

**The effect of vegetation spatial structure on forest resilience to
wildfire and bark beetle disturbance in the Sierra Nevada,
California**

By

MICHAEL J. KOONTZ

DISSERTATION

Submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

ECOLOGY

in the

OFFICE OF GRADUATE STUDIES

of the

UNIVERSITY OF CALIFORNIA

DAVIS

Approved:

(Andrew M. Latimer), Chair

(Malcolm P. North)

(Constance I. Millar)

Committee in Charge

2019

The effect of vegetation spatial structure on forest resilience to wildfire and bark beetle
disturbance in the Sierra Nevada, California

Copyright © 2019

by

Michael J. Koontz

“One thing to remember is to talk to the animals. If you do, they will talk back to you. But if you don’t talk to them, they won’t talk back to you, then you won’t understand. And when you don’t understand, you will fear, and when you fear, you will destroy the animals, and if you destroy the animals, you will destroy yourself”

(Chief Dan George, Tseil-Waututh Nation, North Vancouver)”

Acknowledgements

“My acknowledgments”

Michael J. Koontz
April 2019
ECOLOGY

Abstract

“Complex forest structure affects disturbances.”

Contents

Abstract	v
List of Tables	vii
List of Figures	viii
1 Introduction	1
2 R Markdown Basics	3
Lists	3
Line breaks	4
R chunks	5
Inline code	5
Including plots	6
Loading and exploring data	7
Additional resources	13
3 Mathematics and Science	14
Math	14
Chemistry 101: Symbols	14
Physics	16
Biology	16

4	Tables, Graphics, References, and Labels	17
	Tables	17
	Figures	19
	Footnotes and Endnotes	23
	Bibliographies	24
	Anything else?	25
	Conclusion	26
A	The First Appendix	27
B	The Second Appendix, for Fun	29
	Colophon	30
	References	34

List of Tables

2.1	Max Delays by Airline	11
4.1	Correlation of Inheritance Factors for Parents and Child	18

List of Figures

4.1	UW logo	20
4.2	Mean Delays by Airline	21
4.3	Subdiv. graph	23
4.4	A Larger Figure, Flipped Upside Down	23

Chapter 1

Introduction

Welcome to the *R Markdown* thesis template. This template is based on (and in many places copied directly from) the UW 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 at least 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.

Chapter 2

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 <http://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).

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

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:

R CHUNKS

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

This should be a new paragraph.

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

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.
```

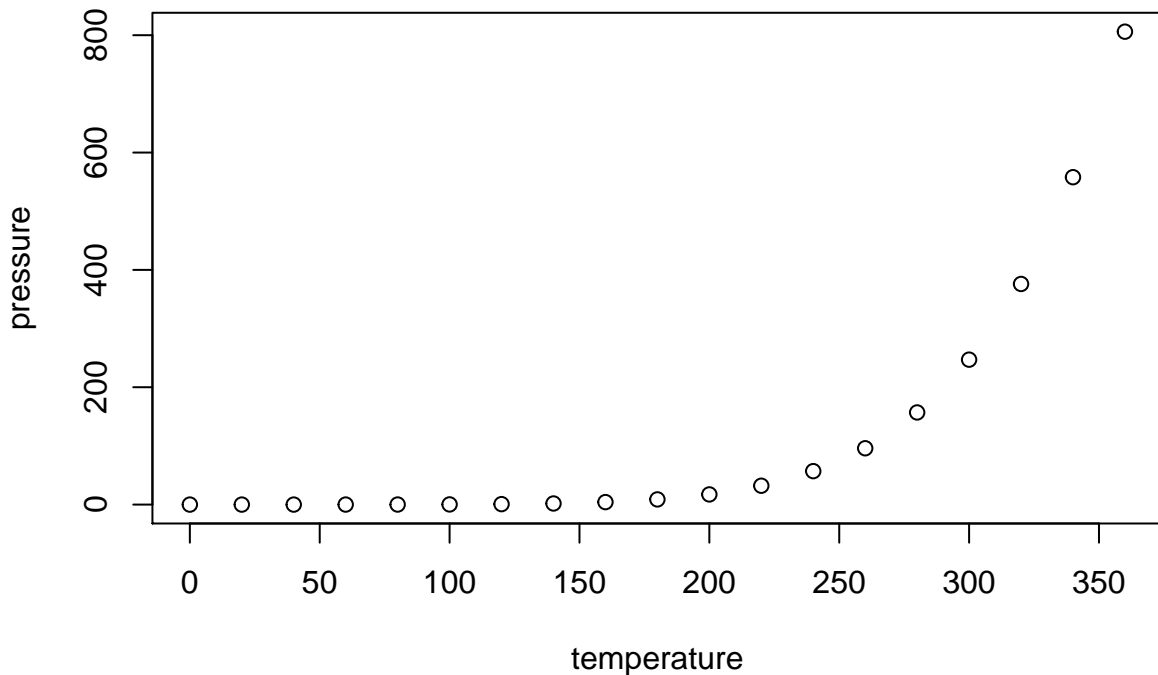
Note the use of `>` here, which signifies a quotation environment that will be indented.

As you see with `2π` 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](#).

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:

LOADING AND EXPLORING DATA



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. More information is available at <http://yihui.name/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.

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 <http://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 command:

