Appalachian State University

Honors Thesis

Writing a Reproducible Thesis with `knitr`

Author(s):
Erin Kreling, Andrew Kryzanek, Kayla Janos, Maureen O'Donnell, and Brain Pham

Supervisor:
Dr. Alan Arnholt

A thesis submitted in fulfilment of the requirements for the degree of Bachelor of Science

in the

Statistics Group
Department of Mathematical Sciences

December 2014

Declaration of Authorship

I, Erin Kreling, Andrew Kryzanek, Kayla Janos, Maureen O'Donnell, and Brain Pham, declare that this thesis titled, 'Writing a Reproducible Thesis with `knitr`' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

"Thanks to my solid academic training, today I can write hundreds of words on virtually any topic without possessing a shred of information, which is how I got a good job in journalism."

Dave Barry

APPALACHIAN STATE UNIVERSITY

Abstract

Alan T. Arnholt
Department of Mathematical Sciences

Bachelor of Science

Writing a Reproducible Thesis with `knitr`

by Erin Kreling, Andrew Kryzanek, Kayla Janos, Maureen O'Donnell, and Brain Pham

The Thesis Abstract is written here (and usually kept to just this page). The page is kept centered vertically so can expand into the blank space above the title too...

Acknowledgements

The acknowledgements and the people to thank go here, don't forget to include your project advisor…

Contents

List of Figures

List of Tables

Abbreviations

LAH **L**ist **A**bbreviations **H**ere

Physical Constants

Speed of Light	c	$=$	$2.997\ 924\ 58 \times 10^{8}\ \mathrm{ms}^{-\mathrm{S}}$ (exact)

Symbols

a	distance	m
P	power	W (Js^{-1})
ω	angular frequency	$rads^{-1}$

For/Dedicated to/To my…

Chapter 1

Introduction to LaTeX

1.1 Welcome and Thank You

Welcome to this LaTeX Thesis Template, a beautiful and easy to use template for writing a thesis using the LaTeX typesetting system.

If you are writing a thesis (or will be in the future) and its subject is technical or mathematical (though it doesn't have to be), then creating it in LaTeX is highly recommended as a way to make sure you can just get down to the essential writing without having to worry over formatting or wasting time arguing with your word processor.

LaTeX is easily able to professionally typeset documents that run to hundreds or thousands of pages long. With simple mark-up commands, it automatically sets out the table of contents, margins, page headers and footers and keeps the formatting consistent and beautiful. One of its main strengths is the way it can easily typeset mathematics, even *heavy* mathematics. Even if those equations are the most horribly twisted and most difficult mathematical problems that can only be solved on a super-computer, you can at least count on LaTeX to make them look stunning.

1.2 Learning LaTeX

LaTeX is not a WYSIWYG (What You See is What You Get) program, unlike word processors such as Microsoft Word or Apple's Pages. Instead, a document written for LaTeX is actually a simple, plain text file that contains *no formatting*. You tell LaTeX how you want the formatting in the finished document by writing in simple commands amongst the text, for example, if I want to use *italic text for emphasis*, I write the '`\textit{}`' command and put the text I want in italics in between the curly braces. This means that LaTeX is a "mark-up" language, very much like HTML.

1.2.1 A (not so short) Introduction to LaTeX

If you are new to LaTeX, there is a very good eBook – freely available online as a PDF file – called, "The Not So Short Introduction to LaTeX". The book's title is typically shortened to just "lshort". You can download the latest version (as it is occasionally updated) from here:
http://www.ctan.org/tex-archive/info/lshort/english/lshort.pdf

It is also available in several other languages. Find yours from the list on this page:
http://www.ctan.org/tex-archive/info/lshort/

It is recommended to take a little time out to learn how to use LaTeX by creating several, small 'test' documents. Making the effort now means you're not stuck learning the system when what you *really* need to be doing is writing your thesis.

1.2.2 A Short Math Guide for LaTeX

If you are writing a technical or mathematical thesis, then you may want to read the document by the AMS (American Mathematical Society) called, "A Short Math Guide for LaTeX". It can be found online here:
http://www.ams.org/tex/amslatex.html
under the "Additional Documentation" section towards the bottom of the page.

