```
# Load required packages from local library into the R session.

library(rattle)       # Dataset: weatherAUS.
library(magrittr)     # Data pipelines: %>% %T>% %<>%.
library(ggplot2)      # Visualise data.
library(xtable)       # Format R data frames as LaTeX tables.
library(Hmisc)        # Escape special LaTeX charaters.
library(diagram)      # Produce a flowchart.
library(dplyr)        # Data wrangling.
library(scales)
library(knitr)
```

The rattle::weatherAUS dataset is loaded into the template variable `ds` and further template variables are setup as introduced by Graham J. Williams (2017). See Chapter 8 for details.

```r
dsname <- "weatherAUS"
ds     <- get(dsname)

nobs   <- nrow(ds)

vnames <- names(ds)
ds    %<>% clean_names(numerals="right")
names(vnames) <- names(ds)

vars   <- names(ds)
target <- "rain_tomorrow"
vars   <- c(target, vars) %>% unique() %>% rev()
```

A random sample of the dataset:

```r
ds %>% sample_frac()
```

```
## # A tibble: 176,747 x 24
##    date       location    min_temp max_temp rainfall evaporation sunshine
##    <date>     <chr>          <dbl>    <dbl>    <dbl>       <dbl>    <dbl>
##  1 2020-02-01 PearceRAAF      13.3     30.1      0          NA       11.8
##  2 2009-07-06 Hobart           5.1      8.6      2           0.6      4.5
##  3 2018-11-26 Perth            9.9     23.2      0           5.6     13.1
##  4 2009-03-22 Brisbane        19.3     28.6      0           6       10.8
##  5 2019-06-26 Mildura          2.8     19.8      0           0.5      8.3
##  6 2010-12-20 Tuggeranong      7       17.6      5.8        NA       NA
##  7 2014-10-16 NorahHead       10.4     21.4      1          NA       NA
##  8 2015-07-16 Portland         8       11.2     11.4         0.6      4.7
##  9 2008-07-21 Darwin          21.4     32.7      0           5.4     10.9
## 10 2013-04-01 Mildura         10.4     22.5      0           5        9.1
## # … with 176,737 more rows, and 17 more variables: wind_gust_dir <ord>,
## #   wind_gust_speed <dbl>, wind_dir_9am <ord>, wind_dir_3pm <ord>,
## #   wind_speed_9am <dbl>, wind_speed_3pm <dbl>, humidity_9am <int>,
## #   humidity_3pm <int>, pressure_9am <dbl>, pressure_3pm <dbl>,
## #   cloud_9am <int>, cloud_3pm <int>, temp_9am <dbl>, temp_3pm <dbl>,
## #   rain_today <fct>, risk_mm <dbl>, rain_tomorrow <fct>
```

References

Bache, Stefan Milton, and Hadley Wickham. 2020. *Magrittr: A Forward-Pipe Operator for r*. https://CRAN.R-project.org/package=magrittr.

Dahl, David B., David Scott, Charles Roosen, Arni Magnusson, and Jonathan Swinton. 2019. *Xtable: Export Tables to LaTeX or HTML*. http://xtable.r-forge.r-project.org/.

Harrell, Frank E, Jr. 2021. *Hmisc: Harrell Miscellaneous*. https://CRAN.R-project.org/package=Hmisc.

Soetaert, Karline. 2020. *Diagram: Functions for Visualising Simple Graphs (Networks), Plotting Flow Diagrams*. https://CRAN.R-project.org/package=diagram.

Wickham, Hadley, Winston Chang, Lionel Henry, Thomas Lin Pedersen, Kohske Takahashi, Claus Wilke, Kara Woo, Hiroaki Yutani, and Dewey Dunnington. 2020. *Ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics*. https://CRAN.R-project.org/package=ggplot2.

Wickham, Hadley, Romain François, Lionel Henry, and Kirill Müller. 2021. *Dplyr: A Grammar of Data Manipulation*. https://CRAN.R-project.org/package=dplyr.

Williams, Graham. 2020. *Rattle: Graphical User Interface for Data Science in r*. https://rattle.togaware.com/.

Williams, Graham J. 2017. *The Essentials of Data Science: Knowledge Discovery Using r*. The r Series. CRC Press.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*

Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

26.2 Basic LaTeX Template

20200602 LaTeX is a language for programming, like R—programming over words and sections and chapters and page layout, rather than programming over data. It is what we call a markup language and has its own collection of commands that tell the LaTeX software what to do. Within RStudio when we create a new knitr (Xie 2021) document a skeleton of LaTeX commands will be automatically inserted into the document for us. From this initial script we can readily produce well formatted reports and presentations.

Once we have a new text document with a filename extension of .Rnw we will enter the LaTeX commands intermixed with the text of our narrative intermixed with the R commands performing the data wrangling and the analyses we perform.

```
\documentclass{article}

\begin{document}

\end{document}
```

This is the most basic of documents—something we often call the *Hello World* example, though perhaps we would want to include the string `Hello World` between the `begin` and `end` of the document.

```
\documentclass{article}

\begin{document}

Hello World.

\end{document}
```

LaTeX provides an extensive collection of commands including the basics like highlighting text (for example, `\bold{...}` or `\italic{...}`) and creating a list of items:

```
* Text for the \bold{first} item.
* Text for the \italic{second} item.
* ...
```

We will see many examples as we proceed through this chapter and there are very many guides to LaTeX available on the Internet. Our aim is to quickly be comfortable with using LaTeX and not to let it get in our way, recognising it as a very powerful language.

References

———. 2021. *Knitr: A General-Purpose Package for Dynamic Report Generation in r*. https://yihui.org/knitr/.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

26.3 RStudio with KnitR

20200602 In creating a new knitr (Xie 2021) document within RStudio the minimal LaTeX markup is inserted. To do so, create a new R Sweave document under the File menu, informing RStudio to use knitr (Xie 2021) which is an advanced version of Sweave. Under the Tools menu choose Global Options and then Sweave to set the Weave option to be `knitr`. We can see the RStudio context below. The file contains the following minimal LaTeX code.

Into this file we will begin entering the LaTeX commands and the text of our narrative and the R code. We enter text to describe our project between the begin and end of the document and then choose the `Compile PDF` button on the toolbar to generate a PDF document that we can view.

References

———. 2021. *Knitr: A General-Purpose Package for Dynamic Report Generation in r*. https://yihui.org/knitr/.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

26.4 Template for a Narrative

20200602 A basic template can be the starting point for any new project. A sample available from http://HandsOnDataScience.com/report.Rnw and shown below introduces further LaTeX commands. We've specified the target page size to be A4 on paper. The British language is chosen for the document. A title, author and date are added and `## ...` commands breaks the document into logical sections with some simple instructions for each section.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

26.5 RStudio Compile PDF

20200602 Below we see the interaction with RStudio showing the template within the editor in the top left window pane. When we have processed the document (compiled to PDF) we will see the document formatted nicely into a PDF.

Within the RStudio edit pane we are editting the LaTeX document. A click of the button will run `knitr` over the document then run the LaTeX software to produce a PDF document.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

26.6 Compiled PDF

20200602 The result is this PDF document shown below. Using a programming language (LaTeX) we are able to completely change the appearance of the document to suite specific purposes. Here the layout and style is quite generic, representing the defaults of LaTeX. These include the centering of the title, author and date, the title being in a larger font than the author and date, which are larger than the paragraph fonts, and so on. This can all be tuned, but can require more sophisticated programming or the use of plugin packages (like `babel`).

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

26.7 SweaveOpts Undefined

20200602 Be sure to select `knitr` under . If you have initially run your document using Sweave rather than knitr (Xie 2021) (RStudio defaults to Sweave) and then attempt to change to knitr (Xie 2021), you may find the following error:

```
! Undefined control sequence.
l.54 \SweaveOpts
                 {concordance=TRUE}
The control sequence at the end of the top line
of your error message was never \def'ed. If you have
misspelled it (e.g., `\hobx'), type `I' and the correct
spelling (e.g., `I\hbox'). Otherwise just continue,
and I'll forget about whatever was undefined.
```

We will find this error message towards the end of the log file obtained by clicking the View Log button. A popup will also tell us that we can convert the file using `Sweave2knitr("report.Rnw")`

RStudio has automatically inserted a required line for compiling with Sweave. As the error message suggests, go to the line containing the undefined sequence (line 54 is noted in this instance, but that is in the automatically generated .tex file rather than the source .Rnw file that we are editting so we will need to search for the string) and delete the offending LaTeX control sequence and the bracketed argument. It should then run just fine in knitr (Xie 2021).