```
flights <- read.csv("data/flights.csv")
```

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] 52808    16
```

```
names(flights)
```

```
[1] "month"      "day"        "dep_time"   "dep_delay"
[5] "arr_time"   "arr_delay"  "carrier"    "tailnum"
[9] "flight"     "dest"       "air_time"   "distance"
[13] "hour"       "minute"     "carrier_name" "dest_name"
```

Another good idea is to take a look at the dataset in table form. With this dataset having more than 50,000 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)
```

LOADING AND EXPLORING DATA

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", "devtools")
# 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 = "http://cran.rstudio.com")
# Load packages
library(aggiedown)
```

The example we show here does the following:

- Selects only the `carrier_name` and `arr_delay` from the `flights` dataset and then assigns this subset to a new variable called `flights2`.
- Using `flights2`, we determine the largest arrival delay for each of the carriers.

```
library(dplyr)
```

```
Attaching package: 'dplyr'
```

The following objects are masked from 'package:stats':

```
filter, lag
```

The following objects are masked from 'package:base':

```
intersect, setdiff, setequal, union
```

```
flights2 <- flights %>%
  select(carrier_name, arr_delay)
max_delays <- flights2 %>%
  group_by(carrier_name) %>%
  summarize(max_arr_delay = max(arr_delay, na.rm = TRUE))
```

A useful function in the `knitr` package for making nice tables in *R Markdown* is called `kable`. It is much easier to use than manually entering values into a table by copying

LOADING AND EXPLORING DATA

and pasting values into Excel or LaTeX. This again goes to show how nice reproducible documents can be! (Note the use of `results="asis"`, which will produce the table instead of the code to create the table.) The `caption.short` argument is used to include a shorter title to appear in the List of Tables.

```
library(knitr)

kable(max_delays,
      col.names = c("Airline", "Max Arrival Delay"),
      caption = "Maximum Delays by Airline",
      caption.short = "Max Delays by Airline",
      longtable = TRUE,
      booktabs = TRUE)
```

Table 2.1: Maximum Delays by Airline

Airline	Max Arrival Delay
Alaska Airlines Inc.	338
American Airlines Inc.	1539
Delta Air Lines Inc.	651
Frontier Airlines Inc.	575
Hawaiian Airlines Inc.	407
JetBlue Airways	273
SkyWest Airlines Inc.	421
Southwest Airlines Co.	694
United Air Lines Inc.	472
US Airways Inc.	347

The last two options make the table a little easier-to-read.

We can further look into the properties of the largest value here for American Airlines Inc. To do so, we can isolate the row corresponding to the arrival delay of 1539 minutes for American in our original `flights` dataset.