1.2.3 Common LaTeX Math Symbols

There are a multitude of mathematical symbols available for LaTeX and it would take a great effort to learn the commands for them all. The most common ones you are likely to use are shown on this page:
`http://www.sunilpatel.co.uk/latexsymbols.html`

You can use this page as a reference or crib sheet, the symbols are rendered as large, high quality images so you can quickly find the LaTeX command for the symbol you need.

1.2.4 LaTeX on a Mac

The LaTeX package is available for many systems including Windows, Linux and Mac OS X. The package for OS X is called MacTeX and it contains all the applications you need – bundled together and pre-customised – for a fully working LaTeX environment and workflow.

MacTeX includes a dedicated LaTeX IDE (Integrated Development Environment) called "TeXShop" for writing your '`.tex`' files and "BibDesk": a program to manage your references and create your bibliography section just as easily as managing songs and creating playlists in iTunes.

1.3 Getting Started with this Template

If you are familiar with LaTeX, then you can familiarise yourself with the contents of the Zip file and the directory structure and then place your own information into the '`Thesis.cls`' file. Section 1.5 on page 7 tells you how to do this. Make sure you read section 1.7 on page 9 about thesis conventions to get the most out of this template and then get started with the '`main.Rnw`' file straightaway.

If you are new to LaTeX it is recommended that you carry on reading through the rest of the information in this document.

1.3.1 About this Template

This LaTeX Thesis Template is originally based and created around a LaTeX style file created by Steve R. Gunn from the University of Southampton (UK), department of Electronics and Computer Science. You can find his original thesis style file at his site, here:
http://www.ecs.soton.ac.uk/~srg/softwaretools/document/templates/

My thesis originally used the '`ecsthesis.cls`' from his list of styles. However, I knew LaTeX could still format better. To get the look I wanted, I modified his style and also created a skeleton framework and folder structure to place the thesis files in.

This Thesis Template consists of that modified style, the framework and the folder structure. All the work that has gone into the preparation and groundwork means that all you have to bother about is the writing.

Before you begin using this template you should ensure that its style complies with the thesis style guidelines imposed by your institution. In most cases this template style and layout will be suitable. If it is not, it may only require a small change to bring the template in line with your institution's recommendations.

1.4 What this Template Includes

1.4.1 Folders

This template comes as a single Zip file that expands out to many files and folders. The folder names are mostly self-explanatory:

Appendices – this is the folder where you put the appendices. Each appendix should go into its own separate '`.Rnw`' file. A template is included in the directory.

Chapters – this is the folder where you put the thesis chapters. A thesis usually has about seven chapters, though there is no hard rule on this. Each chapter should go in its own separate '`.Rnw`' file and they usually are split as:

- Chapter 1: Introduction to the thesis topic

- Chapter 2: Background information and theory
- Chapter 3: (Laboratory) experimental setup
- Chapter 4: Details of experiment 1
- Chapter 5: Details of experiment 2
- Chapter 6: Discussion of the experimental results
- Chapter 7: Conclusion and future directions

This chapter layout is specialised for the experimental sciences.

Figures – this folder contains all figures for the thesis. These are the final images that will go into the thesis document.

Primitives – this is the folder that contains scraps, particularly because one final image in the 'Figures' folder may be made from many separate images and photos, these source images go here. This keeps the intermediate files separate from the final thesis figures.

1.4.2 Files

Included are also several files, most of them are plain text and you can see their contents in a text editor. Luckily, many of them are auxiliary files created by LaTeX or BibTeX and which you don't need to bother about:

Bibliography.bib – this is an important file that contains all the bibliographic information and references that you will be citing in the thesis for use with BibTeX. You can write it manually, but there are reference manager programs available that will create and manage it for you. Bibliographies in LaTeX are a large subject and you may need to read about BibTeX before starting with this.

