Thesis

A Thesis
Presented to
The Division of
Reed College

In Partial Fulfillment
of the Requirements for the Degree

Sarah Maebius

December 6th, 2020

Approved for the Division
()

_____

Acknowledgements

Deeply grateful to...

Table of Contents

Introduction **1**

Chapter 1: R Markdown Basics **3**
1.1 Lists 3
1.2 Line breaks 4
1.3 R chunks 4
1.4 Inline code 4
1.5 Including plots 5
1.6 Loading and exploring data 5
1.7 Research Problem and Background 8
1.8 A Statistical Learning Approach to Identifying Location of Western Redcedars 9
1.9 Sources 9
1.10 Data Chapter 9
1.10.1 Imaging 10
1.10.2 Ground Data 10
1.10.3 Training Data 11

Chapter 2: Mathematics and Science **15**
2.1 Math 15
2.2 Chemistry 101: Symbols 15
2.2.1 Typesetting reactions 16
2.2.2 Other examples of reactions 16
2.3 Physics 16
2.4 Biology 16

Chapter 3: Tables, Graphics, References, and Labels **17**
3.1 Tables 17
3.2 Figures 18
3.3 Footnotes and Endnotes 20
3.4 Bibliographies 20
3.5 Anything else? 22

Conclusion **23**

Appendix A: The First Appendix **25**

Appendix B: The Second Appendix, for Fun **29**

References **31**

List of Tables

3.1 Correlation of Inheritance Factors for Parents and Child 17

List of Figures

3.1 Reed logo 18
3.2 Mean Delays by Airline 19
3.3 Subdiv. graph 20
3.4 A Larger Figure, Flipped Upside Down 20

Abstract

The preface pretty much says it all.

Second paragraph of abstract starts here.

Introduction

Welcome to the *R Markdown* thesis template. This template is based on (and in many places copied directly from) the Reed College LaTeX template, but hopefully it will provide a nicer interface for those that have never used TeX or LaTeX before. Using *R Markdown* will also allow you to easily keep track of your analyses in **R** chunks of code, with the resulting plots and output included as well. The hope is this *R Markdown* template gets you in the habit of doing reproducible research, which benefits you long-term as a researcher, but also will greatly help anyone that is trying to reproduce or build onto your results down the road.

Hopefully, you won't have much of a learning period to go through and you will reap the benefits of a nicely formatted thesis. The use of LaTeX in combination with *Markdown* is more consistent than the output of a word processor, much less prone to corruption or crashing, and the resulting file is smaller than a Word file. While you may have never had problems using Word in the past, your thesis is likely going to be about twice as large and complex as anything you've written before, taxing Word's capabilities. After working with *Markdown* and **R** together for a few weeks, we are confident this will be your reporting style of choice going forward.

Why use it?

R Markdown creates a simple and straightforward way to interface with the beauty of LaTeX. Packages have been written in **R** to work directly with LaTeX to produce nicely formatting tables and paragraphs. In addition to creating a user friendly interface to LaTeX, *R Markdown* also allows you to read in your data, to analyze it and to visualize it using **R** functions, and also to provide the documentation and commentary on the results of your project. Further, it allows for **R** results to be passed inline to the commentary of your results. You'll see more on this later.

Who should use it?

Anyone who needs to use data analysis, math, tables, a lot of figures, complex cross-references, or who just cares about the final appearance of their document should use *R Markdown*. Of particular use should be anyone in the sciences, but the user-friendly nature of *Markdown* and its ability to keep track of and easily include figures, automatically generate a table of contents, index, references, table of figures, etc. should make it of great benefit to nearly anyone writing a thesis project.

For additional help with bookdown

Please visit the free online bookdown reference guide.

Chapter 1

R Markdown Basics

Here is a brief introduction into using *R Markdown*. *Markdown* is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. *R Markdown* provides the flexibility of *Markdown* with the implementation of **R** input and output. For more details on using *R Markdown* see `https://rmarkdown.rstudio.com`.

Be careful with your spacing in *Markdown* documents. While whitespace largely is ignored, it does at times give *Markdown* signals as to how to proceed. As a habit, try to keep everything left aligned whenever possible, especially as you type a new paragraph. In other words, there is no need to indent basic text in the Rmd document (in fact, it might cause your text to do funny things if you do).

1.1 Lists

It's easy to create a list. It can be unordered like

- Item 1
- Item 2

or it can be ordered like

1. Item 1
2. Item 2

Notice that I intentionally mislabeled Item 2 as number 4. *Markdown* automatically figures this out! You can put any numbers in the list and it will create the list. Check it out below.

To create a sublist, just indent the values a bit (at least four spaces or a tab). (Here's one case where indentation is key!)

1. Item 1
2. Item 2
3. Item 3
    - Item 3a
    - Item 3b

1.2 Line breaks

Make sure to add white space between lines if you'd like to start a new paragraph. Look at what happens below in the outputted document if you don't:

Here is the first sentence. Here is another sentence. Here is the last sentence to end the paragraph. This should be a new paragraph.

Now for the correct way:

Here is the first sentence. Here is another sentence. Here is the last sentence to end the paragraph.

This should be a new paragraph.

1.3 R chunks

When you click the **Knit** button above a document will be generated that includes both content as well as the output of any embedded **R** code chunks within the document. You can embed an **R** code chunk like this (`cars` is a built-in **R** dataset):

```
summary(cars)
```

```
   speed           dist
Min.   : 4.0   Min.   :  2.00
1st Qu.:12.0   1st Qu.: 26.00
Median :15.0   Median : 36.00
Mean   :15.4   Mean   : 42.98
3rd Qu.:19.0   3rd Qu.: 56.00
Max.   :25.0   Max.   :120.00
```

1.4 Inline code

If you'd like to put the results of your analysis directly into your discussion, add inline code like this:

> The `cos` of 2π is 1.

