

Diabetes and Public Housing

Marc Macarulay

A thesis
submitted in partial fulfillment of the
requirements for the degree of

Master of Public Health

University of Washington

2020

Committee:

Clarence Spigner, Chair

Alastair Matheson

Program Authorized to Offer Degree:

Health Services

© Copyright 2020

Marc Macarulay

University of Washington

Abstract

Diabetes and Public Housing

Marc Macarulay

Chair of the Supervisory Committee: Clarence Spigner

Health Services

“Here is my abstract”

Table of Contents

	Page
LIST OF FIGURES	ii
LIST OF TABLES	iii
CHAPTER 1: BACKGROUND AND SIGNIFICANCE	I
1.1 Problem Defintion	I
1.2 Lists	2
1.3 Line breaks	3
1.4 R chunks	3
1.5 Inline code	4
1.6 Including plots	5
1.7 Loading and exploring data	6
1.8 Additional resources	II
CHAPTER 2: METHODS	12
2.1 Study Setting and Study Design	12
CHAPTER 3: TABLES, GRAPHICS, REFERENCES, AND LABELS	13
3.1 Tables	13
3.2 Figures	15
3.3 Footnotes and Endnotes	18
3.4 Cross-referencing chapters and sections	19
3.5 Bibliographies	20
3.6 Anything else?	21

CONCLUSION	22
APPENDIX A: THE FIRST APPENDIX	23
APPENDIX B: THE SECOND APPENDIX, FOR FUN	25
COLOPHON	26
REFERENCES	30

List of Figures

Figure Number		Page
3.1	UW logo	15
3.2	Mean Delays by Airline	16
3.3	Subdiv. graph	18
3.4	A Larger Figure, Flipped Upside Down	18

List of Tables

Table Number		Page
1.1	Max Delays by Airline	9
3.1	Correlation of Inheritance Factors for Parents and Child	14

Background and Significance

1.1 PROBLEM DEFINITION

Diabetes is a chronic disease that is characterized by an inability of the body to maintain a healthy blood glucose level, this can cause a variety of symptoms that affect multiple systems in the body and can lead to potentially life-threatening complications. The key regulator hormone of glucose is insulin and it is produced in the pancreas. The absence or malfunction of insulin leads to elevated blood glucose levels called hyperglycemia. When insulin hormone is missing or ineffective the disease is called Diabetes Mellitus, this condition has multiple types.

1.1.1 DIABETES VARIANTS

The most common diabetes variants include type I diabetes mellitus, type II diabetes mellitus, and gestational diabetes. Type I diabetes is usually caused by genetic factors triggering an autoimmune reaction that results in the destruction of insulin producing cells in the pancreas. Also known as Juvenile Diabetes, the type I classification is typically diagnosed relatively early in life during childhood or early adulthood. Whereas, Type II diabetes develops when the body can still produce insulin however the amount is insufficient or when the body becomes resistant to the effects of insulin. Type II diabetes is largely attributed to lifestyle factors including obesity and physical activity levels.

This is a test to see if I can reference(Keene, Guo, & Murillo, 2018)

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).

1.2 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

1.3 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.

1.4 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

1.5 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  $2\pi$  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:

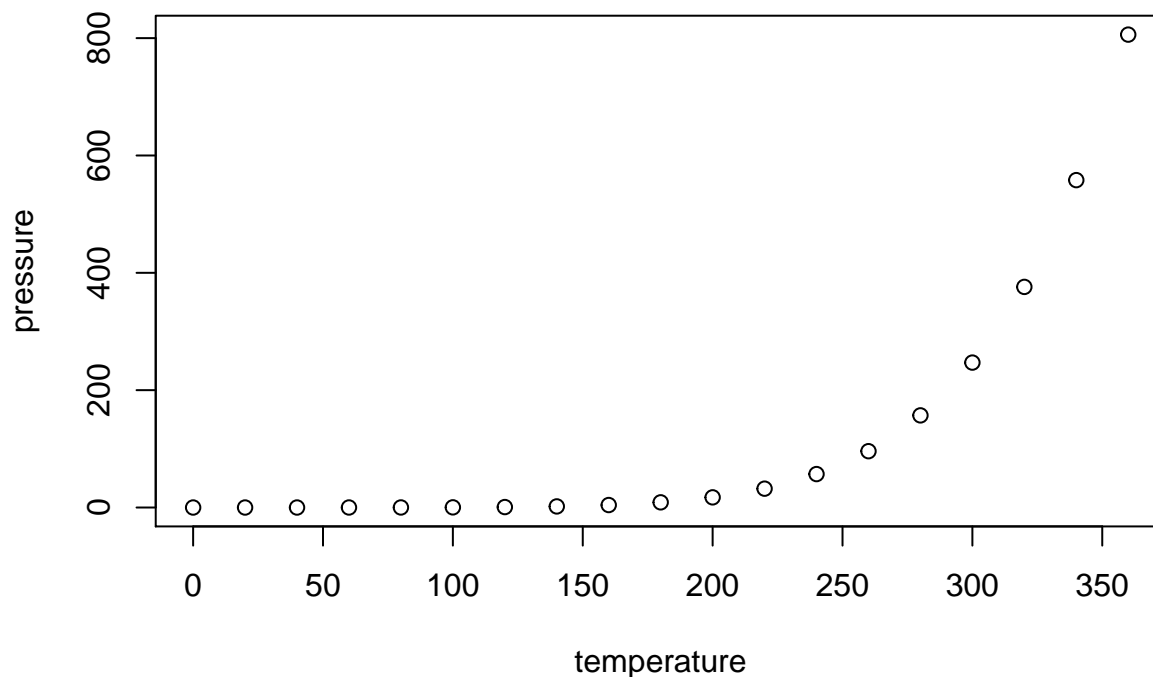
```
The standard deviation is less than 6.
```

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].

1.6 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. 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.

1.7 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"  "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 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)

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 (huskydown will load all of the packages as well)
library(huskydown)
```

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 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)
```

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

```
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 1.1: 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
Virgin America	366

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

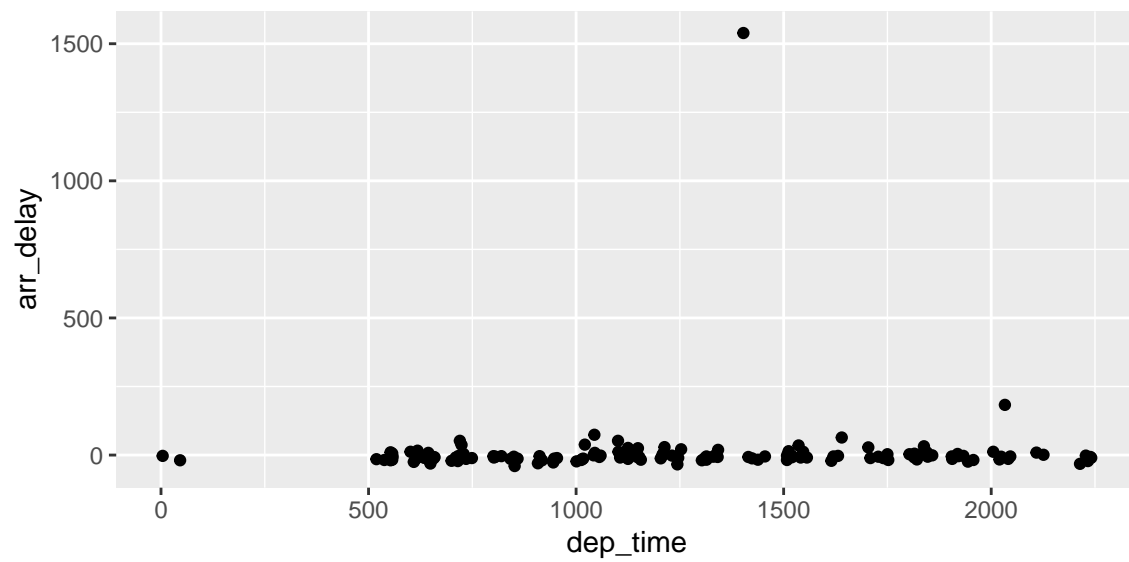
We can further look into the properties of the largest value here for American Airlines Inc. To do so, we can isolate the row corresponding to the arrival delay of 1539 minutes for American in our original `flights` dataset.

