# 1.R_Data_Cleanup

Kim Wong

04/01/2021

### Material taken from

**Graham J. Williams. 2017. The Essentials of Data Science: Knowledge Discovery Using R (1st ed.). Chapman & Hall/CRC.**

### Load required packages

```
library(tidyverse)    # ggplot2, tibble, tidyr, readr, purr, dplyr
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.1.0     v dplyr   1.0.5
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(rattle)       # comcat(), weatherAUS, normVarNames().
```

```
## Loading required package: bitops
```

```
## Rattle: A free graphical interface for data science with R.
## Version 5.4.0 Copyright (c) 2006-2020 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.
```

```
library(magrittr)     # Pipe operator %>% %<>% %T>% equals().
```

```
##
## Attaching package: 'magrittr'
```

```
## The following object is masked from 'package:purrr':
##
##     set_names
```

```
## The following object is masked from 'package:tidyr':
##
##     extract
```

```
library(lubridate)    # Dates and time.
```

```
##
```

```
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(stringi)      # String concat operator %s+%.
library(stringr)      # String manipulation: str_replace().
library(randomForest) # Impute missing values with na.roughfix()
```

```
## randomForest 4.6-14

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:rattle':
##
##     importance

## The following object is masked from 'package:dplyr':
##
##     combine

## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
library(FSelector)    # Feature selection: information.gain().
library(scales)       # Include commas in numbers.
```

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

## The following object is masked from 'package:purrr':
##
##     discard

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

```
library(xtable)       # Generate LaTeX tables.
```

## Location of datafile from web; load CSV file

```
##dspath <- "http://rattle.togaware.com/weatherAUS.csv"
dspath <- "https://rattle.togaware.com/weatherAUS.csv"
weatherAUS <- read_csv(file=dspath, guess_max = 8888)
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   .default = col_double(),
##   Date = col_date(format = ""),
##   Location = col_character(),
##   WindGustDir = col_character(),
##   WindDir9am = col_character(),
```

```
##   WindDir3pm = col_character(),
##   RainToday = col_character(),
##   RainTomorrow = col_character()
## )
## i Use `spec()` for the full column specifications.
```

```
#weatherAUS <- rattle::weatherAUS
```

### Assign original dataset to generic variable

```
ds <- weatherAUS
ds
```

```
## # A tibble: 191,431 x 24
##    Date       Location MinTemp MaxTemp Rainfall Evaporation Sunshine WindGustDir
##    <date>     <chr>      <dbl>   <dbl>    <dbl>       <dbl>    <dbl> <chr>
##  1 2008-12-01 Albury      13.4    22.9      0.6          NA       NA W
##  2 2008-12-02 Albury       7.4    25.1      0            NA       NA WNW
##  3 2008-12-03 Albury      12.9    25.7      0            NA       NA WSW
##  4 2008-12-04 Albury       9.2    28        0            NA       NA NE
##  5 2008-12-05 Albury      17.5    32.3      1            NA       NA W
##  6 2008-12-06 Albury      14.6    29.7      0.2          NA       NA WNW
##  7 2008-12-07 Albury      14.3    25        0            NA       NA W
##  8 2008-12-08 Albury       7.7    26.7      0            NA       NA W
##  9 2008-12-09 Albury       9.7    31.9      0            NA       NA NNW
## 10 2008-12-10 Albury      13.1    30.1      1.4          NA       NA W
## # ... with 191,421 more rows, and 16 more variables: 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>
```

### dimensions of data frame

```
dim(ds) %>% comcat()
```

```
## 191,431 24
```

```
nrow(ds) %>% comcat()
```

```
## 191,431
```

```
ncol(ds) %>% comcat()
```

```
## 24
```

### Use dplyr::glimpse to get a glimpse of the data

```
glimpse(ds)
```

```
## Rows: 191,431
## 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   <chr> "W", "WNW", "WSW", "NE", "W", "WNW", "W", "W", "NNW", "W~
## $ WindGustSpeed <dbl> 44, 44, 46, 24, 41, 56, 50, 35, 80, 28, 30, 31, 61, 44, ~
## $ WindDir9am    <chr> "W", "NNW", "W", "SE", "ENE", "W", "SW", "SSE", "SE", "S~
## $ WindDir3pm    <chr> "WNW", "WSW", "WSW", "E", "NW", "W", "W", "W", "NW", "SS~
## $ 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   <dbl> 71, 44, 38, 45, 82, 55, 49, 48, 42, 58, 48, 89, 76, 65, ~
## $ Humidity3pm   <dbl> 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      <dbl> 8, NA, NA, NA, 7, NA, 1, NA, NA, NA, NA, 8, 8, NA, NA, 0~
## $ Cloud3pm      <dbl> 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     <chr> "No", "No", "No", "No", "No", "No", "No", "No", "No", "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  <chr> "No", "No", "No", "No", "No", "No", "No", "No", "Yes", "~
```