Another example would be the direct calculation of the standard deviation:

> The standard deviation of `speed` in `cars` is 5.2876444.

One last neat feature is the use of the `ifelse` conditional statement which can be used to output text depending on the result of an **R** calculation:

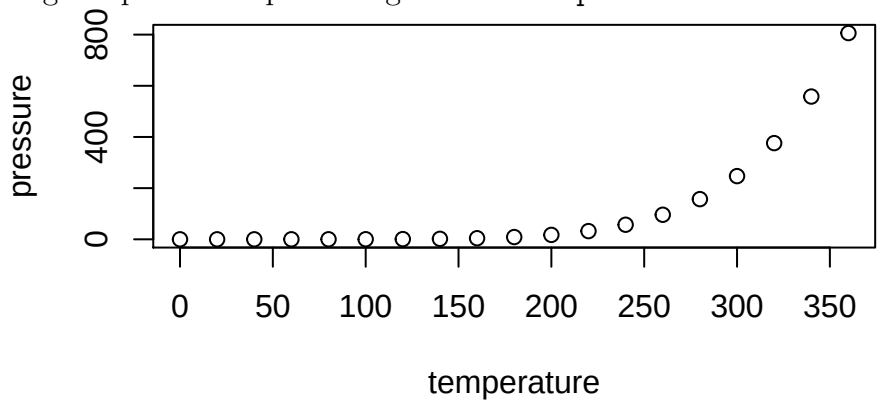
> The standard deviation is less than 6.

Note the use of `>` here, which signifies a quotation environment that will be indented.

As you see with `$2 \pi$` above, mathematics can be added by surrounding the mathematical text with dollar signs. More examples of this are in Mathematics and Science if you uncomment the code in Math.

1.5 Including plots

You can also embed plots. For example, here is a way to use the base **R** graphics package to produce a plot using the built-in `pressure` dataset:



Note that the `echo=FALSE` parameter was added to the code chunk to prevent printing of the **R** code that generated the plot. There are plenty of other ways to add chunk options (like `fig.height` and `fig.width` in the chunk above). More information is available at `https://yihui.org/knitr/options/`.

Another useful chunk option is the setting of `cache=TRUE` as you see here. If document rendering becomes time consuming due to long computations or plots that are expensive to generate you can use knitr caching to improve performance. Later in this file, you'll see a way to reference plots created in **R** or external figures.

1.6 Loading and exploring data

Included in this template is a file called `flights.csv`. This file includes a subset of the larger dataset of information about all flights that departed from Seattle and Portland in 2014. More information about this dataset and its **R** package is available at `https://github.com/ismayc/pnwflights14`. This subset includes only Portland flights and only rows that were complete with no missing values. Merges were also done with the `airports` and `airlines` data sets in the `pnwflights14` package to get more descriptive airport and airline names.

We can load in this data set using the following commands:

```
# flights.csv is in the data directory
# string columns will be read in as strings and not factors now
flights <- read.csv('~/tree_imaging/index/data/flights.csv', stringsAsFactors = FA
```

The data is now stored in the data frame called `flights` in **R**. To get a better feel for the variables included in this dataset we can use a variety of functions. Here we can see the dimensions (rows by columns) and also the names of the columns.

```
dim(flights)
```

```
[1] 12649    16
```

```
names(flights)
```

```
 [1] "month"        "day"          "dep_time"     "dep_delay"    "arr_time"
 [6] "arr_delay"    "carrier"      "tailnum"      "flight"       "dest"
[11] "air_time"     "distance"     "hour"         "minute"       "carrier_name"
[16] "dest_name"
```

Another good idea is to take a look at the dataset in table form. With this dataset having more than 20,000 rows, we won't explicitly show the results of the command here. I recommend you enter the command into the Console ***after*** you have run the **R** chunks above to load the data into **R**.

```
View(flights)
```

While not required, it is highly recommended you use the `dplyr` package to manipulate and summarize your data set as needed. It uses a syntax that is easy to understand using chaining operations. Below I've created a few examples of using `dplyr` to get information about the Portland flights in 2014. You will also see the use of the `ggplot2` package, which produces beautiful, high-quality academic visuals.

We begin by checking to ensure that needed packages are installed and then we load them into our current working environment:

```
# List of packages required for this analysis
pkg <- c("dplyr", "ggplot2", "knitr", "bookdown")
# Check if packages are not installed and assign the
# names of the packages not installed to the variable new.pkg
new.pkg <- pkg[!(pkg %in% installed.packages())]
# If there are any packages in the list that aren't installed,
# install them
if (length(new.pkg)) {
  install.packages(new.pkg, repos = "https://cran.rstudio.com")
}
# Load packages
library(thesisdown)
library(dplyr)
```

```
Warning: package 'dplyr' was built under R version 3.6.2
```

```
library(ggplot2)
```

```
Warning: package 'ggplot2' was built under R version 3.6.2
```

```
library(knitr)
```

```
Warning: package 'knitr' was built under R version 3.6.2
```

1.7 Research Problem and Background

Western redcedar trees are evergreen trees that typically grow up to 75 feet tall and are located over the Pacific Northwest, making it an organism with a tolerance for shaded regions with moist environments. These trees are native to the land and have served many purposes to people and animals living in the vicinity of the trees, including medicinal, building, and habitat functions [1]. However, over the past decade, reports of dead western redcedars have been increasing, suggesting that something other than natural causes is killing off this species. In general, this species experiences several hardships in surviving in the Pacific Northwest, with common causes of death such as forest fires, clearcutting, small animals eating the saplings, and harsh weather including strong winds that easily uproot the trees [1]. Dying redcedar trees can be identified by their branches which turn brownish yellow or fall off completely. Another sign is that the top of a dying redcedar will turn brown and lose leaves [2]. Losing this native tree would have a detrimental effect on animals in the area who rely heavily on the trees for their lifestyle. Scientists have speculated that western redcedar decline might be caused by recent dry summers, the spread of tree disease, insects, or other weather related events [2]. Since this is a recent issue, there is not a lot of resources explaining the decline. This research aims to provide more insight into the cause of the western redcedar decline by first predicting the location of the western redcedar trees in Portland and then predicting their condition in terms of health.