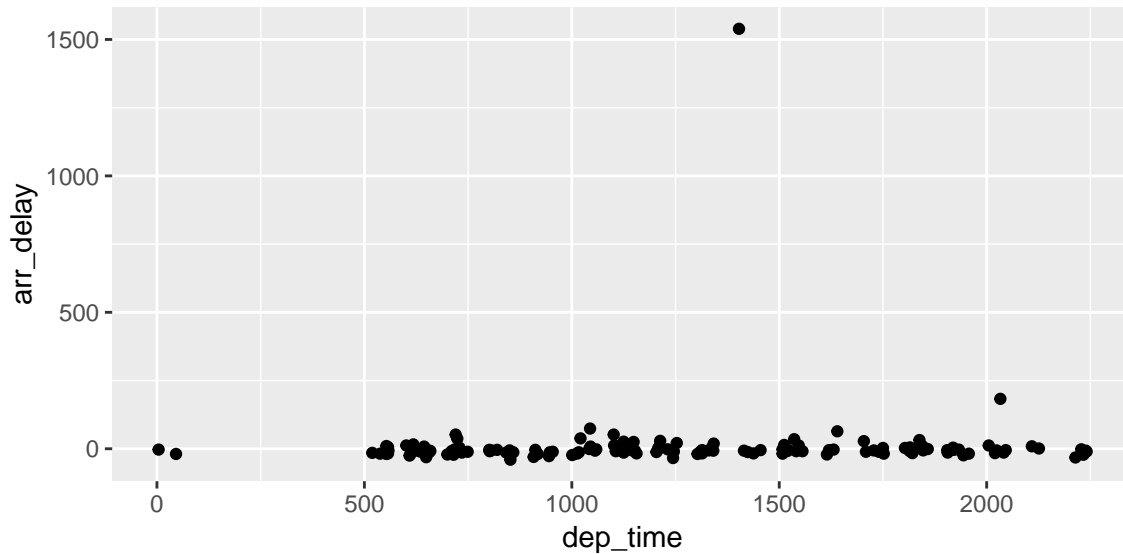
```
flights %>%
  filter(arr_delay == 1539,
         carrier_name == "American Airlines Inc.") %>%
  select(-c(month, day, carrier, dest_name, hour,
            minute, carrier_name, arr_delay))
```

	dep_time	dep_delay	arr_time	tailnum	flight	dest	air_time	distance
1	1403	1553	1934	N595AA	1568	DFW	182	1616

We see that the flight occurred on March 3rd and departed a little after 2 PM on its way to Dallas/Fort Worth. Lastly, we show how we can visualize the arrival delay of all departing flights from Portland on March 3rd against time of departure.

```
library(ggplot2)
flights %>%
  filter(month == 3, day == 3) %>%
  ggplot(aes(x = dep_time,
            y = arr_delay)) +
  geom_point()
```

ADDITIONAL RESOURCES



Additional resources

- *Markdown* Cheatsheet - <https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet>
- *R Markdown* Reference Guide - <https://www.rstudio.com/wp-content/uploads/2015/03/rmarkdown-reference.pdf>
- Introduction to dplyr - <https://cran.rstudio.com/web/packages/dplyr/vignettes/introduction.html>
- ggplot2 Documentation - <http://docs.ggplot2.org/current/>

Chapter 3

Mathematics and Science

Math

T_EX is the best way to typeset mathematics. Donald Knuth designed T_EX 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.

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

CHEMISTRY 101: SYMBOLS

your formula inside the curly brackets. (Notice the use of the backticks here which enclose text that acts as code.)

So, $\text{Fe}_2^{2+}\text{Cr}_2\text{O}_4$ 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: $\text{CuCl} \bullet 7\text{H}_2\text{O}$

Delta: Δ

Reaction Arrows: \longrightarrow or $\xrightarrow{\text{solution}}$

Resonance Arrows: \leftrightarrow

Reversible Reaction Arrows: \rightleftharpoons

Typesetting reactions

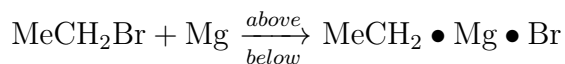
You may wish to put your reaction in an equation environment, which means that LaTeX will place the reaction where it fits and will number the equations for you.



We can reference this combustion of glucose reaction via Equation (3.1).