### Output only variable 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"
```

### Normalize variable names

```
names(ds) <- normVarNames(names(ds))
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"
```

### Use head and tail to glimpse the top and bottom rows of data

```
head(ds)
```

```
## # A tibble: 6 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
## # ... with 17 more variables: wind_gust_dir <chr>, wind_gust_speed <dbl>,
## #   wind_dir_9am <chr>, wind_dir_3pm <chr>, 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 <chr>, risk_mm <dbl>,
## #   rain_tomorrow <chr>
```

```
tail(ds)
```

```
## # A tibble: 6 x 24
##   date       location min_temp max_temp rainfall evaporation sunshine
##   <date>     <chr>       <dbl>    <dbl>    <dbl>       <dbl>    <dbl>
## 1 2021-02-22 Uluru        25.3     37.4        0          NA       NA
## 2 2021-02-23 Uluru        23       33.9        0          NA       NA
## 3 2021-02-24 Uluru        19.7     33          0          NA       NA
## 4 2021-02-25 Uluru        16.9     33.4        0          NA       NA
## 5 2021-02-26 Uluru        17.4     34.9        0          NA       NA
## 6 2021-02-27 Uluru        18.8     37.3        0          NA       NA
## # ... with 17 more variables: wind_gust_dir <chr>, wind_gust_speed <dbl>,
## #   wind_dir_9am <chr>, wind_dir_3pm <chr>, 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 <chr>, risk_mm <dbl>,
## #   rain_tomorrow <chr>
```

### Randomly sample 10 columns of data

```
set.seed(42)
sample_n(ds, size = 10)
```

```
## # A tibble: 10 x 24
##    date       location      min_temp max_temp rainfall evaporation sunshine
##    <date>     <chr>            <dbl>    <dbl>    <dbl>       <dbl>    <dbl>
##  1 2012-07-08 Canberra          -5.4     13.2      0          NA        7.6
##  2 2017-07-16 Williamtown        8.4     17.6      0          NA       NA
##  3 2011-10-28 Portland          12.7     23.1      1.4         4.2      6.9
##  4 2015-05-31 Ballarat           7       11.1      1          NA       NA
##  5 2015-08-06 SydneyAirport      6.9     15.4      0           3.8      8.5
##  6 2020-12-23 Launceston        14.3     24        8          NA       NA
##  7 2013-05-22 Witchcliffe        7.9     19.3      0.8        NA       NA
##  8 2013-03-16 Canberra          15.6     28.3      0           4.8     NA
##  9 2012-03-02 Ballarat          12.1     17.9      7.6        NA       NA
## 10 2020-06-22 Portland           5.6     12.8      6.8        NA       NA
## # ... with 17 more variables: wind_gust_dir <chr>, wind_gust_speed <dbl>,
## #   wind_dir_9am <chr>, wind_dir_3pm <chr>, 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 <chr>, risk_mm <dbl>,
## #   rain_tomorrow <chr>
```

## Data Cleaning

### output the unique cities in Australia

```
ds$location %>% unique() %>% length()
```

```
## [1] 49
```

### get the distribution of observations for cities

```
ds$location %<>% as.factor()
table(ds$location)
```

```
##
##         Adelaide           Albany           Albury     AliceSprings
##             3924             3983             3984             3984
##    BadgerysCreek         Ballarat          Bendigo         Brisbane
##             3936             3984             3975             4137
##           Cairns         Canberra            Cobar     CoffsHarbour
##             3984             4380             3953             3953
##         Dartmoor           Darwin        GoldCoast           Hobart
##             3953             4137             3984             4137
##        Katherine       Launceston       Melbourne MelbourneAirport
##             2522             3984             4137             3953
##          Mildura            Moree     MountGambier      MountGinini
##             3953             3953             3983             3984
##        Newcastle            Nhil        NorahHead    NorfolkIsland
##             3984             2522             3948             3953
##        Nuriootpa       PearceRAAF          Penrith            Perth
##             3952             3952             3983             4136
##     PerthAirport         Portland         Richmond             Sale
##             3952             3953             3953             3953
##       SalmonGums           Sydney    SydneyAirport       Townsville
##             3906             4288             3953             3984
##      Tuggeranong            Uluru       WaggaWagga          Walpole
##             3983             2522             3953             3949
##         Watsonia      Williamtown      Witchcliffe       Wollongong
##             3953             3953             3952             3984
##          Woomera
##             3953
```

### dplyr::selec() used to find variables with a particular string

```
ds %>% select(starts_with("rain_")) %>% sapply(table)
```

```
##     rain_today rain_tomorrow
## No      146077        146080
## Yes      40317         40313
```

### find variable names with rain_