Modelling in this research will be conducted by combining information gathered from remote sensing images with ground level information. There are several sources publishing research done on using satellite imagery for land classification as well as for predicting tree species, which will be the groundwork for this project's application of remote sensing models to the specific topic of western redcedar mortality.

Plenty of literature has been published in utilizing satellite imagery for predictive models, however, many issues arise in classifying land type through remote sensing mainly due to image quality. A single image can be composed of image strips taken over the course of multiple flight paths. Consequently, these images are taken at different times of day, which compiles into a single image with a lot of variation in values due to changes in the weather as well as the different angles of the sun's position [4]. Ideally, a solid classifying model surpasses any error introduced by imperfection in the satellite images. Common approaches to classification on satellite images include support vector machines, random forests, and decision trees. A study on land type classification explores the performance of convolutional neural networks (CNN) as classification models, namely, the GoogleNet and CaffeNet models with feature vector output on two remote sensing datasets [4]. The results of that experiment conclude that using a pre-trained CNN successfully classifies land type on the UC Merced Land Use dataset. The GoogleNet method has the benefit of reducing complexity of filter layers, while the CaffeNet model has convolutional layers followed by pooling layers and fully connected layers.

Even more pertinent to this research is a study on identifying 7 tree species by applying a hyperspectral CNN model. This study was completed over a single north

to south strip of hyperspectral imagery data (16km long by 1km wide) taken over the mountains in California. The collected data from the remote-sensing imagery corresponds with a strip of land for which field data had been collected. The collected field data includes measured information about tree count, tree species, and mortality status. To build the models the data is separated into 10 folds with the application of k-fold cross validation. The hyperspectral image is converted into the form of a tree canopy height model and circular polygons are centered at individual tree canopies in such a way that each circle gets assigned a tree species and mortality status and treated as a single observation. Next, a convolutional neural network is trained on the designated training data, with the results concluding that the model most accurately identifies pine trees at the genus level, and Jeffrey pine species, sugar pine species, and incense cedar species all with F-scores around 0.90 or more. One complication that comes up in the study is the uneven distribution of proportion of examples in each class, which gets accounted for by applying a balanced loss function. Another model included is an RGB model, which does not perform as well as the hyperspectral model in terms of distinguishing tree classes, but does perform around the same level of accuracy in terms of genus classification.

1.8 A Statistical Learning Approach to Identifying Location of Western Redcedars

This work applies the classification methods in [5] to predict tree species using satellite imagery to locate western redcedars in Portland, Oregon in order model the condition of western redcedars and understand the cause of death in this species.

1.9 Sources

[1] `https://plants.usda.gov/plantguide/pdf/cs_thpl.pdf`

[2] `https://ppo.puyallup.wsu.edu/plant-health-concerns/redcedar/`

[3] `https://www.treespnw.com/resources/2018/11/7/are-the-western-redcedars-dying`

[4] `https://arxiv.org/pdf/1508.00092.pdf` (Land Use Classification in Remote Sensing Images by Convolutional Neural Networks)

[5] `https://www.fs.fed.us/psw/publications/north/psw_2019_north009_fricker.pdf` (A Convolutional Neural Network Classifier Identifies Tree Species in Mixed-Confier Forest from Hyperspectral Imagery)

1.10 Data Chapter

There are two sources of data in this project. Data at a pixel level come from satellite images downloaded from Planet.com (`https://www.planet.com`), and data at the ground level come from a library called "pdxTrees" in RStudio (`https://github.`

com/mcconvil/pdxTrees).

1.10.1 Imaging

The images from Planet.com were taken by satellites on September 2nd, 2020 at an altitude of 475km. The compiled multi-spectral image is composed of 4 bands: blue, green, red, and near-infrared, with center wavelengths 490nm, 565nm, 665nm, and 865nm respectively, with an average bandwidth of 41nm. The average spatial extent is around 25km by 8km. The images cover the entire Portland area, with specific measurements of 26km from west to east and 30km from north to south. The spatial resolution of the pixels is approximately 3 meters. Downloaded images are then opened in QGIS where further analysis on the images can be made.

Using Planet.com for gathering satellite images has the added benefit of providing a limited number of downloadable images by making a free account. However, one drawback to using this data is that it has a poor resolution, which will decrease the performance of statistical models trying to predict tree species location.

https://www.planet.com/products/satellite-imagery/files/1610.06_Spec%20Sheet_Combined_Imagery_Product_Letter_ENGv1.pdf https://developers.planet.com/docs/data/sensors/

1.10.2 Ground Data

Ground level data is available from the 'pdxTrees' package in RStudio. This data was collected as part of the Portland's Parks and Recreation's Urban Forestry Tree Inventory Project, which collected park tree information from 2017 to 2019. Originally, the inventory project started with the goal of improving the city's tree management plans. Under the guidance of US Forestry staff, volunteers around the city's neighborhoods are trained in identifying tree species and provided with tools necessary to record tree measurements. The Portland park trees inventory consists of over 25,000 trees, each with information about tree location, size, species, and health. For the purposes of this project, the following variables were selected: Spatial information: Longitude and latitude of tree Species: Tree genus, species, and common name Crown size: Crown width from north to south, crown width from east to west, and the base height of the crown in feet Tree Condition: Categorical variable with four categories (from best to worst), good, fair, poor, and dead To prepare the ground level data for training the model, the data was filtered by species and crown size. The following species were identified and included: maples, oak, douglas fir, western red cedar, oregon bigleaf, sequoia. Since the data is used to identify trees in the satellite images, the ground data was also filtered to include only the trees with a crown width from north to south of 20 feet or more, this ensures that there will be around 9 to 20 pixels for each tree.