References

———. 2021. *Knitr: A General-Purpose Package for Dynamic Report Generation in r*. https://yihui.org/knitr/.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of*

Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

26.8 Including R Commands

20200602 We can include R commands within the knitr (Xie 2021) document and have the commands automatically run when we compile the PDF. The output from the R commands will be displayed together with the R commands themselves.

To include R commands we surround the commands with special markers to create code blocks. The code blocks begin with double less than symbols (or angle brackets) starting in column one and end with double greater than symbols () followed immediately by an equals (). The code block is then terminated by a line containing a single `` `at'' symbol ( @`) starting in column one.

```
<<>>=
... R code ...
@
```

Between the angle brackets we place instructions to tell knitr (Xie 2021) what to do with the R commands. We can tell it to simply echo the commands, but not to evaluate them, or to evaluate the commands without echoing them, and so on. Whilst it is optional it is good practice to provide a label for each block of R code. This is the first element between the angle brackets. A simple example of the beginning of a typical code block is:

```
<<my_label, eval=TRUE, echo=FALSE>>=
```

The label here is `my_label` and we ask knitr (Xie 2021) to evaluate the R commands and thus to also show the output of those commands (this is the default). We do not echo the R commands so that the actual commands themselves will not appear in the resulting document (the default is to echo the commands). Whilst we develop our narrative we will include all of the code chunks and the output into the generated PDF report but once we are ready to produce our final report we might turn the echoing of the R code off.

References

———. 2021. *Knitr: A General-Purpose Package for Dynamic Report Generation in r*. https://yihui.org/knitr/.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

26.9 KnitR Basic Example

20200602 Our first example of running actual R code will generate some random uniform data using stats::runif() and then viewing the utils::head() of the data and calculating the BiocGenerics::mean(). The following code block shows how this will look in the source .Rnw file.

```
<<example_random_mean>>=
# Always include a short comment to support the code.

x <- runif(1000) * 1000
head(x)
mean(x)
@
```

Below is what it looks like after it is processed by knitr (Xie 2021) and then LaTeX (as happens when we click the `Compile PDF` button in RStudio).

```r
# Always include a short comment to support the code.

x <- runif(1000) * 1000
head(x)
```

```
## [1] 371.132988 188.560745  98.770129 467.370752   3.232417 287.578544
```

```r
mean(x)
```

```
## [1] 497.0886
```

Notice that the syntax is colour highlighted and the output is included as comments introduced in R with the `##`. If we were to evaluate these commands ourselves in R the the output would not include the `##`.

References

———. 2021. *Knitr: A General-Purpose Package for Dynamic Report Generation in r*. https://yihui.org/knitr/.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

26.10 Inline R Code

20200602 Often we find ourselves wanting to refer to the results of an R command within the text we are writing rather than as a separate code chunk. This is easily done using the `Sexpr{}` command in LaTeX.

To include today's date within the narrative we can type the command sequence exactly as `Sexpr{Sys.Date()}`. This will be replaced with *2026-07-20* (assuming that is today). Any R function can be called in this way. To format the date, for example, we can use R's base::format() function to specify how the date is displayed, so to produce we would use % `Sexpr{Sys.Date() %>% format(format="%A, %e %B %Y")}`.

We typically intermix a narrative of our dataset with output from R to support and illustrate the discussion. In the following sentence we do this showing first the output from the R command and then the actual R command we included in the source document. For example, the dataset from (G. Williams 2020) has 176,747 (i.e., `\Sexpr{comma(nrow(weatherAUS))}`) observations including observations of the following 4 of the `ncol(weatherAUS)` (i.e., `\Sexpr{ncol(weatherAUS)}`) available variables: MinTemp, MaxTemp, Rainfall, Evaporation. That list was generated using MinTemp, MaxTemp, Rainfall, Evaporation.

LaTeX treats some characters specially and we need to be careful to escape such characters. For example, the underscore `_`" is used to introduce a subscript in LaTeX. It needs to be escaped if we really want an underscore to be included in the compiled PDF document. If not LaTeX will likely complain. As an example we might list one of the variable names from the dataset with an underscore in its name: RISK_MM (`\Sexpr{names(weatherAUS)[23]}`). We will see an error like:

```
KnitR.tex:230: Missing $ inserted.
KnitR.tex:230: leading text: ...an underscore in its name: RISK_
KnitR.tex:232: Missing $ inserted.
```

(Harrell 2021) provides Hmisc::latexTranslate() to assist here. It was used above to print the variable name: `Sexpr{latexTranslate(names(weatherAUS)[23])}`. There are other support functions in Hmisc (Harrell 2021) that are useful when working with knitr (Xie 2021) and LaTeX—

see `library(help=Hmisc)` for further information.

References

Harrell, Frank E, Jr. 2021. *Hmisc: Harrell Miscellaneous*. https://CRAN.R-project.org/package=Hmisc.

Williams, Graham. 2020. *Rattle: Graphical User Interface for Data Science in r*. https://rattle.togaware.com/.

———. 2021. *Knitr: A General-Purpose Package for Dynamic Report Generation in r*. https://yihui.org/knitr/.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*

Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

26.11 Formatting Tables Using Kable

REVIEW Including a typeset table based on a dataset can be accomplished using knitr::kable(). Here we will use the larger dataset from rattle (G. Williams 2020) setting it up as a dplyr::tibble() courtesy of (Wickham et al. 2021). We will then choose specific columns and a random selection of rows to include in the table. The source text we include in our `.Rnw` file is listed in the following code block.

```
<<example_kable, echo=TRUE, results="asis">>=
set.seed(42)
library(rattle)
nobs <- nrow(weatherAUS)
obs  <- sample(nobs, 5)
vars <- 2:6
ds   <- weatherAUS[obs, vars]
kable(ds)
@
```

The result (also showing the R code since we specified *echo*= `TRUE`) is then:

```r
# Set the seed so that results are repeatable.

set.seed(42)

# Load the package from the local library into the R session.

library(rattle)

# Record metadata for a sample of the dataset.

nobs <- nrow(weatherAUS)
obs  <- sample(nobs, 5)
vars <- 2:6
ds   <- weatherAUS[obs, vars]

# Generate the appropriate LaTeX code to display the data.

kable(ds)
```

Location	MinTemp	MaxTemp	Rainfall	Evaporation
Tuggeranong	10.0	25.1	0.2	NA
Wollongong	20.8	30.7	1.0	NA
Dartmoor	5.7	14.9	1.6	2.6
Sale	16.8	28.8	0.0	6.0
WaggaWagga	7.7	31.3	0.0	4.0

Since we are working with a random sample and we would like the sampling to be repeatable we have used base::set.seed() to initialise the random number generator to a fixed value.

References

Wickham, Hadley, Romain François, Lionel Henry, and Kirill Müller. 2021. *Dplyr: A Grammar of Data Manipulation*. https://CRAN.R-project.org/package=dplyr.

Williams, Graham. 2020. *Rattle: Graphical User Interface for Data Science in r*. https://rattle.togaware.com/.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*

Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

26.12 Formatting Options

REVIEW Formatting options are available for fine tuning how the table is to be presented. For example we can remove the row names (the row numbers in the above table) easily with *row.names=* `FALSE` .

```
# Display a table without row names.

ds %>% kable(row.names=FALSE)
```

Location	MinTemp	MaxTemp	Rainfall	Evaporation
Tuggeranong	10.0	25.1	0.2	NA
Wollongong	20.8	30.7	1.0	NA
Dartmoor	5.7	14.9	1.6	2.6
Sale	16.8	28.8	0.0	6.0
WaggaWagga	7.7	31.3	0.0	4.0

We can also limit the number of digits displayed to avoid an impression of a high level of accuracy or to simplify presentation using *digits=*. By doing so the numeric values are rounded to the requested number of decimal points.

```
# Display a table removing digits from numbers.

ds %>% kable(row.names=FALSE, digits=0)
```

Location	MinTemp	MaxTemp	Rainfall	Evaporation
Tuggeranong	10	25	0	NA
Wollongong	21	31	1	NA
Dartmoor	6	15	2	3
Sale	17	29	0	6
WaggaWagga	8	31	0	4