```
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()
```



1.8 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/>

2

Methods

2.1 STUDY SETTING AND STUDY DESIGN

The current study is an analysis of data compile

3

Tables, Graphics, References, and Labels

3.1 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 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 1.1. 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.



Figure 3.1: UW logo

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 `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 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`.

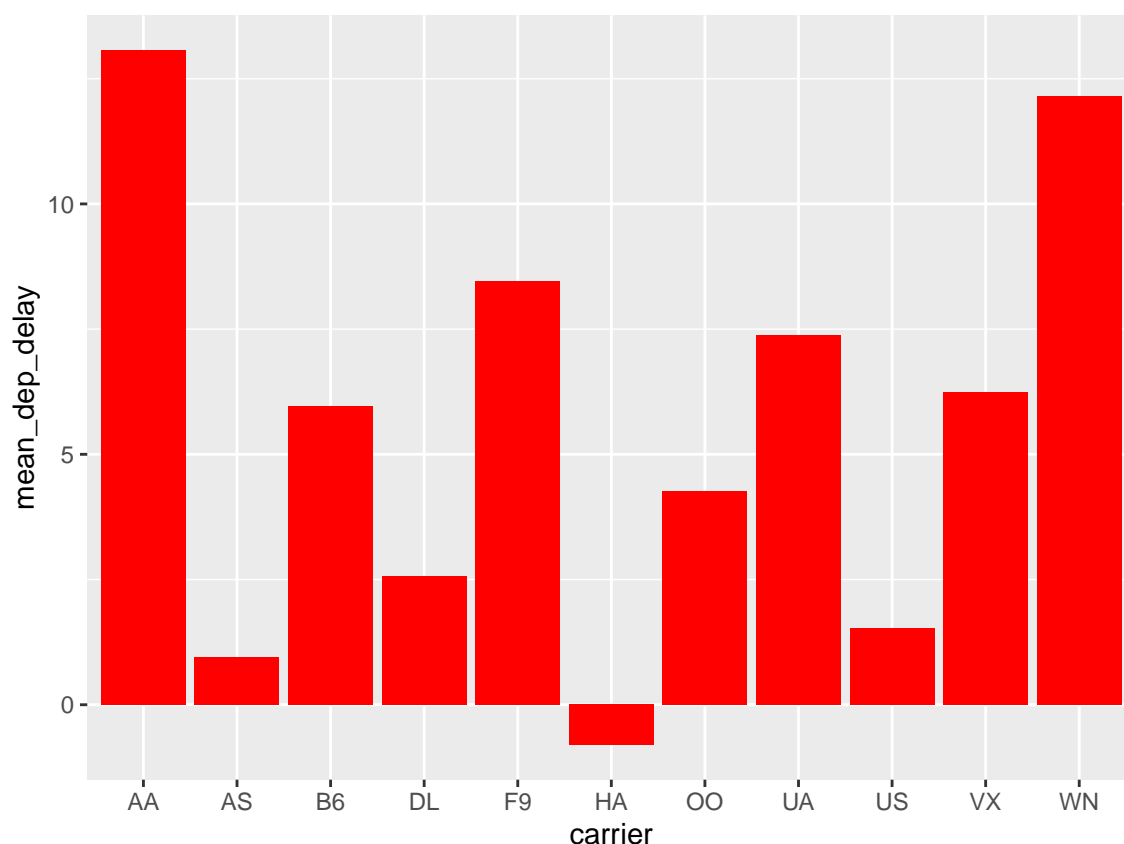


Figure 3.2: Mean Delays by Airline

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 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. 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 3.2.

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

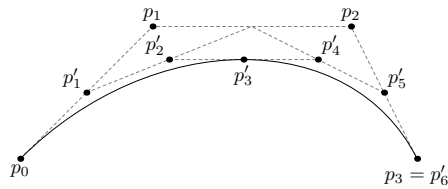


Figure 3.3: Subdiv. graph

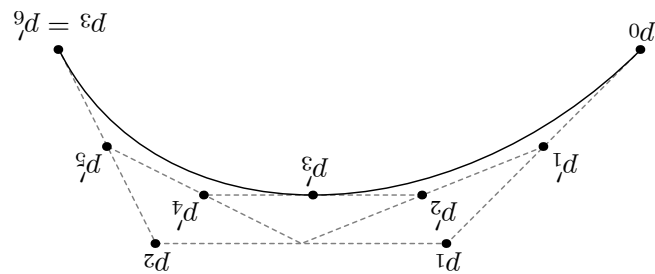


Figure 3.4: A Larger Figure, Flipped Upside Down

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. 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.) 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.

¹footnote text

3.4 CROSS-REFERENCING CHAPTERS AND SECTIONS

The [bookdown documentation](#) is an excellent source for learning how to cross-reference in a bookdown project such as a huskydown document. Here we only cover the most common uses for a typical thesis. If you want something more complex or fancy, please refer to the bookdown documentation and seek help from the developers of that package.

By default, all of your chapter and section headers will get an auto-generated ID label. For example, e.g., `# Chapter 1` will have an auto-generated ID `chapter-1`. Note that the ID label is all lower case, and has no spaces. If you have any kind of punctuation in your header, such as a colon (:), it will not appear in the ID label. Then in your text you can reference chapter one in your Rmd file like this: ‘as discussed in Chapter `\@ref(chapter-1)`’, which will print as ‘as discussed in Chapter 1’

We strongly recommend that you to manually assign ID labels to your chapter header to make it easy to cross-reference. For example, at the top of the Rmd file for this chapter, you can see:

```
# Tables, Graphics, References, and Labels {#ref-labels}
```

The `{#ref-labels}` part of this header is the ID label. It doesn’t show in the output, but is there for us to use for easy cross-referencing, because it can be short, and we don’t need to change it elsewhere our document when we update the chapter header. We can use this custom ID label in our Rmd document like this: ‘as discussed in Chapter `\@ref(ref-labels)`’, which will print as ‘as discussed in Chapter 3’. If you need to show custom text instead of the chapter number, you use this syntax in your Rmd document: `see [my chapter about labels]({#ref-labels}) for more details` which will appear as ‘see [my chapter about labels](#) for more details’

To cross-reference a specific section in the same chapter, we recommend adding a custom ID label to the section header, and using that to cross-reference. For example, earlier in this chapter we have a section on tables and in the Rmd file we see `## Tables {#tables}`. We can cross-reference that in the text like this ‘as discussed in the section on `[tables]({#tables})`’ which will appear as ‘as discussed in the above section on [tables](#)’

To cross-reference a section in a different chapter we can use the ID label from that section directly. For example, we can write in our Rmd document as discussed in the section on [R code chunks] (#r-chunks) in Chapter \@ref(rmd-basics) which will appear as ‘as discussed in the section on [R code chunks](#) in Chapter [1](#)’.

If you prefer to cross-reference by the section number, we can use custom ID labels in our Rmd document. For example, to refer to a section in our first chapter, we can write in the Rmd document: as discussed in section \@ref(r-chunks) in Chapter \@ref(rmd-basics). This will appear with section and chapter numbers like so: as ‘as discussed in section [1.4](#) in Chapter [1](#)’.

3.5 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,

²(???)

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 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.
- 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.

3.6 ANYTHING ELSE?

If you'd like to see examples of other things in this template, please [contact us](#) (email bmarwick@uw.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.

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

```

if(!require(huskydown))
  devtools::install_github("benmarwick/huskydown")
library(huskydown)

```

In Chapter 3:

```

# This chunk ensures that the huskydown package is
# installed and loaded. This huskydown package includes
# 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(huskydown)){
  library(devtools)
  devtools::install_github("benmarwick/huskydown")
}
library(huskydown)
flights <- read.csv("data/flights.csv")

```

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 `upt` with *EBGaramond*(3).

It was written in R Markdown and \LaTeX , and rendered into PDF using **huskydown** and **bookdown**.

This document was typeset using the XeTeX typesetting system, and the **University of Washington Thesis class** class created by Jim Fox. Under the hood, the **University of Washington Thesis LaTeX template** is used to ensure that documents conform precisely to submission standards. Other 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 Washington library.

This version of the thesis was generated on 2020-05-11 14:44:11. 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 3.6.1 (2019-07-05)
```

```

os      Windows 10 x64
system  x86_64, mingw32
ui       RTerm
language (EN)
collate  English_United States.1252
ctype    English_United States.1252
tz       America/Los_Angeles
date     2020-05-11

```

- Packages -----

package	* version	date	lib	source
assertthat	0.2.1	2019-03-21	[1]	CRAN (R 3.6.1)
backports	1.1.4	2019-04-10	[1]	CRAN (R 3.6.0)
bookdown	0.18.1	2020-05-01	[1]	Github (rstudio/bookdown@cd97d40)
callr	3.3.1	2019-07-18	[1]	CRAN (R 3.6.1)
cli	1.1.0	2019-03-19	[1]	CRAN (R 3.6.1)
colorspace	1.4-1	2019-03-18	[1]	CRAN (R 3.6.1)
crayon	1.3.4	2017-09-16	[1]	CRAN (R 3.6.1)
desc	1.2.0	2018-05-01	[1]	CRAN (R 3.6.2)
devtools	* 2.2.1	2019-09-24	[1]	CRAN (R 3.6.2)
digest	0.6.20	2019-07-04	[1]	CRAN (R 3.6.1)
dplyr	* 0.8.3	2019-07-04	[1]	CRAN (R 3.6.1)
ellipsis	0.3.0	2019-09-20	[1]	CRAN (R 3.6.2)
evaluate	0.14	2019-05-28	[1]	CRAN (R 3.6.1)
fs	1.3.1	2019-05-06	[1]	CRAN (R 3.6.1)
ggplot2	* 3.2.0	2019-06-16	[1]	CRAN (R 3.6.0)
git2r	0.26.1	2019-06-29	[1]	CRAN (R 3.6.2)
glue	1.3.1	2019-03-12	[1]	CRAN (R 3.6.1)