```
ds %>% select(starts_with("rain_")) %>% names() %T>% print() -> vnames
```

```
## [1] "rain_today"    "rain_tomorrow"
```

```
ds[vnames] %>% sapply(class)
```

```
##    rain_today rain_tomorrow
##   "character"   "character"
```

**convert variables from character to factor class**

```
ds[vnames] %<>% lapply(factor)
ds[vnames] %>% sapply(class)
```

```
##    rain_today rain_tomorrow
##      "factor"      "factor"
```

**Verify that the distribution has not changed**

```
ds %>% select(starts_with("rain_")) %>% sapply(table)
```

```
##     rain_today rain_tomorrow
## No      146077        146080
## Yes      40317         40313
```

## Review the distribution of observations across levels

```
ds %>% select(contains("_dir")) %>% sapply(table)
```

```
##     wind_gust_dir wind_dir_9am wind_dir_3pm
## E           12007        12159        10906
## ENE         10730        10359        10357
## ESE          9850        10392        11179
## N           11908        15053        11362
## NE           9434        10058        10950
## NNE          8710        10723         8806
## NNW          8653        10226        10186
## NW          10516        11309        11097
## S           11937        11313        12711
## SE          12240        12320        13903
## SSE         11840        11924        11979
## SSW         11739        10072        10732
## SW          11604        10985        12005
## W           13030        10934        13283
## WNW         10786         9826        11675
## WSW         11970         9040        12571
```

## 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"
```

**Confirm that these variables are of type character**

```
ds[vnames] %>% sapply(class)
```

```
## wind_gust_dir  wind_dir_9am  wind_dir_3pm
##   "character"   "character"   "character"
```

**set ordered compass directions**

```
compass <- c("N", "NNE", "NE", "ENE",
             "E", "ESE", "SE", "SSE",
             "S", "SSW", "SW", "WSW",
             "W", "WNW", "NW", "NNW")
```

**use ordered compass directions for factor levels**

```
ds[vnames] %<>% lapply(factor, levels=compass, ordered=TRUE) %>% data.frame() %>% tbl_df() %T>% {sapply
```

```
## Warning: `tbl_df()` was deprecated in dplyr 1.0.0.
## Please use `tibble::as_tibble()` instead.
```

```
##      wind_gust_dir wind_dir_9am wind_dir_3pm
## [1,] "ordered"     "ordered"    "ordered"
## [2,] "factor"      "factor"     "factor"
```

**Verify that the distribution has not changed**

```
ds %>% select(contains("dir")) %>% sapply(table)
```

```
##     wind_gust_dir wind_dir_9am wind_dir_3pm
## N           11908        15053        11362
## NNE          8710        10723         8806
## NE           9434        10058        10950
## ENE         10730        10359        10357
## E           12007        12159        10906
## ESE          9850        10392        11179
## SE          12240        12320        13903
## SSE         11840        11924        11979
## S           11937        11313        12711
## SSW         11739        10072        10732
## SW          11604        10985        12005
## WSW         11970         9040        12571
## W           13030        10934        13283
## WNW         10786         9826        11675
## NW          10516        11309        11097
## NNW          8653        10226        10186
```

## Evaporation and Sunshine

```
cvars <- c("evaporation","sunshine")
head(ds[cvars])
```

```
## # A tibble: 6 x 2
##   evaporation sunshine
##         <dbl>    <dbl>
## 1          NA       NA
## 2          NA       NA
## 3          NA       NA
```

```
## 4          NA       NA
## 5          NA       NA
## 6          NA       NA
```

```
sample_n(ds[c("evaporation","sunshine")], 10)
```

```
## # A tibble: 10 x 2
##    evaporation sunshine
##          <dbl>    <dbl>
##  1        12       11.3
##  2        NA       NA
##  3        NA       NA
##  4        NA       NA
##  5        NA       NA
##  6         7.2     10.8
##  7         4.6     10.2
##  8         3.4      6.4
##  9        NA       NA
## 10         3.6      2.5
```

```
ds[cvars] %>% sapply(class)
```

```
## evaporation    sunshine
##   "numeric"   "numeric"
```

### Categoric

```
ds %>% sapply(is.factor) %>% which() -> catc
```

```
glimpse(ds[catc])
```

```
## Rows: 191,431
## Columns: 6
## $ location      <fct> Albury, Albury, Albury, Albury, Albury, Albury, Albury, ~
## $ wind_gust_dir <ord> W, WNW, WSW, NE, W, WNW, W, W, NNW, W, N, NNE, W, SW, NA~
## $ 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, S~
## $ rain_today    <fct> No, No, No, No, No, No, No, No, No, Yes, No, Yes, Yes, Y~
## $ rain_tomorrow <fct> No, No, No, No, No, No, No, No, Yes, No, Yes, Yes, Yes, ~
```

```
for (v in catc) levels(ds[[v]]) %<>% normVarNames()
```

```
glimpse(ds[catc])
```