Thesis.cls – this is an important file. It is the style file that tells LaTeX how to format the thesis. You will also need to open this file in a text editor and fill in your own information (such as name, department, institution). Luckily, this is not too difficult and is explained in section 1.5 on page 7.

main.pdf – this is your beautifully typeset thesis (in the PDF file format) created by LaTeX.

main.Rnw – this is an important file. This is the file that you tell `RStudio` to compile to produce your thesis as a PDF file. It contains the framework and constructs that tell LaTeXhow to layout the thesis. It is heavily commented so you can read exactly what each line of code does and why it is there. After you put your own information into the '`Thesis.cls`' file, go to this file and begin filling it in – you have now started your thesis!

vector.sty – this is a LaTeX package, it tells LaTeX how to typeset mathematical vectors. Using this package is very easy and you can read the documentation on the site (you just need to look at the '`vector.pdf`' file):

`http://www.ctan.org/tex-archive/macros/latex/contrib/vector/`

lstpatch.sty – this is a LaTeX package required by this LaTeX template and is included as not all TeX distributions have it installed by default. You do not need to modify this file.

Files that are *not* included, but are created by LaTeX as auxiliary files include:

main.aux – this is an auxiliary file generated by LaTeX, if it is deleted LaTeX simply regenerates it when you compile the '`main.Rnw`' file.

Thesis.bbl – this is an auxiliary file generated by BibTeX, if it is deleted, BibTeX simply regenerates it when you compile the '`main.Rnw`' file. Whereas the '`.bib`' file contains all the references you have, this '`.bbl`' file contains the references you have actually cited in the thesis and is used to build the bibliography section of the thesis.

main.blg – this is an auxiliary file generated by BibTeX, if it is deleted BibTeX simply regenerates it when you compile the '`main.Rnw`' file.

main.lof – this is an auxiliary file generated by LaTeX, if it is deleted LaTeX simply regenerates it when you compile the '`main.Rnw`' file. It tells LaTeX how to build the 'List of Figures' section.

main.log – this is an auxiliary file generated by LaTeX, if it is deleted LaTeX simply regenerates it when you compile the '`main.Rnw`' file. It contains messages from LaTeX, if you receive errors and warnings from LaTeX, they will be in this '`.log`' file.

main.lot – this is an auxiliary file generated by LaTeX, if it is deleted LaTeX simply regenerates it when you compile the '`main.Rnw`' file. It tells LaTeX how to build the 'List of Tables' section.

main.out – this is an auxiliary file generated by LaTeX, if it is deleted LaTeX simply regenerates it when you compile the '`main.Rnw`' file.

So from this long list, the files with the '`.sty`', '`.bib`', '`.cls`' and '`.Rnw`' extensions are the most important ones. The other auxiliary files can be ignored or deleted as LaTeX and BibTeX will regenerate them.

1.5 Filling in the '`Thesis.cls`' File

You will need to personalise the thesis template and make it your own by filling in your own information. This is done by editing the '`Thesis.cls`' file in a text editor.

Open the file and scroll down, past all the '`\newcommand`...' items until you see the entries for '`University Name`', '`Department Name`', etc....

Fill out the information about your group and institution and ensure you keep to block capitals where it asks you to. You can also insert web links, if you do, make sure you use the full URL, including the '`http://`' for this.

The last item you should need to fill in is the Faculty Name (in block capitals). When you have done this, save the file and recompile '`main.Rnw`'. All the information you filled in should now be in the PDF, complete with web links. You can now begin your thesis proper!

1.6 The '`main.Rnw`' File Explained

The `main.Rnw` file contains the structure of the thesis. There are plenty of written comments that explain what pages, sections and formatting the LaTeX code is creating. Initially there seems to be a lot of LaTeX code, but this is all formatting, and it has all been taken care of so you don't have to do it.

Begin by checking that your information on the title page is correct. For the thesis declaration, your institution may insist on something different than the text given. If this is the case, just replace what you see with what is required.

Then comes a page which contains a funny quote. You can put your own, or quote your favourite scientist, author, person, etc... Make sure to put the name of the person who you took the quote from.

Next comes the acknowledgements. On this page, write about all the people who you wish to thank (not forgetting parents, partners and your advisor/supervisor).