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

26.13 Improvements Using BookTabs

REVIEW The `booktabs` package for LaTeX provides additional functionality that we can make use of with knitr::kable(). To use this be sure to include the following in the preamble (before the `\begin{document}` of your .Rnw file:

```
# Load the package from the local library into the R session.

\usepackage{booktabs}
```

We can then set *booktabs*= `TRUE` to remove the clutter of the extra lines.

```
# Use the booktabs option to improve presentation of the table.

ds %>% kable(row.names=FALSE, digits=0, booktabs=TRUE)
```

Location	MinTemp	MaxTemp	Rainfall	Evaporation
Tuggeranong	10	25	0	NA
Wollongong	21	31	1	NA
Dartmoor	6	15	2	3
Sale	17	29	0	6
WaggaWagga	8	31	0	4

In the following example we notice that with more rows *booktabs*= `TRUE` will add a small gap every 5 rows.

```r
# Display a tale with more observations.

weatherAUS[sample(nobs, 12), vars] %>%
  kable(row.names=FALSE, digits=0, booktabs=TRUE)
```

Location	MinTemp	MaxTemp	Rainfall	Evaporation
Perth	18	36	0	11
Tuggeranong	-4	16	0	NA
Bendigo	3	22	0	NA
Brisbane	15	20	0	3
PearceRAAF	3	22	0	NA
Tuggeranong	12	25	63	NA
Launceston	13	20	29	NA
Tuggeranong	10	29	0	NA
Wollongong	16	23	0	NA
CoffsHarbour	5	20	0	NA
Hobart	13	20	2	3
Woomera	24	46	0	NA

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

26.14 Formatting Tables Using XTable

REVIEW Whilst knitr::kable() provides basic functionality much more extensive control over the formatting of tables is provided by (Dahl et al. 2019). By default the table produced is called a floating table so that it floats within the document to an appropriate location.

As a floating table we will add a *caption*= and a table reference *label*= to the table so that they do not get lost. We can then refer to the table within the text and have the tables appear somewhere convenient automatically. The code block below, for example, produces Table **??** on page . The table and page numbers are automatically assigned to the table. Within LaTeX we can access the table number using `\@ref(egtbl)` (**??**) and the page number using `\pageref{egtbl}` ().

```
# Load the package from the local library into the R session.

library(xtable)

# Generate a floating table with a caption.

ds %>% xtable(caption="Example xtable.", label="egtbl")
```

% latex table generated in R 4.1.0 by xtable 1.8-4 package % Thu Jun 24 16:09:23 2021
Also note that by default missing values (`NA`) are not printed nor are the extra lines that are printed by default when using knitr::kable().

There are very many formatting options available for fine tuning how the table is to be presented and we cover some of these in the following pages. We also note that some options are provisioned by xtable::xtable() whilst others are available through xtable::print.xtable(). An example option is *include.rownames*= which is an option available with xtable::print.xtable(). The result is seen in Table **??**.

```
# Display a table without row names.

ds %>%
  xtable(caption="Remove row numbers.", label="tblnonums") %>%
  print(include.rownames=FALSE)
```

% latex table generated in R 4.1.0 by xtable 1.8-4 package % Thu Jun 24 16:09:23 2021

References

Dahl, David B., David Scott, Charles Roosen, Arni Magnusson, and Jonathan Swinton. 2019. *Xtable: Export Tables to LaTeX or HTML*. http://xtable.r-forge.r-project.org/.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

26.15 Formatting Numbers with XTable

REVIEW As with knitr::kable() we can limit the number of digits displayed to avoid giving an impression of a high level of accuracy or to simplify the presentation. In Table **??** we have removed all decimal points.

```
# Display a table removing digits from numbers.


ds %>%
  xtable(digits=0
       , caption="Decimal points."
       , label="tbldp0") %>%
  print(include.rownames=FALSE)
```

% latex table generated in R 4.1.0 by xtable 1.8-4 package % Thu Jun 24 16:09:23 2021
When we have large numbers being displayed it is imperative that we include commas to separate the thousands. Very many mistakes are made misreading numbers that include many digits when commas are not included.

```r
# Take a copy of the dataset so as to change the data.

dst     <- ds %>% sample_frac(0.01)

# Randomly create very large numbers across all but the first variable.

dst[-1] <- sample(10000:99999, nrow(dst)) * dst[-1]

# Illustrate the default table display of large numbers.

dst %>%
  xtable(digits=0
       , caption="Large numbers."
       , label="tbllrg") %>%
  print(include.rownames=FALSE)
```

% latex table generated in R 4.1.0 by xtable 1.8-4 package % Thu Jun 24 16:09:23 2021 Consider the result in Table **??**. It is difficult to distinguish between the thousands and millions. We often find ourselves having to carefully count the digits to check whether # A tibble: 1 x 1, MaxTemp, , 1 NA really is # A tibble: 1 x 1, MaxTemp, , 1 NA. To avoid this cognitive load on the reader we should always use a comma to separate the thousands and millions. This simple principle makes it much easier for the reader to appreciate the scale and to avoid misreading data, yet it is so often overlooked. We can see the result in Table **??**.

```r
# Format large numbers using commas as appropriate.

dst %>%
  xtable(digits=0
       , caption="Large numbers formatted."
       , label="tbllrgf") %>%
  print(include.rownames=FALSE,
        format.args=list(big.mark=","))
```

% latex table generated in R 4.1.0 by xtable 1.8-4 package % Thu Jun 24 16:09:23 2021

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

26.16 Adding a Caption and Reference Label

REVIEW We have seen captions and labels above but tell the story here in a slightly different way.

```
print(xtable(ds,
             digits=0,
             caption="Selected observations from **weatherAUS**."),
      include.rownames=FALSE)
```

% latex table generated in R 4.1.0 by xtable 1.8-4 package % Thu Jun 24 16:09:23 2021
Notice that we wanted to emphasise the name of the dataset in the caption. We can make it bold using the `\textbf{}` command of LaTeX within the string passed to *caption*=. We need to repeat the backslash because R itself will attempt to interpret it otherwise. That is, we *escape* the backslash.

As well as adding a caption with a table number, we can add a label to the xtable::xtable() command and then refer to the table within the text using the `\@ref(MyTable)` LaTeX command. Thus, in the source document we use `` `Table \@ref(MyTable)`'' to produce Table **??**'' in the documnet.

```
print(xtable(ds,
             digits=0,
             caption="Selected observations from **weatherAUS**.",
             label="MyTable"),
      include.rownames=FALSE)
```

% latex table generated in R 4.1.0 by xtable 1.8-4 package % Thu Jun 24 16:09:23 2021
The references to Table **??** can appear anywhere in the document. We can even refer to its page number by replacing `\@ref(MyTable)` with `\pageref{MyTable}` to get Table **??** on Page .

–>

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

26.17 Sophisticated Captions

REVIEW Captions can be formatted quite extensively with a little knowledge of LaTeX. For Table **??** we illustrate generating a string in R that is passed through to the *caption*=. We use BiocGenerics::paste() and base::Sys.time() and include some special symbols known to LaTeX as well as an occasion bold and italic font. Notice that because the caption is quite long we do not want the whole caption included in the list of tables in the contents pages. The second argument to *caption*= is the short title to use for the list of tables.

```
# Create a long caption as a single srting.

cpt <- paste("Here we include in the **caption**"
           , "a sample of LaTeX symbols and formats"
           , "that can be included in the string, and"
           , "note that the caption string can be the"
           , "result of R commands, using"
           , "\`paste()` in this instance. Some"
           , "sample symbols include: $\\alpha$"
           , "$\\rightarrow$ $\\wp$."
           , "We also get a timestamp from R:"
           , Sys.time())

# Add the caption to the table.

dst %>%
  xtable(digits=0
       , caption=c(cpt, "Extended caption.")
       , label="tblcap") %>%
  print(include.rownames=FALSE)
```

% latex table generated in R 4.1.0 by xtable 1.8-4 package % Thu Jun 24 16:09:24 2021

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

26.18 Including Figures

REVIEW Plots and other graphics can be the difference between a dull report that few will ever read and an exciting report that invites the reader into an interesting narrative. Through knitr (Xie 2021) we can readily include the stunning graphics that can be generated by R. Whilst we have yet to cover this capability, some simple examples will suffice for now.