Portland's Tree Inventory is appropriate for this project, because it provides the variables necessary for identifying Western Redcedars as well as the larger trees in the dataset, which is ideal for constructing a training set of tree polygons in a satellite image. One way this data would be even better suited for the purposes of this project

would be if the street trees subdata also contained variables for tree canopy size.

1.10.3 Training Data

Characteristics of Training Data

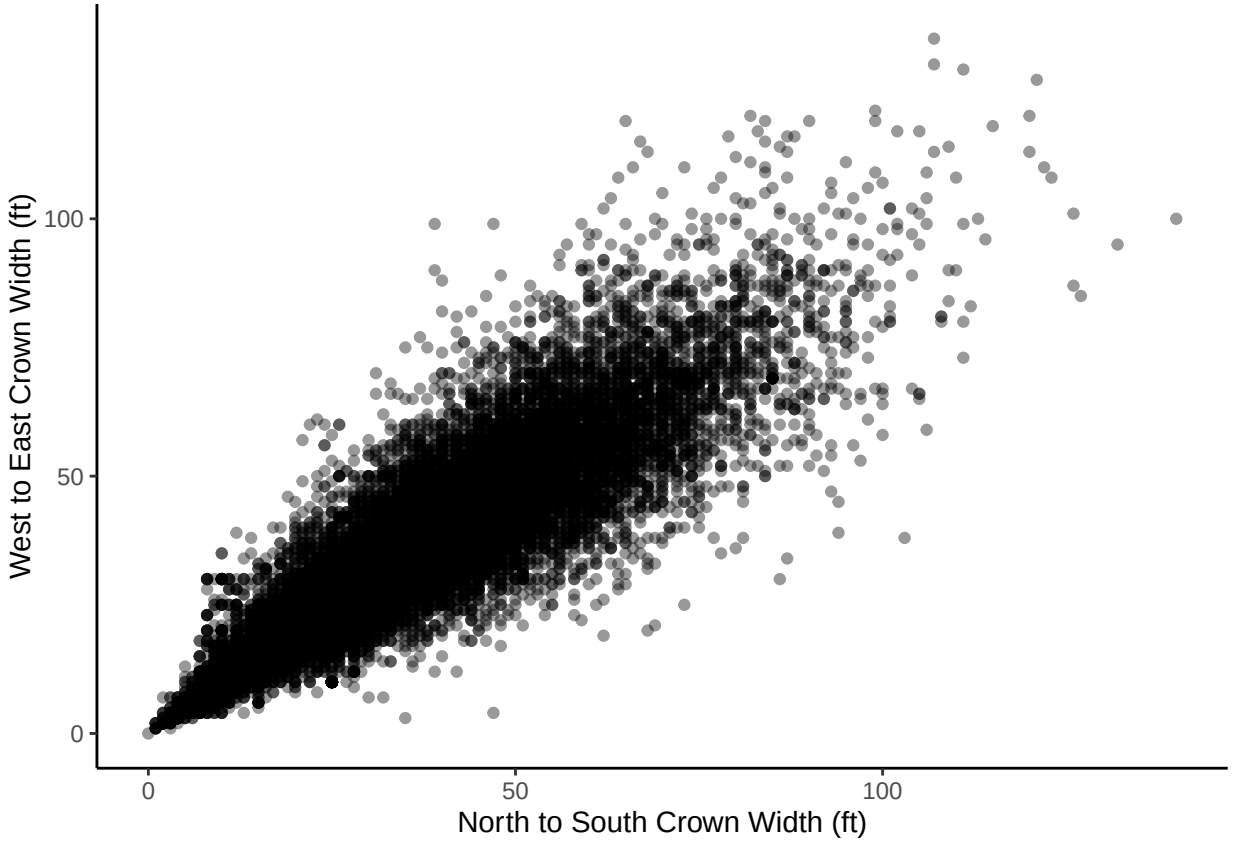
The training data at the ground-level was filtered by tree size and tree type. Only trees with at least 20 feet in north to south crown width were included in the training dataset, to ensure that the polygons around each tree contain around 9 to 20 pixels. Figure **??** plots the north to south crown width values against the west to east crown width values to present the ranges of crown widths in the entire pdxTrees parks dataset and to justify only using the north to south crown width for filtering the dataset since the two variables have a strong positive correlation.

```
library(pdxTrees)
```

```
Warning: package 'pdxTrees' was built under R version 3.6.2
```

```
library(ggplot2)
pdxTrees_parks <- get_pdxTrees_parks()
ggplot(pdxTrees_parks, aes(x = Crown_Width_NS, y = Crown_Width_EW)) +
  geom_point(alpha = 0.4) +
  theme_classic() + labs(x = "North to South Crown Width (ft)", y = "West to East
```

```
Warning: Removed 263 rows containing missing values (geom_point).
```



Six tree types were determined to have large enough canopies on average and appear frequently enough to construct a training dataset with plenty of observations for each species of tree: Douglas-Fir, English Oak, Giant Sequoia, Maple, Western Redcedar. Figure **??** provides the number of trees under each common name category. The largest group contains 6485 observations of Douglas-Fir trees, and the smallest group contains 135 observations of English Oak trees.

```
library(tidyverse)
```

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

v tibble  3.0.4     v purrr   0.3.4
v tidyr   1.0.2     v stringr 1.4.0
v readr   1.3.1     v forcats 0.5.0