The contents pages, list of figures and tables are all taken care of for you and do not need to be manually created or edited. The next set of pages are optional and can be deleted since they are for a more technical thesis: insert a list of abbreviations you have used in the thesis, then a list of the physical constants and numbers you refer to and finally, a list of mathematical symbols used in any formulae. Making the effort to fill these tables means the reader has a one-stop place to refer to instead of searching the internet and references to try and find out what you meant by certain abbreviations or symbols.

The list of symbols is split into the Roman and Greek alphabets. Whereas the abbreviations and symbols ought to be listed in alphabetical order (and this is *not* done automatically for you) the list of physical constants should be grouped into similar themes.

The next page contains a one line dedication. Who will you dedicate your thesis to?

Finally, there is the section where the chapters are included. Uncomment the lines (delete the '`%`' character) as you write the chapters. Each chapter should be written in its own file and put into the 'Chapters' folder and named '`Chapter1.Rnw`', '`Chapter2.Rnw`, etc... Similarly for the appendices, uncomment the lines as you need them. Each appendix should go into its own file and placed in the 'Appendices' folder.

After the preamble, chapters and appendices finally comes the bibliography. The bibliography style (called '`unsrtnat`') is used for the bibliography and is a fully featured style that will even include links to where the referenced paper can be found online. Do not under estimate how grateful you reader will be to find that a reference to a paper is just a click away. Of course, this relies on you putting the URL information into the BibTeX file in the first place.

1.7 Thesis Features and Conventions

To get the best out of this template, there are a few conventions that you may want to follow.

One of the most important (and most difficult) things to keep track of in such a long document as a thesis is consistency. Using certain conventions and ways of doing things (such as using a Todo list) makes the job easier. Of course, all of these are optional and you can adopt your own method.

1.7.1 Printing Format

This thesis template is designed for single sided printing as most theses are printed and bound this way. This means that the left margin is always wider than the right (for binding). Four out of five people will now judge the margins by eye and think, "I never noticed that before.".

The headers for the pages contain the page number on the right side (so it is easy to flip through to the page you want) and the chapter name on the left side.

The text is set to 11 point and a line spacing of 1.3. Generally, it is much more readable to have a smaller text size and wider gap between the lines than it is to have a larger text size and smaller gap. Again, you can tune the text size and spacing should you want or need to. The text size can be set in the options for the '`\documentclass`' command at the top of the '`main.Rnw`' file and the spacing can be changed by setting a different value in the '`\setstretch`' commands (scattered throughout the '`main.Rnw`' file).

1.7.2 Using A4 Letter Paper

If you cloned the repository from https://github.com/alanarnholt/STT4870.git, then the template is defined to use `letterpaper`. If you wish to use A4 paper, change line 30 of the `main.Rnw` file to

```
\usepackage[a4paper,includehead,includefoot]{geometry}
```

The margins for the document regardless of papersize can be edited by changing the values in line 31 of the `main.Rnw` file.

1.7.3 References

The '`natbib`' package is used to format the bibliography and inserts references such as this one (Arnold et al., 1998). The options used in the '`main.Rnw`' file mean that the references are listed in numerical order as they appear in the text. Multiple references are rearranged in numerical order (e.g. (Wieman and Hollberg, 1991; Hawthorn et al., 2001)) and multiple, sequential references become reformatted to a reference range (e.g. (Wieman and Hollberg, 1991; Hawthorn et al., 2001; Arnold et al., 1998)). This is done automatically for you. To see how you use references, have a look at the '`Chapter1.Rnw`' source file. Many reference managers allow you to simply drag the reference into the document as you type.

Scientific references should come *before* the punctuation mark if there is one (such as a comma or period). The same goes for footnotes[1]. You can change this but the most important thing is to keep the convention consistent throughout the thesis. Footnotes themselves should be full, descriptive sentences (beginning with a capital letter and ending with a full stop).

To see how LaTeX typesets the bibliography, have a look at the very end of this document (or just click on the reference number links).