References

———. 2021. *Knitr: A General-Purpose Package for Dynamic Report Generation in r*. https://yihui.org/knitr/.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

26.19 Sample Figure

REVIEW To include figures generated by R in our document we simply add plotting commands to the code chunk. Here for example is R code to generate a simple density plot of the 3pm temperature in 4 cities over a year. We use (Wickham et al. 2020) to generate the figure.

```
# Load packages from the local library into the R session.

library(rattle)   # For the weatherAUS dataset.
library(ggplot2)  # To generate a density plot.

# Identify cities of interest.

cities <- c("Canberra", "Darwin", "Melbourne", "Sydney")

# Generate the plot.

weatherAUS %>%
  subset(Location %in% cities & ! is.na(Temp3pm)) %>%
  ggplot(aes(x=Temp3pm, colour=Location, fill=Location)) +
  geom_density(alpha=0.55)
```

In the source document (the .Rnw file) the above R code is actually inserted between the chunk begin and end marks within the document itself. Those marks are:

```
<<myfigure, eval=FALSE>>=
... R code ...
@
```

Notice the use of *eval*= `FALSE`, which allows the R code to be included in the text of the final document, as it is above, but will not yet generate the plot to be included in the figure. We leave that for a little later.

The code chunk begins by attaching the requisite packages: (G. Williams 2020) to access the dataset; and (Wickham et al. 2020) for the function to generate the actual plot.

The four cities we wish to plot are then identified, and we generate a BiocGenerics::subset() of the weatherAUS dataset containing just those cities. We pass the subset on to ggplot2::ggplot() and identify `Temp3pm` for the x-axis, using `location` to colour and fill the plot. We add a layer to the figure containing a density plot with a level of transparency specified as an *alpha*= value. We can see the figure below.

```
<<myfigure, eval=FALSE>>=
... R code ...
@
```

We include the figure in the final document as above simply by removing the *eval*= `FALSE` from the previous code chunk. Thus the R code is evaluated and a plot is generated. We have actually replaced the *eval*= `FALSE` with *echo*= `FALSE` so as not to print the R code a second time.

We do not actually need to rewrite the R code again in a second chunk, given the code has already been provided in the first chunk on the previous page. We use a feature of knitr (Xie 2021) where an empty chunk having the same name as a previous chunk is actually a reference to that previous chunk. Thus in our source .Rnw text document we add the following two lines. This is effectively replaced by the R code from the previous block of the same name.

```
<<myfigure, echo=FALSE}
```

```
@
```

This is exactly what we included at the beginning of this section in the actual source document for this page. Noticing that we have replaced *eval*= `FALSE` with *echo*= `FALSE`, we cause the original R code to be executed, generating the plot which is included as the figure above. Using *echo*= `FALSE` simply ensures we do not include the R code itself in the final output, this time. That is, the R code is replaced with the figure it generates.

Notice how the figure takes up quite a bit of space on the page.

References

Wickham, Hadley, Winston Chang, Lionel Henry, Thomas Lin Pedersen, Kohske Takahashi, Claus Wilke, Kara Woo, Hiroaki Yutani, and Dewey Dunnington. 2020. *Ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics*. https://CRAN.R-project.org/package=ggplot2.

Williams, Graham. 2020. *Rattle: Graphical User Interface for Data Science in r*. https://rattle.togaware.com/.

———. 2021. *Knitr: A General-Purpose Package for Dynamic Report Generation in r*. https://yihui.org/knitr/.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*

Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

26.20 Adjusting Aspect

REVIEW

```
# Load packages from the local library into the R session.

library(rattle)   # For the weatherAUS dataset.
library(ggplot2)  # To generate a density plot.

# Identify cities of interest.

cities <- c("Canberra", "Darwin", "Melbourne", "Sydney")

# Generate the plot.

weatherAUS %>%
  subset(Location %in% cities & ! is.na(Temp3pm)) %>%
  ggplot(aes(x=Temp3pm, colour=Location, fill=Location)) +
  geom_density(alpha=0.55)
```

We can fine tune the size of the figure to suit the document and presentation. In this example we have asked R to widen the figure from 7 inches to 14 inches using *fig.width*=. The code chunk is:

```
<<myfigure, echo=FALSE, fig.width=14}
... R code ...
@
```

Underneath, knitr (Xie 2021) is using a PDF device on which the plot is generated, and then saved to file for inclusion in the final document. The PDF device grDevices::pdf(), by default, will generate a 7 inch by 7 inch plot (see `?pdf` for details). This is the plot dimensions as we see on the previous page. By setting *fig.width*= (and *fig.height*=) we can change the dimensions. In our example here we have doubled the width, resulting in a more pleasing plot.

Notice that as a consequence of the figure being larger the fonts have remained the same size, resulting them appearing smaller now when we include the figure in the same area on the printed page.

References

———. 2021. *Knitr: A General-Purpose Package for Dynamic Report Generation in r*. https://yihui.org/knitr/.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

26.21 Choosing Dimensions

REVIEW

```
# Load packages from the local library into the R session.

library(rattle)   # For the weatherAUS dataset.
library(ggplot2)  # To generate a density plot.

# Identify cities of interest.

cities <- c("Canberra", "Darwin", "Melbourne", "Sydney")

# Generate the plot.

weatherAUS %>%
  subset(Location %in% cities & ! is.na(Temp3pm)) %>%
  ggplot(aes(x=Temp3pm, colour=Location, fill=Location)) +
  geom_density(alpha=0.55)
```

Often a bit of trial and error is required to get the dimensions right. Notice though that increasing the *fig.width*= as we did for the previous plot, and/or increasing the *fig.height*=, effectively also reduces the font size. Actually, the font size remains constant whilst the figure grows (or shrinks) in size. Sometimes it is better to reduce the fig.width or fig.height to retain a good sized font.

The above plot was generated with the following knitr (Xie 2021) options.

```
<<myfigure, echo=FALSE, fig.height=3.5}
... R code ...
@
```

References

———. 2021. *Knitr: A General-Purpose Package for Dynamic Report Generation in r*. https://yihui.org/knitr/.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

26.22 Setting Output Width

REVIEW

```
# Load packages from the local library into the R session.

library(rattle)   # For the weatherAUS dataset.
library(ggplot2)  # To generate a density plot.

# Identify cities of interest.

cities <- c("Canberra", "Darwin", "Melbourne", "Sydney")

# Generate the plot.

weatherAUS %>%
  subset(Location %in% cities & ! is.na(Temp3pm)) %>%
  ggplot(aes(x=Temp3pm, colour=Location, fill=Location)) +
  geom_density(alpha=0.55)
```

We can use the *out.width*= and *out.height*= to adjust how much space a figure takes up in the final document. The above figure is reduced to fill just half the textwidth of the document using:

```
<<myfigure, fig.align="center", fig.width=3.5, out.width="0.5\\textwidth"}
... R code ...
@
```

If that is too wide, we can reduce it to 90% of the page width with:

```
<<myfigure, fig.align="center", fig.width=3.5, out.width="0.9\\textwidth"}
... R code ...
@
```

```r
# Load packages from the local library into the R session.

library(rattle)   # For the weatherAUS dataset.
library(ggplot2)  # To generate a density plot.

# Identify cities of interest.

cities <- c("Canberra", "Darwin", "Melbourne", "Sydney")

# Generate the plot.

weatherAUS %>%
  subset(Location %in% cities & ! is.na(Temp3pm)) %>%
  ggplot(aes(x=Temp3pm, colour=Location, fill=Location)) +
  geom_density(alpha=0.55)
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

26.23 Add a Caption and Label

REVIEW

```
# Load packages from the local library into the R session.

library(rattle)   # For the weatherAUS dataset.
library(ggplot2)  # To generate a density plot.

# Identify cities of interest.

cities <- c("Canberra", "Darwin", "Melbourne", "Sydney")

# Generate the plot.

weatherAUS %>%
  subset(Location %in% cities & ! is.na(Temp3pm)) %>%
  ggplot(aes(x=Temp3pm, colour=Location, fill=Location)) +
  geom_density(alpha=0.55)
```

Adding a caption (which automatically also adds a label) is done using *fig.cap*=.