Warning: package 'tibble' was built under R version 3.6.2

Warning: package 'purrr' was built under R version 3.6.2

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

```
train_points <- pdxTrees_parks %>%
  dplyr::filter(Species %in% c("QURO", "SEGI", "ACPL", "PSME", "ACMA", "THPL"), Cr

train_points %>%
  group_by(Common_Name) %>%
  summarise(counts = n()) %>%
  arrange(desc(counts))
```

```
`summarise()` ungrouping output (override with `.groups` argument)
```

```
# A tibble: 6 x 2
  Common_Name      counts
  <chr>             <int>
1 Douglas-Fir        6485
2 Norway Maple       1406
3 Western Redcedar    652
4 Bigleaf Maple       464
5 Giant Sequoia       315
6 English Oak         135
```

Creating Spatial Polygons

The training dataset was converted into shapefiles for each type of tree and exported into QGIS where polygons were manually drawn around 100 of the trees for each species. Displaying the point shapefiles layer over the raster images downloaded from Planet.com in QGIS provided a guide point for locating individual trees around which a polygon can be created by tracing the tree canopy. Careful attention was paid to avoid drawing polygons around trees with canopies that overlap with other trees of different species along with trees that are cast in shadows or otherwise partially obstructed by surrounded structures. For the five different species of trees, at least 100 polygons were drawn around trees of that type, with each polygon containing at least 6 pixels and at most 20 pixels. The polygon shapefiles were then exported into RStudio where the rest of the analysis was conducted.

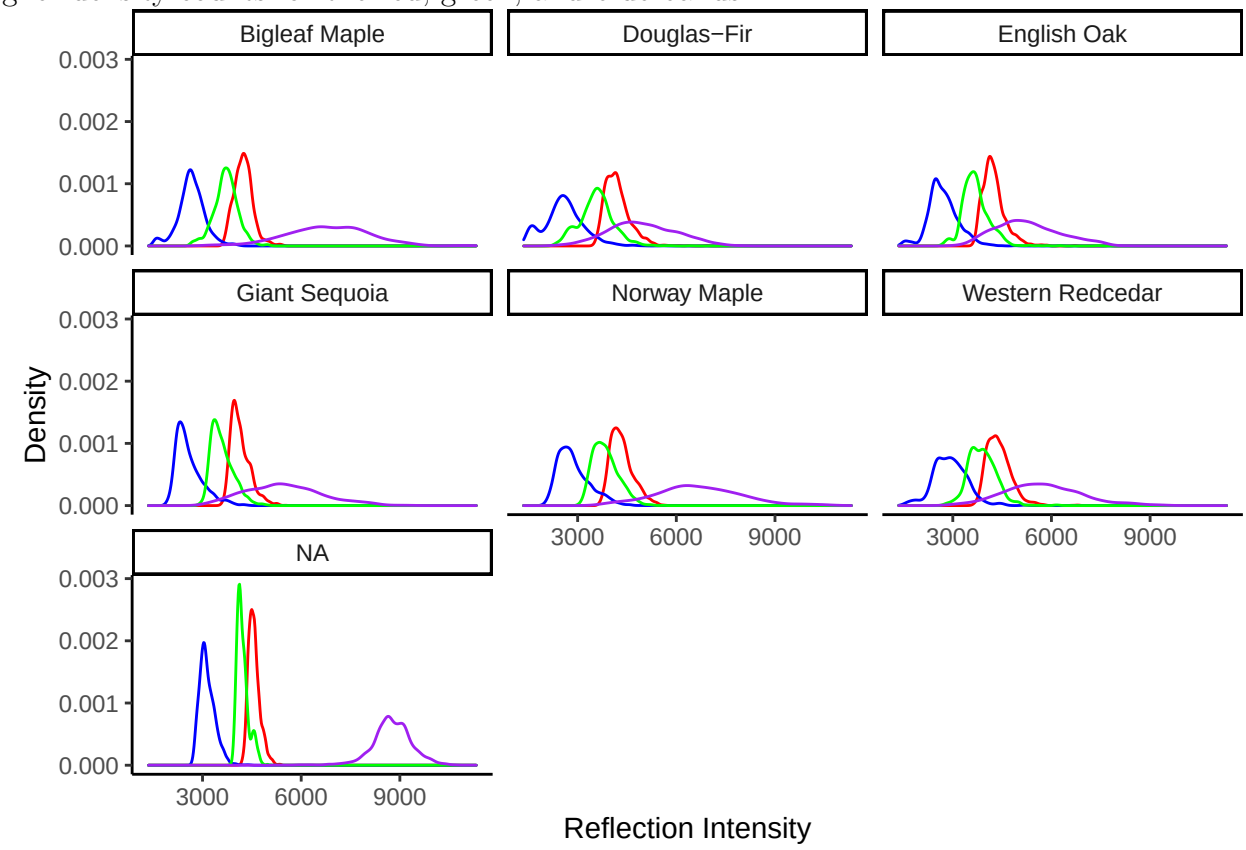
Combining Ground-level Data with Pixel-level Data

The first spatial join in RStudio was conducted to match up each Spatial Polygon with a point in the training dataset. Then the raster images were loaded and joined with the polygons to extract the pixel values inside each polygon for all 4 bands. Ultimately, a pixel table with rows representing each pixel with its corresponding polygon, light reflection intensity values for all 4 bands, and the ground information about that tree. Table **??** contains the summary of the total number of pixels for each tree type.

```
`summarise()` ungrouping output (override with `.groups` argument)
```

```
# A tibble: 7 x 2
  Cmmn_Nm          counts
  <fct>             <int>
1 Bigleaf Maple       885
2 Douglas-Fir        1380
3 English Oak        1025
4 Giant Sequoia      1436
5 Norway Maple       2340
6 Western Redcedar   1364
7 <NA>               2923
```

Figure ?? displays the range of reflection intensity pixel values per tree type. Each tree species has similar ranges of values over the red, green, blue, and infared bands. Of the species included in the training data, the Giant Sequoia trees appear to have higher density counts for the red, green, and blue bands.