1.7.4 Figures

There will hopefully be many figures in your thesis (that should be placed in the 'Figures' folder). The way to insert figures into your thesis is to use a code template like this:

```
\begin{figure}[htbp]
  \centering
    \includegraphics{Figures/Electron.pdf}
    \rule{35em}{0.5pt}
  \caption[An Electron]{An electron (artist's impression)
```

[1] Such as this footnote, here down at the bottom of the page.

```
  \label{fig:Electron}}
\end{figure}
```

Also look in the source file. Putting this code into the source file produces the picture of the electron that you can see in the figure below.

FIGURE 1.1: An electron (artist's impression)

Sometimes figures don't always appear where you write them in the source. The placement depends on how much space there is on the page for the figure. Sometimes there is not enough room to fit a figure directly where it should go (in relation to the text) and so LaTeX puts it at the top of the next page. Positioning figures is the job of LaTeX and so you should only worry about making them look good!

Figures usually should have labels just in case you need to refer to them (such as in Figure 1.1). The '`\caption`' command contains two parts, the first part, inside the square brackets is the title that will appear in the 'List of Figures', and so should be short. The second part in the curly brackets should contain the longer and more descriptive caption text.

The '`\rule`' command is optional and simply puts an aesthetic horizontal line below the image. If you do this for one image, do it for all of them.

The LaTeX Thesis Template is able to use figures that are either in the PDF or JPEG file format.

1.7.5 Typesetting mathematics

If your thesis is going to contain heavy mathematical content, be sure that LaTeX will make it look beautiful, even though it won't be able to solve the equations for you.

The "Not So Short Introduction to LaTeX" (available here) should tell you everything you need to know for most cases of typesetting mathematics. If you need more information, a much more thorough mathematical guide is available from the AMS called, "A Short Math Guide to LaTeX" and can be downloaded from:
ftp://ftp.ams.org/pub/tex/doc/amsmath/short-math-guide.pdf

There are many different LaTeX symbols to remember, luckily you can find the most common symbols here. You can use the web page as a quick reference or crib sheet and because the symbols are grouped and rendered as high quality images (each with a downloadable PDF), finding the symbol you need is quick and easy.

You can write an equation, which is automatically given an equation number by LaTeX like this:

```
\begin{equation}
E = mc^{2}
  \label{eqn:Einstein}
\end{equation}
```

This will produce Einstein's famous energy-matter equivalence equation:

$$E = mc^2 \tag{1.1}$$

All equations you write (which are not in the middle of paragraph text) are automatically given equation numbers by LaTeX. If you don't want a particular equation numbered, just put the command, '`\nonumber`' immediately after the equation.

1.8 Sectioning and Subsectioning

You should break your thesis up into nice, bite-sized sections and subsections. LaTeX automatically builds a table of Contents by looking at all the '`\chapter{}`', '`\section{}`' and '`\subsection{}`' commands you write in the source.

The table of Contents should only list the sections to three (3) levels. A '`\chapter{}`' is level one (1). A '`\section{}`' is level two (2) and so a '`\subsection{}`' is level three (3). In your thesis it is likely that you will even use a '`\subsubsection{}`', which is level four (4). Adding all these will create an unnecessarily cluttered table of Contents and so you should use the '`\subsubsection*{}`' command instead (note the asterisk). The asterisk (*) tells LaTeX to omit listing the subsubsection in the Contents, keeping it clean and tidy.

1.9 In Closing

You have reached the end of this mini-guide. You can now rename or overwrite this pdf file and begin writing your own '`Chapter1.Rnw`' and the rest of your thesis. The easy work of setting up the structure and framework has been taken care of for you. It's now your job to fill it out!

Good luck and have lots of fun!

Guide written by —
Sunil Patel: www.sunilpatel.co.uk
Edited and modified for `knitr` by —
Alan T. Arnholt: www1.appstate.edu/~arnholta

Chapter 2

Using Git

What is version control, and why should you use it? Version control is a way to track files over time. By using version control, you will be able to retrace your steps to a previous working (read un-hosed) version of your files. You may be using a form of version control now with files named like the following:

- YourNameCVJanuary2014.docx
- YourNameCVMarch2014.docx
- chapter1-012412.tex
- chapter2-032312.tex

You may even back up your files for major projects in many different places. When working on book projects in the past, I would back up my files on three different local machines and two servers. That works fine until you start using the files from one location and forget that you updated the files on another machine, and you are using an old version of a file for new updates. Now you have new material on old files and may have overwritten several weeks of work. Expletives follow, and you set to "un-hosing" your work which may take longer than it took to write the original document. Is this a real scenario? Yes, and the problem only grows exponentially when working with colleagues who all have access to the same files on a major project.