```
<<myfigure, fig.cap="The 3pm temperature for four locations.", fig.pos="h",...}
... R code ...
@
```

We have also used (but can't see) *fig.pos*= `"h"` which requests placement of the figure `` `here'' rather than letting it float. Other options are to place the figure at the top of a page ( "t" ), or the bottom of a page ( "b"`). We can leave it empty and the placement is done automatically—that is, the figure floats to an appropriate location.

Once a caption is added, a label is also added to the figure so that it can be referred to in the document. The label is made up of `fig:` followed by the chunk label, which is `myfigure` in this example. So we can refer to the figure using `\@ref(fig:myfigure)` and `\pageref{fig:myfigure}`, which allows us to refer to Figure @ref(fig:knitr:myfigure_3pm_temp) on Page .

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

26.24 Animation: Basic Example

REVIEW

We can generate multiple plots and they then form an animation. For this to work, we use the knitr (Xie 2021) option *fig.show*= `"animate"`. For the LaTeX processing we then need to attach the `animate` package:

```
\usepackage{animate}
```

To view the animation live we need to use Acrobat Reader to view the PDF file.

This example comes from http://taiyun.github.io/blog/2012/07/k-means/.

```
library(animation)
par(mar=c(3, 3, 1, 1.5), mgp=c(1.5, 0.5, 0), bg="white")
cent <- 1.5 * c(1, 1, -1, -1, 1, -1, 1, -1)
x <- NULL
for (i in 1:8) x <- c(x, rnorm(25, mean=cent[i]))
x <- matrix(x, ncol=2)
colnames(x) <- c("X1", "X2")
kmeans.ani(x, centers=4, pch=1:4, col=1:4)
```

References

———. 2021. *Knitr: A General-Purpose Package for Dynamic Report Generation in r*. https://yihui.org/knitr/.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

26.25 Adding a Flowchart

REVIEW LaTeX and R both support extensive capability for generating pictures. Here we use the (Soetaert 2020) package to draw a flow chart, something that is a common requirement in documentation. The simple example shows the process of converting a .Rnw file using knitr (Xie 2021) to a LaTeX file which can then be processed by `pdflatex` to generate the .pdf file. The result can be seen in Figure @ref(fig:knitr:knitr_diagram_flowchart).

```
# Load package from the local library into the R session.

library(diagram)

# Identify some names for the nodes of the flowchart.

names   <- c(".Rnw", ".tex", ".pdf")

# Set up the connectivity data.

connect <- c(0,       0,          0,
             "knitr", 0,          0,
             0,       "pdflatex", 0)

# Constract a connectivity matrix and plot the flowchart.

matrix(nrow=3, ncol=3, byrow=TRUE, data=connect) %>%
  plotmat(pos=c(1, 2), name=names, box.col="orange")
```

(#fig:knitr:knitr_diagram_flowchart)The knitr process to generate a pdf.

There are many more possibilities provided by diagram (Soetaert 2020) and if interested you can explore `demo(plotmat)` and `demo(plotweb)`.

References

Soetaert, Karline. 2020. *Diagram: Functions for Visualising Simple Graphs (Networks), Plotting Flow Diagrams*. https://CRAN.R-project.org/package=diagram.

———. 2021. *Knitr: A General-Purpose Package for Dynamic Report Generation in r*. https://yihui.org/knitr/.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*

Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

26.26 Adding Bibliographies

REVIEW knitr (Xie 2021) supports the automatic generation of a bibliography for the packages attached into R. We can take advantage of this by specifying a LaTeX bibliography package like `natbib`. I prefer the (author, year) style and so I include the following in the preamble of my LaTeX documents.

```
\usepackage[authoryear]{natbib}
```

If we attach the rattle (G. Williams 2020) package, we might like to cite it with the following LaTeX command:

```
[@R-rattle]
```

This will produce a citation like (G. Williams 2020). The `` `p'' in `\citep` places the parentheses around the citation, without which we get G. Williams (2020).

Then we can ask knitr (Xie 2021) to generate bibliographic entries for each attached package:

```
write_bib(sub("^.*/", "", grep("^/", searchpaths(), value=TRUE)),
          file="mydoc.bib")
```

Note that the bibliography is saved to a file named `mydoc.bib`.

At the end of the LaTeX document, where we want the bibliography to appear, we add the following:

```
\bibliographystyle{jss}
\bibliography{mydoc}
```

See Page to see what the bibliography will look like.

References

Williams, Graham. 2020. *Rattle: Graphical User Interface for Data Science in r*. https://rattle.togaware.com/.

———. 2021. *Knitr: A General-Purpose Package for Dynamic Report Generation in r*. https://yihui.org/knitr/.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*

Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

26.27 Referencing Chunks in LaTeX

REVIEW We may like to reference code chunks from other parts of our document. LaTeX handles cross references nicely, and there are packages that extend this to user defined cross references. The `amsthm` package provides one such mechanism.

To add a cross referencing capability we can tell LaTeX to use the `amsthm` package and to create a new theorem environment called `rcode` by including the following in the document preamble (prior to the `\begin{document}`):

```
\usepackage{amsthm}
\newtheorem{rcode}{R Code}[section]
```

We then tell knitr (Xie 2021) to add an `rcode` environment around the chunks. The following chunk achieves this by adding some code to a hook function for knitr (Xie 2021). This will typically also be in the preamble of the document, though not necessarily.

```
<<setup, include=FALSE}
knit_hooks$set(rcode=function(before, options, envir)
{
  if (before)
    sprintf('\\begin{rcode}\ {#%s}l{}', options$label)
  else
    '\\end{rcode}'
})
@
```

Once we've done that, we add *rcode*= `TRUE` for those chunks we wish to refer to. Here is a chunk as an example:

```
<<demo_chunk_ref, rcode=TRUE}
seq(0, 10, 2)
@
```

The output from the chunk is:

```
seq(0, 10, 2)
```

```
## [1]  0  2  4  6  8 10
```

We can then refer to the R code chunk with `\@ref(demo_chunk_ref)`, which prints the R Code reference number as @ref(demo_chunk_ref), and `\pageref{demo_chunk_ref}`, which prints the page number as .

References

———. 2021. *Knitr: A General-Purpose Package for Dynamic Report Generation in r*. https://yihui.org/knitr/.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

26.28 Truncating Long Lines

REVIEW The output from knitr (Xie 2021) will sometimes be longer than fits within the limits of the page. We can add a hook to knitr (Xie 2021) so that whenever a line is longer than some parameter, it is truncated and replaced with `` `...''. The hook extends the output function. Notice we take a copy of the current output` hook and then run that after our own processing.

```
opts_chunk$set(out.truncate=80)
hook_output <- knit_hooks$get("output")
knit_hooks$set(output=function(x, options)
{
  if (options$results != "asis")
  {
    # Split string into separate lines.
    x <- unlist(stringr::str_split(x, "\n"))
    # Truncate each line to length specified.
    if (!is.null(m <- options$out.truncate))
    {
      len <- nchar(x)
      x[len>m] <- paste0(substr(x[len>m], 0, m-3), "...")
    }
    # Paste lines back together.
    x <- paste(x, collapse="\n")
    # Continue with any other output hooks
  }
  hook_output(x, options)
})
```

This is useful to avoid ugly looking long lines that extend beyond the limits of the page. We can illustrate it here by first not truncating at all (`out.truncate=NULL`):

```r
paste("This is a very long line that is truncated",
      "at character 80 by default. We change the point",
      "at which it gets truncated using out.truncate=")
```

```
## [1] "This is a very long line that is truncated at character 80 by default. We cl
```

Now we use the default to truncate it.

```r
paste("This is a very long line that is truncated",
      "at character 80 by default. We change the point",
      "at which it gets truncated using out.truncate=")
```

```
## [1] "This is a very long line that is truncated at character 80 by default. We cl
```

Here is another example, with `out.truncate=40` included in the knitr options.

```r
paste("This is a very long line that is truncated",
      "at character 80 by default. We change the point",
      "at which it gets truncated using out.truncate=")
```

```
## [1] "This is a very long line that is truncated at character 80 by default. We cl
```

References

———. 2021. *Knitr: A General-Purpose Package for Dynamic Report Generation in r*. https://yihui.org/knitr/.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

26.29 Truncating Too Many Lines

REVIEW Another issue we sometimes want to deal with is limiting the number of lines of output displayed from knitr (Xie 2021). We can add a hook to knitr (Xie 2021) so that whenever we have reached a particular line count for the output of any command, we replace all the remaining lines with `` `...''. The hook again extends the output function. Notice we take a copy of the current output` hook and then run that after our own processing.