Chapter 2

Mathematics and Science

2.1 Math

TEX is the best way to typeset mathematics. Donald Knuth designed TEX when he got frustrated at how long it was taking the typesetters to finish his book, which contained a lot of mathematics. One nice feature of *R Markdown* is its ability to read LaTeX code directly.

If you are doing a thesis that will involve lots of math, you will want to read the following section which has been commented out. If you're not going to use math, skip over or delete this next commented section.

2.2 Chemistry 101: Symbols

Chemical formulas will look best if they are not italicized. Get around math mode's automatic italicizing in LaTeX by using the argument `$\mathrm{formula here}$`, with your formula inside the curly brackets. (Notice the use of the backticks here which enclose text that acts as code.)

So, $Fe_2^{2+}Cr_2O_4$ is written `$\mathrm{Fe_2^{2+}Cr_2O_4}$`.
Exponent or Superscript: O^-
Subscript: CH_4

To stack numbers or letters as in Fe_2^{2+}, the subscript is defined first, and then the superscript is defined.
Bullet: $CuCl \bullet 7H_2O$
Delta: Δ
Reaction Arrows: $\longrightarrow$ or $\xrightarrow{solution}$
Resonance Arrows: $\leftrightarrow$
Reversible Reaction Arrows: $\rightleftharpoons$

2.2.1 Typesetting reactions

You may wish to put your reaction in an equation environment, which means that LaTeX will place the reaction where it fits and will number the equations for you.

$$\mathrm{C_6H_{12}O_6 + 6O_2 \longrightarrow 6CO_2 + 6H_2O} \tag{2.1}$$

We can reference this combustion of glucose reaction via Equation (2.1).

2.2.2 Other examples of reactions

$\mathrm{NH_4Cl_{(s)} \rightleftharpoons NH_{3(g)} + HCl_{(g)}}$
$\mathrm{MeCH_2Br + Mg \xrightarrow[below]{above} MeCH_2 \bullet Mg \bullet Br}$

2.3 Physics

Many of the symbols you will need can be found on the math page `https://web.reed.edu/cis/help/latex/math.html` and the Comprehensive LaTeX Symbol Guide (`https://mirror.utexas.edu/ctan/info/symbols/comprehensive/symbols-letter.pdf`).

2.4 Biology

You will probably find the resources at `https://www.lecb.ncifcrf.gov/~toms/latex.html` helpful, particularly the links to bsts for various journals. You may also be interested in TeXShade for nucleotide typesetting (`https://homepages.uni-tuebingen.de/beitz/txe.html`). Be sure to read the proceeding chapter on graphics and tables.

Chapter 3

Tables, Graphics, References, and Labels

3.1 Tables

In addition to the tables that can be automatically generated from a data frame in **R** that you saw in R Markdown Basics using the `kable()` function, you can also create tables using *pandoc*. (More information is available at `https://pandoc.org/README.html#tables`.) This might be useful if you don't have values specifically stored in **R**, but you'd like to display them in table form. Below is an example. Pay careful attention to the alignment in the table and hyphens to create the rows and columns.

Table 3.1: Correlation of Inheritance Factors for Parents and Child

Factors	Correlation between Parents & Child	Inherited
Education	-0.49	Yes
Socio-Economic Status	0.28	Slight
Income	0.08	No
Family Size	0.18	Slight
Occupational Prestige	0.21	Slight

We can also create a link to the table by doing the following: Table 3.1. If you go back to Loading and exploring data and look at the `kable` table, we can create a reference to this max delays table too: Table **??**. The addition of the `(\#tab:inher)` option to the end of the table caption allows us to then make a reference to Table `\@ref(tab:label)`. Note that this reference could appear anywhere throughout the document after the table has appeared.

3.2 Figures

If your thesis has a lot of figures, *R Markdown* might behave better for you than that other word processor. One perk is that it will automatically number the figures accordingly in each chapter. You'll also be able to create a label for each figure, add a caption, and then reference the figure in a way similar to what we saw with tables earlier. If you label your figures, you can move the figures around and *R Markdown* will automatically adjust the numbering for you. No need for you to remember! So that you don't have to get too far into LaTeX to do this, a couple **R** functions have been created for you to assist. You'll see their use below.

In the **R** chunk below, we will load in a picture stored as `reed.jpg` in our main directory. We then give it the caption of "Reed logo", the label of "reedlogo", and specify that this is a figure. Make note of the different **R** chunk options that are given in the R Markdown file (not shown in the knitted document).

```
include_graphics(path = "figure/reed.jpg")
```

Figure 3.1: Reed logo

Here is a reference to the Reed logo: Figure 3.1. Note the use of the `fig:` code here. By naming the **R** chunk that contains the figure, we can then reference that figure later as done in the first sentence here. We can also specify the caption for the figure via the R chunk option `fig.cap`.

Below we will investigate how to save the output of an **R** plot and label it in a way similar to that done above. Recall the `flights` dataset from Chapter 1. (Note that we've shown a different way to reference a section or chapter here.) We will next explore a bar graph with the mean flight departure delays by airline from Portland for 2014.

```
mean_delay_by_carrier <- flights %>%
  group_by(carrier) %>%
  summarize(mean_dep_delay = mean(dep_delay))
```

```
`summarise()` ungrouping output (override with `.groups` argument)
```

```
ggplot(mean_delay_by_carrier, aes(x = carrier, y = mean_dep_delay)) +
  geom_bar(position = "identity", stat = "identity", fill = "red")
```