Other examples of reactions





Physics

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

Biology

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

Chapter 4

Tables, Graphics, References, and Labels

Tables

By far the easiest way to present tables in your thesis is to store the contents of the table in a CSV or Excel file, then read that file in to your R Markdown document as a data frame. Then you can style the table with the `kable` function, or functions in the `kableExtra` package.

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 <http://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. Generally I don't recommend this approach of typing the table directly into your R Markdown document.

Table 4.1: 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 4.1. 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 2.1. 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.

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 `uw.png` in our main directory. We then give it the caption of "UW logo", the label of "uwlogo", 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/uw.png")
```

Here is a reference to the UW logo: Figure 4.1. 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`.



Figure 4.1: UW logo

FIGURES

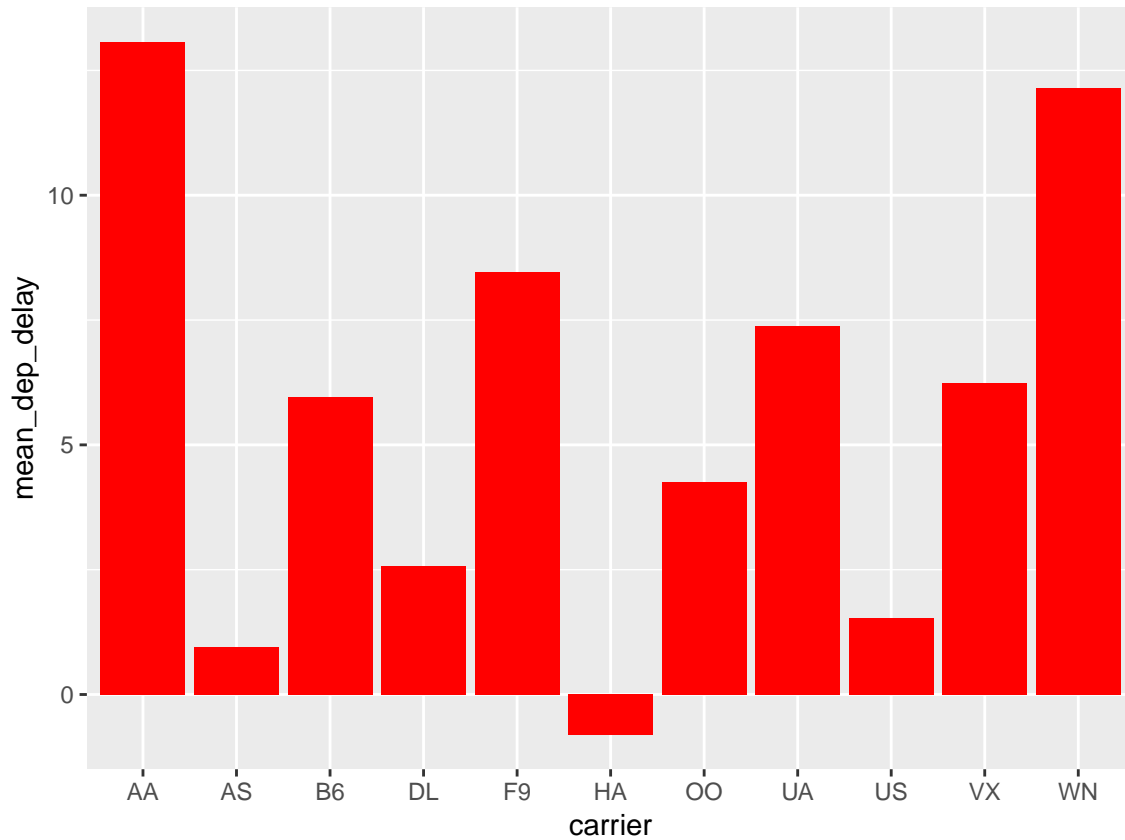


Figure 4.2: Mean Delays by Airline

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 2. (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. Note also the use of the `scale` parameter which is discussed on the next page.