```
opts_chunk$set(out.lines=4)
hook_output <- knit_hooks$get("output")
knit_hooks$set(output=function(x, options)
{
  if (options$results != "asis")
  {
    # Split string into separate lines.
    x <- unlist(stringr::str_split(x, "\n"))
    # Trim to the number of lines specified.
    if (!is.null(n <- options$out.lines))
    {
      if (length(x) > n)
      {
        # Truncate the output.
        x <- c(head(x, n), "....\n")
      }
    }
    # Paste lines back together.
    x <- paste(x, collapse="\n")
  }
  hook_output(x, options)
})
```

We can then illustrate it:

```
weather[2:8]
```

```
## # A tibble: 366 x 7
##    Location MinTemp MaxTemp Rainfall Evaporation Sunshine WindGustDir
##    <chr>      <dbl>   <dbl>    <dbl>       <dbl>    <dbl> <ord>
##  1 Canberra     5      24.3      0           3.4      6.3 NW
##  2 Canberra    14      26.9      3.6         4.4      9.7 ENE
##  3 Canberra    13.7    23.4      3.6         5.8      3.3 NW
##  4 Canberra    13.3    15.5     39.8         7.2      9.1 NW
##  5 Canberra     7.6    16.1      2.8         5.6     10.6 SSE
##  6 Canberra     6.2    16.9      0           5.8      8.2 SE
##  7 Canberra     6.1    18.2      0.2         4.2      8.4 SE
##  8 Canberra     8.3    17        0           5.6      4.6 E
##  9 Canberra     8.8    19.5      0           4        4.1 S
## 10 Canberra     8.4    22.8     16.2         5.4      7.7 E
## # … with 356 more rows
```

Now we set `out.lines=2` to only include the first two lines.

```
weather[2:8]
```

```
## # A tibble: 366 x 7
##    Location MinTemp MaxTemp Rainfall Evaporation Sunshine WindGustDir
##    <chr>      <dbl>   <dbl>    <dbl>       <dbl>    <dbl> <ord>
##  1 Canberra     5      24.3      0           3.4      6.3 NW
##  2 Canberra    14      26.9      3.6         4.4      9.7 ENE
##  3 Canberra    13.7    23.4      3.6         5.8      3.3 NW
##  4 Canberra    13.3    15.5     39.8         7.2      9.1 NW
##  5 Canberra     7.6    16.1      2.8         5.6     10.6 SSE
##  6 Canberra     6.2    16.9      0           5.8      8.2 SE
##  7 Canberra     6.1    18.2      0.2         4.2      8.4 SE
##  8 Canberra     8.3    17        0           5.6      4.6 E
##  9 Canberra     8.8    19.5      0           4        4.1 S
## 10 Canberra     8.4    22.8     16.2         5.4      7.7 E
## # … with 356 more rows
```

References

———. 2021. *Knitr: A General-Purpose Package for Dynamic Report Generation in r*. https://yihui.org/knitr/.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

26.30 Selective Lines of Code

REVIEW Another useful formatting trick is to include only the top few and bottom few lines of a block of code. We can again do this using hooks. This time it is through a source hook.

```
opts_chunk$set(src.top=NULL)
opts_chunk$set(src.bot=NULL)
knit_hooks$set(source=function(x, options)
{
  # Split string into separate lines.
  x <- unlist(stringr::str_split(x, "\n"))
  # Trim to the number of lines specified.
  if (!is.null(n <- options$src.top))
  {
    if (length(x) > n)
    {
      # Truncate the output.
      if (is.null(m <-options$src.bot)) m <- 0
      x <- c(head(x, n+1), "\n....\n", tail(x, m+2))
   }
  }
  # Paste lines back together.
  x <- paste(x, collapse="\n")
  hook_source(x, options)
})
```

Now we repaet this code chunk in the source of this current document, but we set `src.top=4` and `src.bot=4`:

```r
opts_chunk$set(src.top=NULL)
opts_chunk$set(src.bot=NULL)
knit_hooks$set(source=function(x, options)
{
  # Split string into separate lines.
  x <- unlist(stringr::str_split(x, "\n"))
  # Trim to the number of lines specified.
  if (!is.null(n <- options$src.top))
  {
    if (length(x) > n)
    {
      # Truncate the output.
      if (is.null(m <-options$src.bot)) m <- 0
      x <- c(head(x, n), "\n....\n", tail(x, m+2))
    }
  }
  # Paste lines back together.
  x <- paste(x, collapse="\n")
  hook_source(x, options)
})
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

26.31 Knitr Options

REVIEW We have explored only some of the functionality of knitr (Xie 2021) and LaTeX. Together there is virtually nothing they can not do—being programming languages anything that can be done, can be done using these tools.

As a summary below we list the common knitr (Xie 2021) options. The options are added to the chunk start line or else we can set the options using `opts_chunk\$set()`. The arguments to the function can be any number of named options with their values. For example:

```
# Set global defaults for knitr options.

opts_chunk$set(size="footnotesize", message=FALSE, tidy=FALSE)
```

Once this is run, the options remain in force as the default values until they are again changed using `opts_chunk\$set()`. They can be overriden per chunk in the usual way.

```
background="#F7F7F7"        # The background colour of the code chunks.
cache.path="cache/"         #
comment=NA                  # Suppresses "`## `" in R output.
echo=FALSE                  # Do not show R commands---just the output.
echo=3:5                    # Only echo lines 3 to 5 of the chunk.
eval=FALSE                  # Do not run the R code---its just for display.
eval=2:4                    # Only evaluate lines 2 to 4 of the chunk.
fig.align="center"          #
fig.cap="Caption..."        #
fig.keep="high"             #
fig.lp="fig:"               # Prefix for the label assigned to the figure.
fig.path="figures/plot"     #
fig.scap="Short cap."       # For the table of figures in the contents.
fig.show="animate"          # Collect figures into an animation.
fig.show="hold"             #
fig.height=9                # Height of generated figure.
fig.width=12                # Width of generated figure.
include=FALSE               # Include code but not output/picture.
message=FALSE               # Do not display messages from the commands.
out.height=".6\\textheight" # Figure takes up 60% of the page height.
out.width=".8\\textwidth"   # Figure takes up 80% of the page width.
results="markup"            # The output from commands will be formatted.
results="hide"              # Do not show output from commands.
results="asis"              # Retain R command output as LaTeX code.
size="footnotesize"         # Useful for Beamer slides.
tidy=FALSE                  # Retain my own formatting used in the R code.
```

New options defined in this Chapter and used in this book:

```
out.lines=4                 # Number of lines of R output to show.
out.truncate=80             # Truncate R output lines beyond this.
src.bot=NULL                # Number of lines of output at top to show.
src.top=NULL                # Number of lines of output at bottom to show.
```

References

———. 2021. *Knitr: A General-Purpose Package for Dynamic Report Generation in r*. https://yihui.org/knitr/.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

26.32 Knitr Resources

We list below other resources that augment the material we have presented in this chapter.

- Xie (2014) delves much further into knitr and LaTeX and other document formats supported by knitr.
- Gandrud (2014) delves not only into knitr, LaTeX, R and RStudio, but also the suite of tools available for the professional data scientist, including utilities such as make, git, and shell.
- http://yihui.name/knitr/ is knitr's home. This includes documentation on knitr (Xie 2021) and useful discussion forums.
- The knitr (Xie 2021) author's presentation on knitr (Xie 2021) http://yihui.name/slides/2012-knitr-RStudio.html. The presentation provides a basic introduction to knitr (Xie 2021).
- http://bcb.dfci.harvard.edu/~aedin/courses/ReproducibleResearch/ReproducibleResearch.pdf This lecture provides some insights into literate programming in R.
- LaTeX supports many characters and symbols but finding the right incantation is difficult. This web site, http://detexify.kirelabs.org/classify.html, does a great job. Draw the character you want and it will show you how to get it .

References

Gandrud, Christopher. 2014. *Repreoducible Research with R and RStdio*. The R Series. CRC Press.

Xie, Yihui. 2014. *Dynamic Documents with R and Knitr*. The R Series. CRC Press.

———. 2021. *Knitr: A General-Purpose Package for Dynamic Report Generation in r*. https://yihui.org/knitr/.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*

Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

27 Coding with Style

20200105 Data scientists write programs to ingest, manage, wrangle, visualise, analyse, report on, data in many ways, followed by training, evaluating, deploying and maintaining models. It is an art to be able to communicate our explorations and understandings as data scientists, and of course we do so through language. Whether that is a programming language or a natural language, our focus needs to be on communication and an understanding of who we are communicating to. Indeed it is this last point that is often misunderstood. We regularly find that most programmers spend most of their time reading other peoples programs and increasingly, people not trained in software engineering are reading other people's programs. Programming languages become the means of communicating between people.