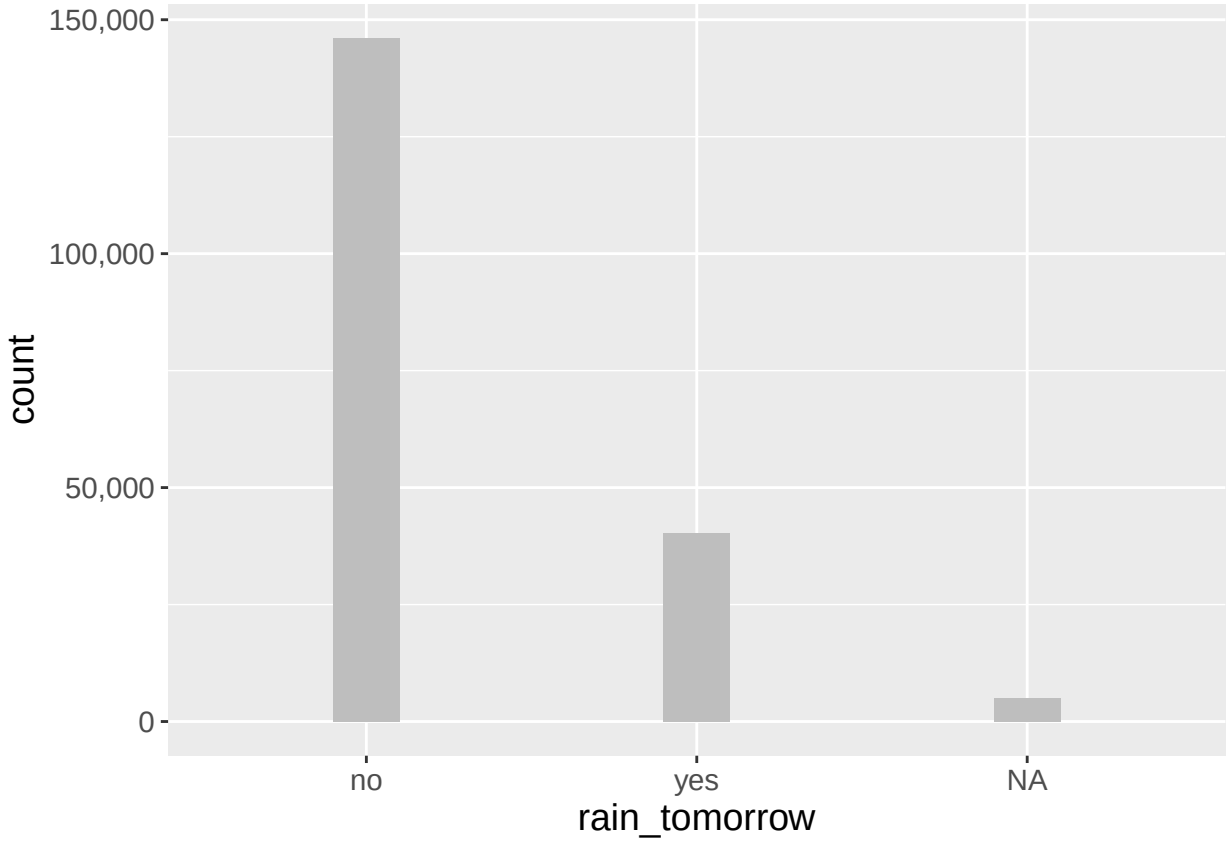
```
## Rows: 191,431
## Columns: 6
## $ location      <fct> albury, albury, albury, albury, albury, albury, albury, ~
## $ wind_gust_dir <ord> w, wnw, wsw, ne, w, wnw, w, w, nnw, w, n, nne, w, sw, NA~
## $ wind_dir_9am  <ord> w, nnw, w, se, ene, w, sw, sse, se, s, sse, ne, nnw, w, ~
## $ wind_dir_3pm  <ord> wnw, wsw, wsw, e, nw, w, w, w, nw, sse, ese, ene, nnw, s~
## $ rain_today    <fct> no, no, no, no, no, no, no, no, no, yes, no, yes, yes, y~
## $ rain_tomorrow <fct> no, no, no, no, no, no, no, no, yes, no, yes, yes, yes, ~
```

### Prepare target and make sure it is a factor type

```
target <- "rain_tomorrow"
ds[[target]] %<>% as.factor()
ds[target] %>% table()
```

```
## .
##     no    yes
## 146080  40313
```

```
ds %>%
  ggplot(aes_string(x=target)) +
  geom_bar(width=0.2, fill="grey") +
  scale_y_continuous(labels=comma) +
  theme(text=element_text(size=14))
```