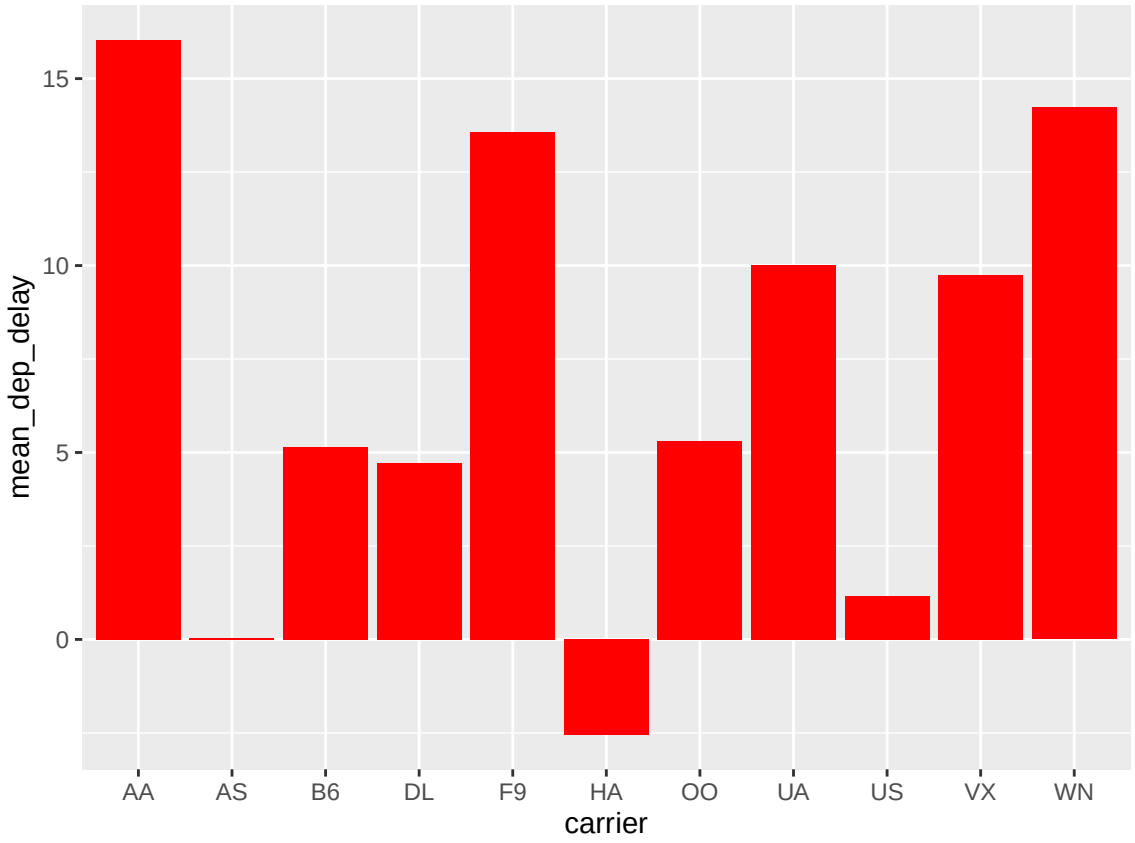


Figure 3.2: Mean Delays by Airline

Here is a reference to this image: Figure 3.2.

A table linking these carrier codes to airline names is available at `https://github.com/ismayc/pnwflights14/blob/master/data/airlines.csv`.

Next, we will explore the use of the `out.extra` chunk option, which can be used to shrink or expand an image loaded from a file by specifying `"scale= "`. Here we use the mathematical graph stored in the "subdivision.pdf" file.

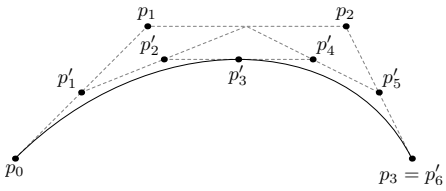


Figure 3.3: Subdiv. graph

Here is a reference to this image: Figure 3.3. Note that `echo=FALSE` is specified so that the **R** code is hidden in the document.

More Figure Stuff

Lastly, we will explore how to rotate and enlarge figures using the `out.extra` chunk option. (Currently this only works in the PDF version of the book.)

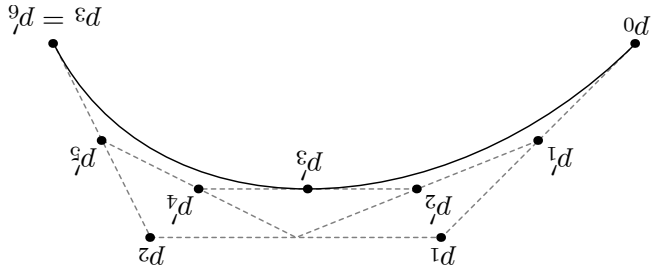


Figure 3.4: A Larger Figure, Flipped Upside Down

As another example, here is a reference: Figure 3.4.

3.3 Footnotes and Endnotes

You might want to footnote something.[1] The footnote will be in a smaller font and placed appropriately. Endnotes work in much the same way. More information can be found about both on the CUS site or feel free to reach out to `data@reed.edu`.

3.4 Bibliographies

Of course you will need to cite things, and you will probably accumulate an armful of sources. There are a variety of tools available for creating a bibliography database (stored with the .bib extension). In addition to BibTeX suggested below, you may want to consider using the free and easy-to-use tool called Zotero. The Reed librarians have created Zotero documentation at `https://libguides.reed.`

[1] footnote text

edu/citation/zotero. In addition, a tutorial is available from Middlebury College at https://sites.middlebury.edu/zoteromiddlebury/.

R Markdown uses *pandoc* (https://pandoc.org/) to build its bibliographies. One nice caveat of this is that you won't have to do a second compile to load in references as standard LaTeX requires. To cite references in your thesis (after creating your bibliography database), place the reference name inside square brackets and precede it by the "at" symbol. For example, here's a reference to a book about worrying: (Molina & Borkovec, 1994). This Molina1994 entry appears in a file called thesis.bib in the bib folder. This bibliography database file was created by a program called BibTeX. You can call this file something else if you like (look at the YAML header in the main .Rmd file) and, by default, is to placed in the bib folder.