We can be as creative and differently expressive and personal in using programming languages as we are in using natural languages. Indeed, we can sometimes identify the authors of narratives written in a particular programming language, just as we might identify a Shakespearean play from a Noël Coward play.

Thus when we communicate with each other using programming languages, we must keep in mind that it truly is a mechanism for communicating to other people.

Of course our programs must be executable by computers but computers generally care little about our programs except that they be syntactically correct. So our focus should be on engaging others to read and understand the narratives we present through our programs. A badly written and presented program is not a pleasure to read. And to boot, it's not a pleasant experience for the computer either, in the sense that where care is not taken, our narratives will more like have bugs.

Please, aim to write programs that clearly and effectively communicate the story of our data to others, no matter how short or long that program is. Think of the other. Have empathy for those who need to read what you write.

In this chapter we present simple stylistic guidelines for programming in R that support and encourage the transparency of our programs. Our programming style aims to ensure consistency and ease our understanding whilst of course also encouraging correct programs for execution by computer. Over time develop your own stylistic nuances and idiosyncrasies.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

27.1 Style Matters

20200105 Programming is an art and a way to express ourselves. Often that expression is unique to us individually. Just as we can often ascertain the author of a play or the artist of a painting from their style we can often tell the programmer from the program coding structures and styles.

As we write programs we should keep in mind that something like 90% of a programmers' time (at least in business and government) is spent reading and modifying and extending other programmers' code. We need to facilitate the task—so that others can quickly come to a clear understanding of the narrative.

As data scientists we also practice this art of programming and indeed even more so to share the narrative of what we discover through our living and breathing of data. Writing our programs so that others understand why and how we analysed our data is crucial. Data science is so much more than simply building black box models—we should be seeking to expose and share the process and the knowledge that is discovered from the data.

%A common dichotomy in the artificial intelligence and machine learning %communities over the years has viewed ourselves as either building %black boxes that behave intelligently or else seeking to understand %intelligence and to then capture this knowledge computationally. Data %science often lives in the latter camp with the goal to discover and %expose knowledge and to then operationalise that knowledge in %intelligent applications. Telling the narrative we discover from our %data is important.

Data scientists rarely begin a new project with an empty coding sheet. Regularly we take our own or other's code as a starting point and begin from that. We find code on Stack Overflow or elsewhere on the Internet and modify it to suit our needs. We collect templates from other data scientists and build from there, tuning the templates for our specific needs and datasets.

In being comfortable to share our code and narratives with others we often develop a style. Our style is personal to us as we innovate and express ourselves and we need consistency in how we do that. Often a style guide helps us as we journey through a new language and gives us a foundation for developing, over time, our own style.

A style guide is also useful for sharing our tips and tricks for communicating clearly through our programs—our expression of how to solve a problem or actually how we model the world. We express this in the form of a language—a language that also happens to be executable by a computer. In this language we follow precisely specified syntax/grammar to develop sentences, paragraphs, and whole stories. Whilst there is infinite leeway in how we express ourselves and we each express ourselves differently, we share a common set of principles as our style guide.

The style guide here has evolved from over 30 years of programming and data experience. Nonetheless we note that style changes over time. Change can be motivated by changes in the technology itself and we should allow variation as we mature and learn and change our views.

Irrespective of whether the specific style suggestions here suit you or not, when coding do aim to communicate to other readers in the first instance. When we write programs we .

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

27.2 Naming Files

20201102

Files containing R code use the uppercase `.R` extension. This aligns with the fact that the language is unambiguously called `R` and not `r`.

Preferred

```
power_analysis.R
```

Discouraged

```
power_analysis.r
```

Some files may contain a single support function. Name the file to match the name of the function defined within the file. If a file contains the support function `my_fancy_plot()` then name the file as below to differentiate it from analysis scripts.

Preferred

```
my_fancy_plot.R
```

Discouraged

```
utility_functions.R
MyFancyPlot.R
my.fancy.plot.R
my_fancy_plot.r
```

R binary data filenames end in `.RData`. This is descriptive of the file containing data for R and conforms to a capitalised naming scheme.

Preferred

```
weather.RData
```

Discouraged

```
weather.rdata
weather.Rdata
weather.rData
```

Standard file names use lowercase where there is a choice.

Preferred

```
weather.csv
```

Discouraged

```
weather.CSV
```

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

27.3 Multiple File Scripts

20200105

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

27.4 Naming Objects

20201102

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

27.5 Naming Functions

20201102

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

27.6 Comments

20200105

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

27.7 Layout

20200105

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

27.8 If-Else Issue

20200105

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

27.9 Indentation

20200105

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

27.10 Alignment

20200105

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

27.11 Sub-Block Alignment

20200105

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

27.12 Function Guidelines

20200105

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

27.13 Function Definition Layout

20200105

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

27.14 Function Call Layout

20200105

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

27.15 Functions from Packages

20200105

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

27.16 Assignment

20200105

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

27.17 Miscellaneous

20200105

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

27.18 Good Practise

20200105

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

27.19 Kuhn Checklist

Max Kuhn, author of developed a checklist and posted it to the R developers mailing list in January 2012. I have paraphrased some of the points here and embellished it a little with my views, but they are mostly in sync with Kuhn's views.

- Extend the work of others and avoid redundancy. Reuse others functions, with due credit, to add any missing features.
- For a categorical model builder ensure the target is a factor (like Yes/No) rather than integers (like 1/0). The factor levels should be identified in the resulting model object and the stats::predict() function should return predicted classes as factors with the same levels and ordering of levels. Support a to switch between predicted classes and class probabilities. Use for probabilities.
- Implement a separate stats::predict(), using `predict.class()` where *class* is the class of the object returned by the model builder. Do not use special functions like `modelPredict()`.
- Provide both a formula interface as in `foo` (y~{}x, data=ds) and non-formula interface as in `foo` (x, y) to the function. "Formula methods are really inefficient at this time for large dimensional data but are fantastically convenient. There are some good reasons to not use formulas, such as functions that do not use a design matrix (e.g., party::cforest()) or need factors to be handled in a non-standard way (e.g., Cubist::cubist())."
- "Don't require a test set when model building."
- If not all variables are used in the resulting model, allow the required subset of variables to be provided for stats::predict() and not all the original variables, and avoid referencing variables by position rather than name.

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

27.20 Style Resources

20200105 There are many style guides available and the guidelines here are generally consistent and overlap considerably with many others. In this chapter I aim to capture the motivation for each choice. My style choices are based on over 30 years of programming in very many different languages. Some elements of style are personal preference and others have very solid foundations. Unfortunately in reading some style guides the choices made are not always explained and without the motivation we do not really have a basis to choose or to debate.

The canonical style guide for the tidyverse is Hadley Wickham's style guide at https://style.tidyverse.org/. Other guidelines can be found at Google and Hadley Wickham's older guide and . These are similar though with idiosyncratic differences. Also see Wikipedia for an excellent summary of many styles.

Rasmus Bth, in , reviews naming conventions used in R, finding that the initial lower case capitalised word scheme for functions was the most popular, and dot separated names for arguments similarly. We are however seeing a migration away from the dot in variable names as it is also used as a class separator for object oriented coding. Using the underscore is now preferred.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.

References

Aragon, Tomas J. 2020. *Epitools: Epidemiology Tools*. https://CRAN.R-project.org/package=epitools.

Bache, Stefan Milton, and Hadley Wickham. 2020. *Magrittr: A Forward-Pipe Operator for r*. https://CRAN.R-project.org/package=magrittr.

Bilisoly, Roger. 2008. *Practical Text Mining with Perl*. Wiley Series on Methods and Applications in Data Mining. Wiley. http://books.google.com.au/books?id=YkMFVbsrdzkC.

Breiman, Leo, Adele Cutler, Andy Liaw, and Matthew Wiener. 2018. *randomForest: Breiman and Cutler's Random Forests for Classification and Regression*. https://www.stat.berkeley.edu/~breiman/RandomForests/.

Dahl, David B., David Scott, Charles Roosen, Arni Magnusson, and Jonathan Swinton. 2019. *Xtable: Export Tables to LaTeX or HTML*. http://xtable.r-forge.r-project.org/.