### Parititioning the data set into dependent and independent variables

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

```
##  [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"
```

**What we wish to predict is if it will "rain tomorrow" given historical weather data. The variable "rain_tomorrow" is therefore the target that depends on the other data. One convention is to place the target in front of the other data.**

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

```
##  [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"
```

**risk_mm records the amount of rain that fell tomorrow; it measures the risk of the outcome we are predicting. Therefore, risk_mm is an output variable. Also, variables date and location are identifiers; these variables are not used as independent variables for building predictive models.**

```
risk <- "risk_mm"
id <- c("date", "location")
```

### Identifying irrelevant variables within a dataset

**Ignore identifiers and risk variables**

```
union(id, risk) -> ignore
ignore
```

```
## [1] "date"     "location" "risk_mm"
```

### Helper function to count unique entries

```
count_unique <- function(x) {length(unique(x))}
```

```
ds[vars] %>% sapply(count_unique) %>% equals(nrow(ds)) %>% which() %>% names() %T>% print() -> ids
```

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

**Let's just look at the data for Sydney.**

```
ds_sydney <- filter(ds, location=="sydney")
ds_sydney[vars] %>% sapply(count_unique) %>% equals(nrow(ds_sydney)) %>% which () %>% names()
```

```
## [1] "date"
```

**Helper function to count the number of missing values**

```
count_na <- function(x) {sum(is.na(x))}
```

**Check for variables with completely missing data**

```
ds[vars] %>% sapply(count_na) %>% equals(nrow(ds)) %>% which () %>% names() %T>% print() -> missing
```

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

**Let's just look at the data for Sydney.**

```
ds_sydney <- filter(ds, location=="sydney")
ds_sydney[vars] %>% sapply(count_na) %>% equals(nrow(ds_sydney)) %>% which () %>% names()
```

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

**Let's just look at the data for Albury**

```
ds_albury <- filter(ds, location=="albury")
ds_albury[vars] %>% sapply(count_na) %>% equals(nrow(ds_albury)) %>% which () %>% names()
```

```
## [1] "evaporation" "sunshine"
```

## Flag variable will many missing entries

```
missing.threshold <- 0.8
ds[vars] %>% sapply(count_na) %>% '>'(missing.threshold*nrow(ds)) %>% which () %>% names() %T>% print()
```

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

## Check Sydney

```
missing.threshold <- 1.0
ds_sydney[vars] %>% sapply(count_na) %>% '>'(missing.threshold*nrow(ds_sydney)) %>% which () %>% names()
```

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

## Flag variables with too many factor levels

```
count_levels <- function(x){ds %>% extract2(x) %>% levels() %>% length()}
```

```
levels.threshold <- 16
ds[vars] %>% sapply(is.factor) %>% which() %>% names() %>% sapply(count_levels) %>% '>='(levels.thresho
```

```
## [1] "location"      "wind_gust_dir" "wind_dir_9am"  "wind_dir_3pm"
```

## Flag constants

```
all_same <- function(x){all(x==x[1L])}
```

```
ds[vars] %>% sapply(all_same) %>% which() %>% names() %T>% print() -> constants
```

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

## Flag correlated variables

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

```
##  [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"
```

```
ds[numc] %>%
  cor(use="complete.obs") %>%
  ifelse(upper.tri(., diag=TRUE), NA, .) %>%
  abs() %>%
  data.frame() %>%
  tbl_df() %>%
  set_colnames(numc) %>%
  mutate(var1=numc) %>%
  gather(var2, cor, -var1) %>%
  na.omit() %>%
  arrange(-abs(cor)) %T>%
  print() ->
mc
```

```
## # A tibble: 120 x 3
##    var1           var2              cor
##    <chr>          <chr>           <dbl>
##  1 temp_3pm       max_temp        0.984
##  2 pressure_3pm   pressure_9am    0.962
##  3 temp_9am       min_temp        0.908
##  4 temp_9am       max_temp        0.894
##  5 temp_3pm       temp_9am        0.870
##  6 max_temp       min_temp        0.753
##  7 temp_3pm       min_temp        0.730
##  8 cloud_3pm      sunshine        0.700
##  9 wind_speed_3pm wind_gust_speed 0.690
## 10 humidity_3pm   humidity_9am    0.679
## # ... with 110 more rows
```

### Added correlated variables to ignore set

```
correlated <- c("temp_3pm", "pressure_3pm", "temp_9am")
ignore <- union(ignore,correlated)
ignore
```

```
## [1] "date"         "location"     "risk_mm"      "temp_3pm"     "pressure_3pm"
## [6] "temp_9am"
```

### Remove ignore variables from full set

```
length(vars)
```

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

```
vars %<>% setdiff(ignore) %T>% print()
```

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

```
length(vars)
```

```
## [1] 18
```