I now use version control, specifically Git, for virtually all of my work. Notes for classes I am teaching have their own repositories (repos), and students and other interested parties can clone my repos. If you have material that you would like to remain private, you can set up private repositories. Thankfully, I have not lost a single file I could not recover since switching my work to Git.

One last thought before we talk about actually installing and using Git. You may be thinking, I have never lost a file because I back up all of my files on an external hardrive. Great; however, suppose you lose your machine and external hard drive due to a catastrophic event. Now what? Well, if you are not using some form of version control, your work is most likely gone for good. If you were using version control, you just need to set up a new machine and continue your work where you left off.

2.1 Downloading Git

Download and install the lastest version of Git from http://git-scm.com/downloads. Figure 2.1 on the next page shows the Git download site.

2.1.1 Mac Users

Install the downloaded file by clicking on the downloaded `*.dmg` file then clicking on the `*.pkg` file. Figure 2.2 on the following page shows the files in the Finder. If you get a message indicating the file is from an untrusted source, ignore the warning and click on the `Open` button. If there is no option to `Open`, hold down the CTRL key, select `*.pkg` file, then choose `Open With -> Installer (default)`.

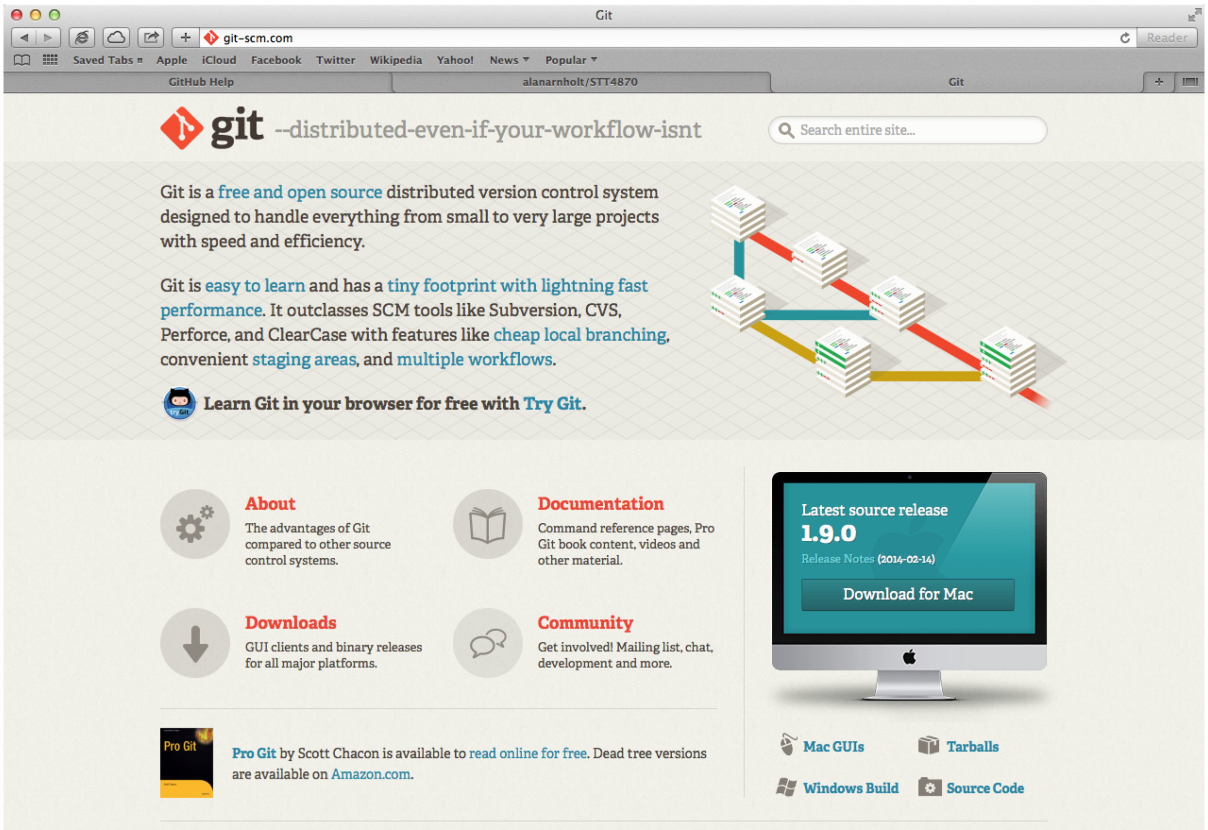


Figure 2.1: Git Download site