Dragulescu, Adrian, and Cole Arendt. 2020. *Xlsx: Read, Write, Format Excel 2007 and Excel 97/2000/XP/2003 Files*. https://github.com/colearendt/xlsx.

Durant, Will. 1926. *The Story of Philosophy*. 2012th ed. Simon; Schuster.

Feinerer, Ingo, and Kurt Hornik. 2020. *Tm: Text Mining Package*. http://tm.r-forge.r-project.org/.

Fellows, Ian. 2018. *Wordcloud: Word Clouds*. https://CRAN.R-project.org/package=wordcloud.

Firke, Sam. 2021. *Janitor: Simple Tools for Examining and Cleaning Dirty Data*. https://github.com/sfirke/janitor.

Gagolewski, Marek, Bartek Tartanus, and others; IBM, Unicode, Inc., and others. 2021. *Stringi: Character String Processing Facilities*. https://CRAN.R-project.org/package=stringi.

Gandrud, Christopher. 2014. *Repreoducible Research with R and RStdio*. The R Series. CRC Press.

Hansen, Kasper Daniel, Jeff Gentry, Li Long, Robert Gentleman, Seth Falcon, Florian Hahne, and Deepayan Sarkar. 2018. *Rgraphviz: Provides Plotting Capabilities for r Graph Objects*.

Harrell, Frank E, Jr. 2021. *Hmisc: Harrell Miscellaneous*. https://CRAN.R-project.org/package=Hmisc.

Hester, Jim. 2020. *Glue: Interpreted String Literals*. https://CRAN.R-project.org/package=glue.

Hornik, Kurt. 2020. *RWeka: R/Weka Interface*. https://CRAN.R-project.org/package=RWeka.

Hothorn, Torsten, Kurt Hornik, Carolin Strobl, and Achim Zeileis. 2021. *Party: A Laboratory for Recursive Partytioning*. http://party.R-forge.R-project.org.

Hothorn, Torsten, and Achim Zeileis. 2021. *Partykit: A Toolkit for Recursive Partytioning*. http://partykit.r-forge.r-project.org/partykit/.

Kaiser, Sebastian, Rodrigo Santamaria, Tatsiana Khamiakova, Martin Sill, Roberto Theron, Luis Quintales, Friedrich Leisch, Ewoud De Troyer, and Sami Leon. 2021. *Biclust: BiCluster Algorithms*. https://CRAN.R-project.org/package=biclust.

Keitt, Tim. 2012. *colorRamps: Builds Color Tables*. https://CRAN.R-project.org/package=colorRamps.

Knuth, Donald E. 1984. "Literate Programming." *The Computer Journal (British Computer Society)* 27 (2): 97–111. http://www.literateprogramming.com/knuthweb.pdf.

Kuhn, Max, and Ross Quinlan. 2021. *C50: C5.0 Decision Trees and Rule-Based Models*. https://topepo.github.io/C5.0/.

Liu, Zhongxin, Xin Xia, Christoph Treude, David Lo, and Shanping Li. 2019. "Automatic Generation of Pull Request Descriptions." *2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, November. https://doi.org/10.1109/ase.2019.00026.

Milborrow, Stephen. 2020. *Rpart.plot: Plot Rpart Models: An Enhanced Version of Plot.rpart*. http://www.milbo.org/rpart-plot/index.html.

Neuwirth, Erich. 2014. *RColorBrewer: ColorBrewer Palettes*. https://CRAN.R-project.org/package=RColorBrewer.

Ooms, Jeroen. 2021. *Writexl: Export Data Frames to Excel Xlsx Format*. https://CRAN.R-project.org/package=writexl.

Qin, D., X. Zhou, L. Chen, G. Huang, and Y. Zhang. 2020. "Dynamic Connection-Based Social Group Recommendation." *IEEE Transactions on Knowledge and Data Engineering* 32 (3): 453–67.

R Core Team. 2021. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Rinker, Tyler. 2018. *qdapDictionaries: Dictionaries and Word Lists for the Qdap Package*. http://trinker.github.com/qdapDictionaries/.

———. 2020a. *Qdap: Bridging the Gap Between Qualitative Data and Quantitative Analysis*. http://trinker.github.io/qdap/.

———. 2020b. *Wakefield: Generate Random Data Sets*. https://github.com/trinker/wakefield.

Ripley, Brian. 2021. *Nnet: Feed-Forward Neural Networks and Multinomial Log-Linear Models*. http://www.stats.ox.ac.uk/pub/MASS4/.

Romanski, Piotr, Lars Kotthoff, and Patrick Schratz. 2021. *FSelector: Selecting Attributes*. https://github.com/larskotthoff/fselector.

Sarkar, Deepayan. 2021. *Lattice: Trellis Graphics for r*. http://lattice.r-forge.r-project.org/.

Schloerke, Barret, Di Cook, Joseph Larmarange, Francois Briatte, Moritz Marbach, Edwin Thoen, Amos Elberg, and Jason Crowley. 2021. *GGally: Extension to Ggplot2*. https://CRAN.R-project.org/package=GGally.

Sing, Tobias, Oliver Sander, Niko Beerenwinkel, and Thomas Lengauer. 2020. *ROCR: Visualizing the Performance of Scoring Classifiers*. http://ipa-tys.github.io/ROCR/.

Soetaert, Karline. 2020. *Diagram: Functions for Visualising Simple Graphs (Networks), Plotting Flow Diagrams*. https://CRAN.R-project.org/package=diagram.

Spinu, Vitalie, Garrett Grolemund, and Hadley Wickham. 2021. *Lubridate: Make Dealing with Dates a Little Easier*. https://CRAN.R-project.org/package=lubridate.

Therneau, Terry, and Beth Atkinson. 2019. *Rpart: Recursive Partitioning and Regression Trees*. https://CRAN.R-project.org/package=rpart.

Velten, Kai. 2009. *Mathematical Modeling and Simulation*. Wiley.

Wickham, Hadley. 2014. *Advanced R*. Chapman & Hall/CRC the r Series. Chapman & Hall.

———. 2019a. *Lobstr: Visualize r Data Structures with Trees*. https://github.com/r-lib/lobstr.

———. 2019b. *Stringr: Simple, Consistent Wrappers for Common String Operations*. https://CRAN.R-project.org/package=stringr.

———. 2021. *Tidyr: Tidy Messy Data*. https://CRAN.R-project.org/package=tidyr.

Wickham, Hadley, and Jennifer Bryan. 2019. *Readxl: Read Excel Files*. https://CRAN.R-project.org/package=readxl.

Wickham, Hadley, Winston Chang, Lionel Henry, Thomas Lin Pedersen, Kohske Takahashi, Claus Wilke, Kara Woo, Hiroaki Yutani, and Dewey Dunnington. 2020. *Ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics*. https://CRAN.R-project.org/package=ggplot2.

Wickham, Hadley, Romain François, Lionel Henry, and Kirill Müller. 2021. *Dplyr: A Grammar of Data Manipulation*. https://CRAN.R-project.org/package=dplyr.

Wickham, Hadley, and Jim Hester. 2020. *Readr: Read Rectangular Text Data*. https://CRAN.R-project.org/package=readr.

Wickham, Hadley, and Dana Seidel. 2020. *Scales: Scale Functions for Visualization*. https://CRAN.R-project.org/package=scales.

Williams, Graham. 2020. *Rattle: Graphical User Interface for Data Science in r*. https://rattle.togaware.com/.

Williams, Graham J. 2011. *Data Mining with Rattle and R: The Art of Excavating Data for Knowledge Discovery.* Use R! New York: Springer.

Williams, Graham J. 2017. *The Essentials of Data Science: Knowledge Discovery Using r*. The r Series. CRC Press.

Xie, Yihui. 2014. *Dynamic Documents with R and Knitr*. The R Series. CRC Press.

———. 2021. *Knitr: A General-Purpose Package for Dynamic Report Generation in r*. https://yihui.org/knitr/.

---

*Your donation will support ongoing development and give you access to the **PDF version of this book**. Desktop Survival Guides include Data Science, GNU/Linux, and MLHub. Books available on Amazon include Data Mining with Rattle and Essentials of Data Science. Popular open source software includes rattle, wajig, and mlhub. Hosted by Togaware, a pioneer of free and open source software since 1984.*
Copyright © 1995-2021 Graham.Williams@togaware.com Creative Commons Attribution-ShareAlike 4.0.