### Construct formula for modeling

```
form <- formula(target %s+% " ~ .") %T>% print()
```

```
## rain_tomorrow ~ .
```

### Identify attribute subset using correlation and entropy measures. FSelector::cfs

```
cfs(form, ds[vars])
```

```
## [1] "rainfall"     "sunshine"     "humidity_3pm" "cloud_3pm"    "rain_today"
```

### Use information gain to identify variables of importance. FSelector::information.gain

```
information.gain(form, ds[vars]) %>%
  rownames_to_column("variable") %>%
  arrange(-attr_importance)
```

```
##           variable attr_importance
## 1     humidity_3pm     0.109083345
## 2         rainfall     0.057504485
## 3         sunshine     0.049311931
## 4        cloud_3pm     0.047070812
## 5       rain_today     0.046236942
## 6     humidity_9am     0.037681166
## 7        cloud_9am     0.031954207
## 8     pressure_9am     0.027694874
## 9  wind_gust_speed     0.026462228
## 10        max_temp     0.013995663
## 11    wind_dir_9am     0.008661298
## 12   wind_gust_dir     0.005841446
## 13        min_temp     0.005730390
## 14  wind_speed_3pm     0.005054382
## 15     evaporation     0.004797909
## 16    wind_dir_3pm     0.004688325
## 17  wind_speed_9am     0.004118976
```

### Identify and remove observations with missing target

```
dim(ds)
```

```
## [1] 191431     24
```

```
ds %>% extract2(target) %>% is.na() -> missing_target
sum(missing_target)
```

```
## [1] 5038
```

```
ds %<>% filter(!missing_target)
dim(ds)
```

```
## [1] 186393     24
```

### Remove observations with missing entries

```
ods <- ds

omit <- NULL

ds[vars] %>% nrow()
```

```
## [1] 186393
```

```
ds[vars] %>% is.na() %>% sum() %>% comcat()
```

```
## 442,254
```

```
mo <- attr(na.omit(ds[vars]), "na.action")
```

```
omit <- union(omit,mo)
```

```
if (length(omit)) ds <- ds[-omit,]
```

```
ds[vars] %>% nrow() %>% comcat()
```

```
## 64,248
```

```
ds[vars] %>% is.na() %>% sum() %>% comcat()
```

```
## 0
```

```
ds <- ods
omit <- NULL
```

### Augment data with derived features

```
ds %<>%
  mutate(year   = factor(format(date,"%Y")),
         season = format(ds$date, "%m") %>%
                  as.integer() %>%
                  sapply(function(x)
                    switch(x,
                           "summer", "summer", "autumn",
                           "autumn", "autumn", "winter",
                           "winter", "winter", "spring",
                           "spring", "spring", "summer")) %>%
                  as.factor()) %T>%
                  {select(., date, year, season) %>% sample_n(10) %>% print()}
```

```
## # A tibble: 10 x 3
##    date       year  season
##    <date>     <fct> <fct>
##  1 2018-12-13 2018  summer
##  2 2017-05-19 2017  autumn
##  3 2014-05-30 2014  autumn
##  4 2017-09-09 2017  spring
##  5 2020-05-31 2020  autumn
##  6 2018-10-13 2018  spring
##  7 2010-03-22 2010  autumn
##  8 2011-07-26 2011  winter
##  9 2011-03-08 2011  autumn
```

```
## 10 2012-10-29 2012  spring
```

```
vars %<>% c("season")
id %<>% c("year")
```