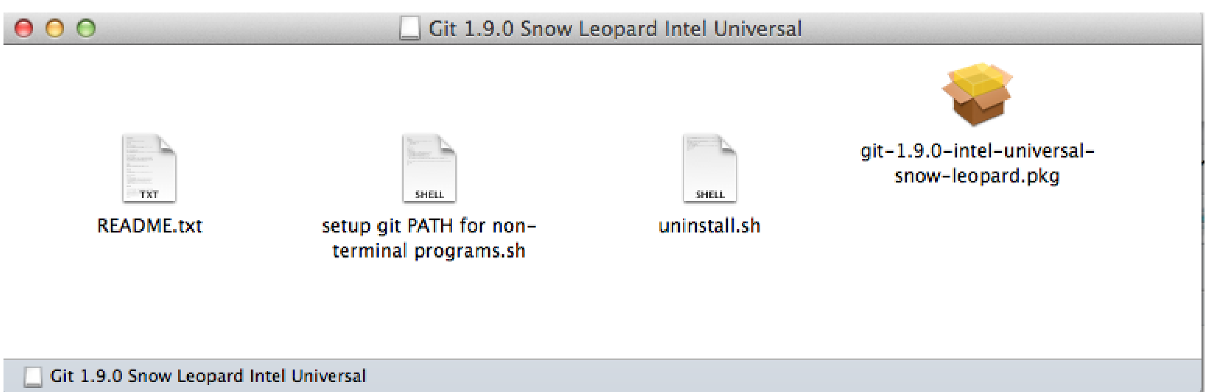


Figure 2.2: Files in Finder

2.1.2 Windows Users

Once the download is complete, right click on the downloaded file to install it as an administrator. Use the default options at each step of the installation if you are unsure of what you are doing. When the installation arrives at the screen adjusting your PATH environment, click in the circle to the left of **Run Git from the Windows Command Prompt**. You may need to add the path to where the `bash.exe` resides manually. Run the following at the `R` prompt to make sure `R` knows where to find `bash.` Note that the path below will be dependent on the operating system you are using.

```
Sys.which("bash")


       bash
"/bin/bash"
```

If the output does not specify the path to `bash,` the path to `bash` is not properly configured.

To interact with Git, find the program named Git Bash. Git Bash is the command line environment Windows uses to interact with Git. Git Bash should be located in the Git directory within your Start Menu, provided you performed a default installation.

2.2 Initial Setup

If you have never used Git before, you need to do some setup first. Run the commands in Git Example 2.1 on page 20 so that Git knows your name and email. The commands are all issued in the Terminal (Mac) or at the command prompt of Git Bash (Windows). The Terminal application is usually found in `/applications/Utilities.` A quick way to open a `terminal` window is by clicking on the magnifying glass icon and typing `terminal` in spotlight (Figure 2.3 on the following page.)