gtable	0.3.0	2019-03-25	[1]	CRAN	(R 3.6.1)
highr	0.8	2019-03-20	[1]	CRAN	(R 3.6.1)
htmltools	0.4.0	2019-10-04	[1]	CRAN	(R 3.6.2)
huskydown	* 0.0.5	2020-05-01	[1]	Github	(benmarwick/huskydown@a909835)
knitr	* 1.27	2020-01-16	[1]	CRAN	(R 3.6.2)
labeling	0.3	2014-08-23	[1]	CRAN	(R 3.6.0)
lazyeval	0.2.2	2019-03-15	[1]	CRAN	(R 3.6.1)
magrittr	1.5	2014-11-22	[1]	CRAN	(R 3.6.1)
memoise	1.1.0	2017-04-21	[1]	CRAN	(R 3.6.2)
munsell	0.5.0	2018-06-12	[1]	CRAN	(R 3.6.1)
pillar	1.4.2	2019-06-29	[1]	CRAN	(R 3.6.1)
pkgbuild	1.0.6	2019-10-09	[1]	CRAN	(R 3.6.2)
pkgconfig	2.0.2	2018-08-16	[1]	CRAN	(R 3.6.1)
pkgload	1.0.2	2018-10-29	[1]	CRAN	(R 3.6.2)
prettyunits	1.0.2	2015-07-13	[1]	CRAN	(R 3.6.1)
processx	3.4.0	2019-07-03	[1]	CRAN	(R 3.6.1)
ps	1.3.0	2018-12-21	[1]	CRAN	(R 3.6.1)
purrr	0.3.3	2019-10-18	[1]	CRAN	(R 3.6.2)
R6	2.4.0	2019-02-14	[1]	CRAN	(R 3.6.1)
Rcpp	1.0.1	2019-03-17	[1]	CRAN	(R 3.6.1)
remotes	2.1.0	2019-06-24	[1]	CRAN	(R 3.6.2)
rlang	0.4.3	2020-01-24	[1]	CRAN	(R 3.6.2)
rmarkdown	2.1	2020-01-20	[1]	CRAN	(R 3.6.3)
rprojroot	1.3-2	2018-01-03	[1]	CRAN	(R 3.6.1)
rstudioapi	0.10	2019-03-19	[1]	CRAN	(R 3.6.1)
scales	1.0.0	2018-08-09	[1]	CRAN	(R 3.6.1)
sessioninfo	1.1.1	2018-11-05	[1]	CRAN	(R 3.6.2)
stringi	1.4.3	2019-03-12	[1]	CRAN	(R 3.6.0)

stringr	1.4.0	2019-02-10	[1]	CRAN	(R 3.6.1)
testthat	2.3.1	2019-12-01	[1]	CRAN	(R 3.6.2)
tibble	2.1.3	2019-06-06	[1]	CRAN	(R 3.6.1)
tidyselect	0.2.5	2018-10-11	[1]	CRAN	(R 3.6.1)
usethis	* 1.6.1	2020-04-29	[1]	CRAN	(R 3.6.3)
withr	2.1.2	2018-03-15	[1]	CRAN	(R 3.6.1)
xfun	0.8	2019-06-25	[1]	CRAN	(R 3.6.1)
yaml	2.2.0	2018-07-25	[1]	CRAN	(R 3.6.0)

[1] C:/Users/Marc/Documents/R/win-library/3.6

[2] C:/Program Files/R/R-3.6.1/library

References

- Keene, D. E., Guo, M., & Murillo, S. (2018). “That wasn’t really a place to worry about diabetes”: Housing access and diabetes self-management among low-income adults. *Social Science & Medicine*, 197, 71–77. <http://doi.org/10.1016/J.SOCSCIMED.2017.11.051>