### Augment data with model-generated features

```
set.seed(4242)
nclust <- 5

ds[c("location",numc)] %>%
  group_by(location) %>%
  summarise_all(funs(mean(.,na.rm=TRUE))) %T>%
  {locations <<- .$location} %>%
  select(-location) %>%
  sapply(function(x) ifelse(is.nan(x),0,x)) %>%
  as.data.frame() %>%
  sapply(scale) %>%
  kmeans(nclust) %T>%
  print() %>%
  extract2("cluster") ->
cluster
```

```
## Warning: `funs()` was deprecated in dplyr 0.8.0.
## Please use a list of either functions or lambdas:
##
##   # Simple named list:
##   list(mean = mean, median = median)
##
##   # Auto named with `tibble::lst()`:
##   tibble::lst(mean, median)
##
##   # Using lambdas
##   list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))
```

```
## K-means clustering with 5 clusters of sizes 10, 9, 3, 11, 16
##
## Cluster means:
##     min_temp   max_temp   rainfall evaporation   sunshine wind_gust_speed
## 1  0.1196161  0.4853625 -0.6424759  0.73880180  1.0104693     -0.02573600
## 2  1.2893477  0.6091285  1.4483272  0.45179727  0.5282667      0.32272082
## 3  0.9474763  2.0498609 -0.6937602  0.64484113 -0.3918250      0.26234736
## 4 -0.5028720 -0.4664017  0.0638168 -1.16986478 -1.0675780     -0.33369160
## 5 -0.6319454 -0.7096841 -0.3269306 -0.03251277 -0.1212663      0.01477738
##   wind_speed_9am wind_speed_3pm humidity_9am humidity_3pm pressure_9am
## 1     0.02126014     -0.4732464  -0.87832729   -0.9388059    0.2975726
## 2     0.66328971      1.1023842  -0.02789297    0.6618238    0.2917131
## 3     0.26706045     -0.3864947  -2.27247110   -1.8902231    0.2873585
## 4    -0.58174898     -0.6113393   0.61408776    0.3863937   -1.0178557
## 5    -0.03650946      0.1684514   0.56854735    0.3032489    0.2958245
##   pressure_3pm   cloud_9am   cloud_3pm   temp_9am   temp_3pm
## 1    0.2967963 -0.14225644 -0.04047112  0.2504660  0.5017555
## 2    0.2908433  0.24621523  0.22235357  1.1213116  0.5836234
## 3    0.2816821  0.07681595  0.10224201  1.4563930  2.0830993
## 4   -1.0171344 -1.41328121 -1.45286859 -0.4962396 -0.4886922
```

```
## 5    0.2973675  0.90764205  0.87989734 -0.7191880 -0.6964907
##
## Clustering vector:
##  [1] 1 5 5 3 4 5 5 1 2 5 1 2 4 2 2 5 3 5 5 5 1 1 5 4 4 4 4 2 5 1 4 1 1 5 5 5 4 2
## [39] 2 2 4 3 1 4 5 2 4 5 1
##
## Within cluster sum of squares by cluster:
## [1]  48.53711  56.46908  27.87802 161.05342  95.28448
##  (between_SS / total_SS =  49.3 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```
head(cluster)
```

```
## [1] 1 5 5 3 4 5
```

```
names(cluster) <- locations

ds %<>% mutate(cluster="area" %>% paste0(cluster[ds$location]) %>% as.factor)

ds %>% select(location, cluster) %>% sample_n(10)
```

```
## # A tibble: 10 x 2
##    location          cluster
##    <fct>             <fct>
##  1 albany            area5
##  2 katherine         area3
##  3 coffs_harbour     area2
##  4 norfolk_island    area2
##  5 cobar             area1
##  6 alice_springs     area3
##  7 cairns            area2
##  8 melbourne_airport area5
##  9 gold_coast        area2
## 10 mount_ginini      area4
```

```
vars %<>% c("cluster")
```

### Sanity check of clusters

```
cluster[levels(ds$location)] %>% sort()
```

```
##          adelaide          brisbane             cobar           mildura
##                 1                 1                 1                 1
##             moree       pearce_raaf             perth     perth_airport
##                 1                 1                 1                 1
##       wagga_wagga           woomera            cairns     coffs_harbour
##                 1                 1                 2                 2
##            darwin        gold_coast    norfolk_island            sydney
##                 2                 2                 2                 2
##    sydney_airport        townsville       williamtown     alice_springs
##                 2                 2                 2                 3
##         katherine             uluru    badgerys_creek          dartmoor
```

```
##                3                3                4                4
##     mount_ginini         newcastle             nhil       norah_head
##                4                4                4                4
##          penrith       salmon_gums       tuggeranong          walpole
##                4                4                4                4
##       witchcliffe           albany           albury         ballarat
##                4                5                5                5
##          bendigo          canberra           hobart       launceston
##                5                5                5                5
##         melbourne melbourne_airport     mount_gambier        nuriootpa
##                5                5                5                5
##          portland          richmond             sale         watsonia
##                5                5                5                5
##        wollongong
##                5
```

## Preparing Metadata

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

```
##  [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"     "season"         "cluster"
```

### Get integer index for each input variable in the original dataset

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

```
##  [1]  3  4  5  6  7  8  9 10 11 12 13 14 15 16 18 19 22 26 27
```

### Get the number of observations

```
ds %>% nrow() %T>% comcat() -> nobs
```

```
## 186,393
```

### Sanity check that the dimensions for various data subsets are correct

```
dim(ds) %>% comcat()
```

```
## 186,393 27
```

```
dim(ds[vars]) %>% comcat()
```

```
## 186,393 20
```

```
dim(ds[inputs]) %>% comcat()
```

```
## 186,393 19
```

```
dim(ds[inputi]) %>% comcat()
```

```
## 186,393 19
```

### Identify 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 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"
```

```
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] "year"            "season"          "cluster"
```

### Identify categoric variables by index

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

```
## [1]  8 10 11 22 26 27
```

### Idenify categoric variables by name

```
ds %>%
  names() %>%
  extract(cati) %T>%
```

```
  print() ->
numc
```

```
## [1] "wind_gust_dir" "wind_dir_9am"  "wind_dir_3pm"  "rain_today"
## [5] "season"        "cluster"
```

## Setup various components for model building

### Create the formula for a classification model

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