```
flights %>% group_by(carrier) %>%  
  summarize(mean_dep_delay = mean(dep_delay)) %>%  
  ggplot(aes(x = carrier, y = mean_dep_delay)) +  
  geom_bar(position = "identity", stat = "identity", fill = "red")
```

Here is a reference to this image: Figure 4.2.

CHAPTER 4. TABLES AND STUFF

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

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. Some Zotero documentation is at <http://libguides.reed.edu/citation/zotero>. In addition, a tutorial is available from Middlebury College at <http://sites.middlebury.edu/zoteromiddlebury/>.

R Markdown uses *pandoc* (<http://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: (???). 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 (<http://web.reed.edu/cis/help/latex/index.html>)². There are three pages on this topic: *bibtex* (which talks about using BibTeX, at <http://web.reed.edu/cis/help/latex/bibtex.html>), *bibtexstyles* (about how to find and use the bibliography style that best suits your needs, at <http://web.reed.edu/cis/help/latex/bibtexstyles.html>) and *bibman* (which covers how to make and maintain a bibliography by hand, without BibTeX, at <http://web.reed.edu/cis/help/latex/bibman.html>). The last page will not be useful un-

²(???)

ANYTHING ELSE?

less 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.
- 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.

Anything else?

If you'd like to see examples of other things in this template, please [contact us](#) (email rapeek@ucdavis.edu) with your suggestions. We love to see people using *R Markdown* for their theses, so we'll do our best 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 readability and/or setup.

In the main Rmd file

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

In Chapter 4:

```
# This chunk ensures that the huskydown package is  
# installed and loaded. This huskydown package includes
```

APPENDIX A. THE FIRST APPENDIX

```
# the template files for the thesis and also two functions
# used for labeling and referencing
if(!require(devtools))
  install.packages("devtools", repos = "http://cran.rstudio.com")
if(!require(dplyr))
  install.packages("dplyr", repos = "http://cran.rstudio.com")
if(!require(ggplot2))
  install.packages("ggplot2", repos = "http://cran.rstudio.com")
if(!require(ggplot2))
  install.packages("bookdown", repos = "http://cran.rstudio.com")
if(!require(aggiedown)){
  library(devtools)
  devtools::install_github("rapeek/aggiedown")
}
library(aggiedown)
flights <- read.csv("data/flights.csv")
```

Appendix B

The Second Appendix, for Fun

Colophon

This document is set in [EB Garamond](#), [Source Code Pro](#) and [Lato](#). The body text is set at 11pt with *lmr*.

It was written in R Markdown and \LaTeX , and rendered into PDF using [aggiedown](#) and [bookdown](#).

This document was typeset using the XeTeX typesetting system, and the University of California Thesis class. Under the hood, the elements of the document formatting source code have been taken from the [Latex, Knitr, and RMarkdown templates for UC Berkeley's graduate thesis](#), and [Dissertate: a LaTeX dissertation template to support the production and typesetting of a PhD dissertation at Harvard, Princeton, and NYU](#)

The source files for this thesis, along with all the data files, have been organised into an R package, `xxx`, which is available at <https://github.com/xxx/xxx>. A hard copy of the thesis can be found in the University of XXX library.

This version of the thesis was generated on 2019-04-16 18:14:55. The repository is currently at this commit:

The computational environment that was used to generate this version is as follows:

```
- Session info -----  
setting  value
```

```

version  R version 3.5.2 (2018-12-20)
os       macOS Mojave 10.14.3
system   x86_64, darwin15.6.0
ui        X11
language (EN)
collate   en_US.UTF-8
ctype     en_US.UTF-8
tz        America/Denver
date      2019-04-16

```

- Packages -----

package	* version	date	lib	source
aggiedown	* 1.0	2019-04-16	[1]	Github (ryanpeek/aggiedown@ae99300)
assertthat	0.2.1	2019-03-21	[1]	CRAN (R 3.5.2)
backports	1.1.4	2019-04-10	[1]	CRAN (R 3.5.2)
bookdown	0.9	2018-12-21	[1]	CRAN (R 3.5.0)
callr	3.2.0	2019-03-15	[1]	CRAN (R 3.5.2)
cli	1.1.0	2019-03-19	[1]	CRAN (R 3.5.2)
colorspace	1.4-1	2019-03-18	[1]	CRAN (R 3.5.2)
crayon	1.3.4	2017-09-16	[1]	CRAN (R 3.5.0)
desc	1.2.0	2018-05-01	[1]	CRAN (R 3.5.0)
devtools	* 2.0.2	2019-04-08	[1]	CRAN (R 3.5.2)
digest	0.6.18	2018-10-10	[1]	CRAN (R 3.5.0)
dplyr	* 0.8.0.1	2019-02-15	[1]	CRAN (R 3.5.2)
evaluate	0.13	2019-02-12	[1]	CRAN (R 3.5.2)
fs	1.2.7	2019-03-19	[1]	CRAN (R 3.5.2)

APPENDIX B. THE SECOND APPENDIX, FOR FUN

ggplot2	* 3.1.1	2019-04-07	[1]	CRAN	(R 3.5.2)
glue	1.3.1	2019-03-12	[1]	CRAN	(R 3.5.2)
gtable	0.3.0	2019-03-25	[1]	CRAN	(R 3.5.2)
highr	0.8	2019-03-20	[1]	CRAN	(R 3.5.2)
htmltools	0.3.6	2017-04-28	[1]	CRAN	(R 3.5.0)
knitr	* 1.22	2019-03-08	[1]	CRAN	(R 3.5.2)
labeling	0.3	2014-08-23	[1]	CRAN	(R 3.5.0)
lazyeval	0.2.2	2019-03-15	[1]	CRAN	(R 3.5.2)
magrittr	1.5	2014-11-22	[1]	CRAN	(R 3.5.0)
memoise	1.1.0	2017-04-21	[1]	CRAN	(R 3.5.0)
munSELL	0.5.0	2018-06-12	[1]	CRAN	(R 3.5.0)
pillar	1.3.1	2018-12-15	[1]	CRAN	(R 3.5.0)
pkgbuild	1.0.3	2019-03-20	[1]	CRAN	(R 3.5.2)
pkgconfig	2.0.2	2018-08-16	[1]	CRAN	(R 3.5.0)
pkgload	1.0.2	2018-10-29	[1]	CRAN	(R 3.5.0)
plyr	1.8.4	2016-06-08	[1]	CRAN	(R 3.5.0)
png	0.1-7	2013-12-03	[1]	CRAN	(R 3.5.0)
prettyunits	1.0.2	2015-07-13	[1]	CRAN	(R 3.5.0)
processx	3.3.0	2019-03-10	[1]	CRAN	(R 3.5.2)
ps	1.3.0	2018-12-21	[1]	CRAN	(R 3.5.0)
purrr	0.3.2	2019-03-15	[1]	CRAN	(R 3.5.2)
R6	2.4.0	2019-02-14	[1]	CRAN	(R 3.5.2)
Rcpp	1.0.1	2019-03-17	[1]	CRAN	(R 3.5.2)
remotes	2.0.4	2019-04-10	[1]	CRAN	(R 3.5.2)
rlang	0.3.4	2019-04-07	[1]	CRAN	(R 3.5.2)
rmarkdown	1.12	2019-03-14	[1]	CRAN	(R 3.5.2)

rprojroot	1.3-2	2018-01-03	[1]	CRAN	(R 3.5.0)
rstudioapi	0.10	2019-03-19	[1]	CRAN	(R 3.5.2)
scales	1.0.0	2018-08-09	[1]	CRAN	(R 3.5.0)
sessioninfo	1.1.1	2018-11-05	[1]	CRAN	(R 3.5.0)
stringi	1.4.3	2019-03-12	[1]	CRAN	(R 3.5.2)
stringr	1.4.0	2019-02-10	[1]	CRAN	(R 3.5.2)
tibble	2.1.1	2019-03-16	[1]	CRAN	(R 3.5.2)
tidyselect	0.2.5	2018-10-11	[1]	CRAN	(R 3.5.0)
usethis	* 1.5.0	2019-04-07	[1]	CRAN	(R 3.5.2)
withr	2.1.2	2018-03-15	[1]	CRAN	(R 3.5.0)
xfun	0.6	2019-04-02	[1]	CRAN	(R 3.5.2)
yaml	2.2.0	2018-07-25	[1]	CRAN	(R 3.5.0)

[1] /Library/Frameworks/R.framework/Versions/3.5/Resources/library

References