For more information about BibTeX and bibliographies, see our CUS site (https://web.reed.edu/cis/help/latex/index.html)[2]. There are three pages on this topic: *bibtex* (which talks about using BibTeX, at https://web.reed.edu/cis/help/latex/bibtex.html), *bibtexstyles* (about how to find and use the bibliography style that best suits your needs, at https://web.reed.edu/cis/help/latex/bibtexstyles.html) and *bibman* (which covers how to make and maintain a bibliography by hand, without BibTeX, at https://web.reed.edu/cis/help/latex/bibman.html). The last page will not be useful unless you have only a few sources.

If you look at the YAML header at the top of the main .Rmd file you can see that we can specify the style of the bibliography by referencing the appropriate csl file. You can download a variety of different style files at https://www.zotero.org/styles. Make sure to download the file into the csl folder.

Tips for Bibliographies

- Like with thesis formatting, the sooner you start compiling your bibliography for something as large as thesis, the better. Typing in source after source is mind-numbing enough; do you really want to do it for hours on end in late April? Think of it as procrastination.
- The cite key (a citation's label) needs to be unique from the other entries.
- When you have more than one author or editor, you need to separate each author's name by the word "and" e.g. `Author = {Noble, Sam and Youngberg, Jessica},`.
- Bibliographies made using BibTeX (whether manually or using a manager) accept LaTeX markup, so you can italicize and add symbols as necessary.
- To force capitalization in an article title or where all lowercase is generally used, bracket the capital letter in curly braces.
- You can add a Reed Thesis citation[3] option. The best way to do this is to use the phdthesis type of citation, and use the optional "type" field to enter "Reed thesis" or "Undergraduate thesis."

[2]Reed College (2007)
[3]Noble (2002)

3.5 Anything else?

If you'd like to see examples of other things in this template, please contact the Data @ Reed team (email `data@reed.edu`) with your suggestions. We love to see people using *R Markdown* for their theses, and are happy to help.

Conclusion

If we don't want Conclusion to have a chapter number next to it, we can add the `{-}` attribute.

More info

And here's some other random info: the first paragraph after a chapter title or section head *shouldn't be* indented, because indents are to tell the reader that you're starting a new paragraph. Since that's obvious after a chapter or section title, proper typesetting doesn't add an indent there.

Appendix A

The First Appendix

This first appendix includes all of the R chunks of code that were hidden throughout the document (using the `include = FALSE` chunk tag) to help with readibility and/or setup.

In the main Rmd file

```
# This chunk ensures that the thesisdown package is
# installed and loaded. This thesisdown package includes
# the template files for the thesis.
if(!require(devtools))
  install.packages("devtools", repos = "http://cran.rstudio.com")
if(!require(thesisdown))
  devtools::install_github("ismayc/thesisdown")
library(thesisdown)
```

In Chapter 3:

```
# This chunk ensures that the thesisdown package is
# installed and loaded. This thesisdown package includes
# the template files for the thesis and also two functions
# used for labeling and referencing
if (!require(remotes)) {
  if (params$`Install needed packages for {thesisdown}`) {
    install.packages("remotes", repos = "https://cran.rstudio.com")
  } else {
    stop(
      paste(
        'You need to run install.packages("remotes")',
        "first in the Console."
      )
    )
  }
}
```

```
if (!require(dplyr)) {
  if (params$`Install needed packages for {thesisdown}`) {
    install.packages("dplyr", repos = "https://cran.rstudio.com")
  } else {
    stop(
      paste(
        'You need to run install.packages("dplyr")',
        "first in the Console."
      )
    )
  }
}
if (!require(ggplot2)) {
  if (params$`Install needed packages for {thesisdown}`) {
    install.packages("ggplot2", repos = "https://cran.rstudio.com")
  } else {
    stop(
      paste(
        'You need to run install.packages("ggplot2")',
        "first in the Console."
      )
    )
  }
}
if (!require(bookdown)) {
  if (params$`Install needed packages for {thesisdown}`) {
    install.packages("bookdown", repos = "https://cran.rstudio.com")
  } else {
    stop(
      paste(
        'You need to run install.packages("bookdown")',
        "first in the Console."
      )
    )
  }
}
if (!require(thesisdown)) {
  if (params$`Install needed packages for {thesisdown}`) {
    remotes::install_github("ismayc/thesisdown")
  } else {
    stop(
      paste(
        "You need to run",
        'remotes::install_github("ismayc/thesisdown")',
```

```
        "first in the Console."
      )
    )
  }
}
library(thesisdown)
library(dplyr)
library(ggplot2)
library(knitr)
flights <- read.csv("data/flights.csv", stringsAsFactors = FALSE)
```

Appendix B

The Second Appendix, for Fun

References

Angel, E. (2000). *Interactive computer graphics : A top-down approach with opengl*. Boston, MA: Addison Wesley Longman.

Angel, E. (2001a). *Batch-file computer graphics : A bottom-up approach with quicktime*. Boston, MA: Wesley Addison Longman.

Angel, E. (2001b). *Test second book by angel*. Boston, MA: Wesley Addison Longman.

Molina, S. T., & Borkovec, T. D. (1994). The Penn State worry questionnaire: Psychometric properties and associated characteristics. In G. C. L. Davey & F. Tallis (Eds.), *Worrying: Perspectives on theory, assessment and treatment* (pp. 265–283). New York: Wiley.

Noble, S. G. (2002). *Turning images into simple line-art* (Undergraduate thesis). Reed College.

Reed College. (2007). LaTeX your document. Retrieved from `https://web.reed.edu/cis/help/LaTeX/index.html`