```
## 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 + cloud_9am + cloud_3pm + rain_today +
##     season + cluster
## <environment: 0x7f8de06de8e0>
```

### Generate training, validation and testing datasets

```
seed=424242
set.seed(seed)

nobs %>%
  sample(0.70*nobs) %T>%
  {length(.) %>% comcat()} %T>%
  {sort(.) %>% head(30) %>% print()} ->
train
```

```
## 130,475
##  [1]  1  2  5  7  8  9 10 11 13 15 16 17 18 19 20 21 22 23 25 26 27 30 31 32 34
## [26] 35 38 39 40 42
```

```
nobs %>%
  seq_len() %>%
  setdiff(train) %>%
  sample(0.15*nobs) %T>%
  {length(.) %>% comcat()} %T>%
  {sort(.) %>% head(15) %>% print()} ->
validate
```

```
## 27,958
##  [1]  3  4  6 24 29 36 45 63 80 83 86 87 95 98 99
```

```
nobs %>%
  seq_len() %>%
  setdiff(union(train, validate)) %T>%
  {length(.) %>% comcat()} %T>%
  {head(.) %>% print(15)} ->
test
```

```
## 27,960
## [1] 12 14 28 33 37 41
```

### Set up cache of values for target and risk variables

```
tr_target <- ds[train,][[target]] %T>% {head(.,20) %>% print()}
```

```
##  [1] no  no  no  no  no  no  yes yes yes no  no  no  no  no  no  no  no  no  no
## [20] no
## Levels: no yes
```

```
tr_risk <- ds[train,][[risk]] %T>% {head(., 20) %>% print()}
```

```
##  [1]  0.0  0.0  0.0  0.0  0.0  0.0 13.8 13.6  5.6  0.0  0.0  0.0  0.2  0.0  0.2
## [16]  0.0  0.0  0.0  0.0  0.0
```

```
va_target <- ds[validate,][[target]] %T>% {head(.,20) %>% print()}
```

```
##  [1] yes no  no  no  yes yes no  no  no  yes no  no  no  yes no  no  no  no  no
## [20] no
## Levels: no yes
```

```
va_risk <- ds[validate,][[risk]] %T>% {head(., 20) %>% print()}
```

```
##  [1] 15.4  0.0  0.4  0.0 35.6  7.2  0.4  0.0  0.0  1.4  0.0  0.0  0.0  8.4  0.0
## [16]  0.0  0.0  0.0  0.0  0.0
```

```
te_target <- ds[test,][[target]] %T>% {head(., 20) %>% print()}
```

```
##  [1] yes no  yes no  no  no  no  no  no  no  no  no  no  no  yes no  no  no  no
## [20] no
## Levels: no yes
```

```
te_risk <- ds[test,][[risk]] %T>% {head(., 20) %>% print()}
```

```
##  [1] 15.6  0.0  1.2  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  3.0
## [16]  0.0  0.0  0.0  0.0  0.0
```

## Save dataset

### specify folder name

```
fpath <- getwd() %>% print()
```

```
## [1] "/Users/kimwong/OneDrive - University of Pittsburgh/Documents/Kim F. Wong/CRC_Workshop/2021/Advan
```

### generate timestamp

```
dsdate  <- "_" %s+% format(Sys.Date(), "%Y%m%d") %T>% print()
```

```
## [1] "_20210401"
```

### specify filename for dataset

```
dsname="cleaned_weatherAUS"
dsrdata <-
  file.path(fpath, dsname %s+% dsdate %s+% ".RData") %T>%
  print()
```

```
## [1] "/Users/kimwong/OneDrive - University of Pittsburgh/Documents/Kim F. Wong/CRC_Workshop/2021/Advan
```

**Save R objects to binary RData format**

```
save(ds, dsname, dspath, dsdate, nobs,
     vars, target, risk, id, ignore, omit,
     inputi, inputs, numi, numc, cati, catc,
     form, seed, train, validate, test,
     tr_target, tr_risk, va_target, va_risk, te_target, te_risk,
     file=dsrdata)
```

**Check file size**

```
file.size(dsrdata) %>% comma()
```

```
## [1] "7,961,996"
```

**Reload dataset**

```
load(dsrdata) %>% print()
```

```
##  [1] "ds"        "dsname"    "dspath"    "dsdate"    "nobs"      "vars"
##  [7] "target"    "risk"      "id"        "ignore"    "omit"      "inputi"
## [13] "inputs"    "numi"      "numc"      "cati"      "catc"      "form"
## [19] "seed"      "train"     "validate"  "test"      "tr_target" "tr_risk"
## [25] "va_target" "va_risk"   "te_target" "te_risk"
```