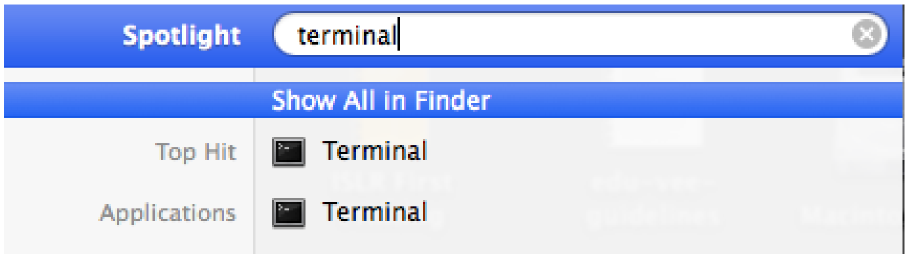


FIGURE 2.3: Spotlight

2.2.1 Mac Users

By clicking on the Terminal application, a Terminal window will open like the one in Figure 2.4.

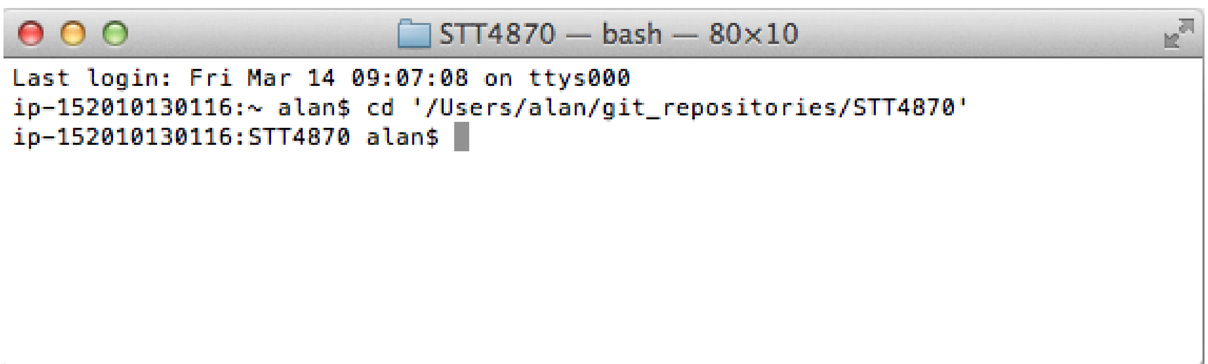


FIGURE 2.4: Terminal window

2.2.2 Windows Users

To open Git Bash, click on the `Windows icon -> Git -> Git Bash`. The program is most likely located in the Git directory within your Start Menu (or the directory into which Git was installed). By clicking on the Git Bash icon in Figure 2.5 on the following page, a window similar to Figure 2.6 on the next page will open.

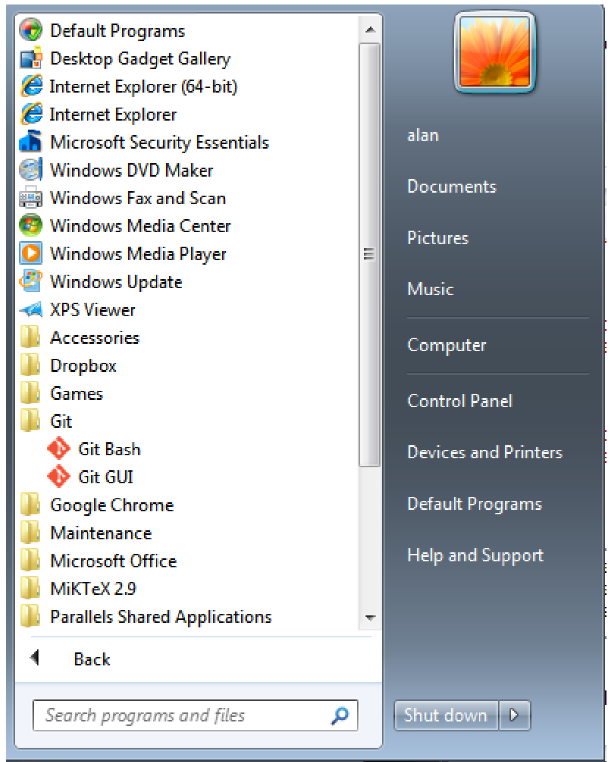


FIGURE 2.5: Windows Start Menu

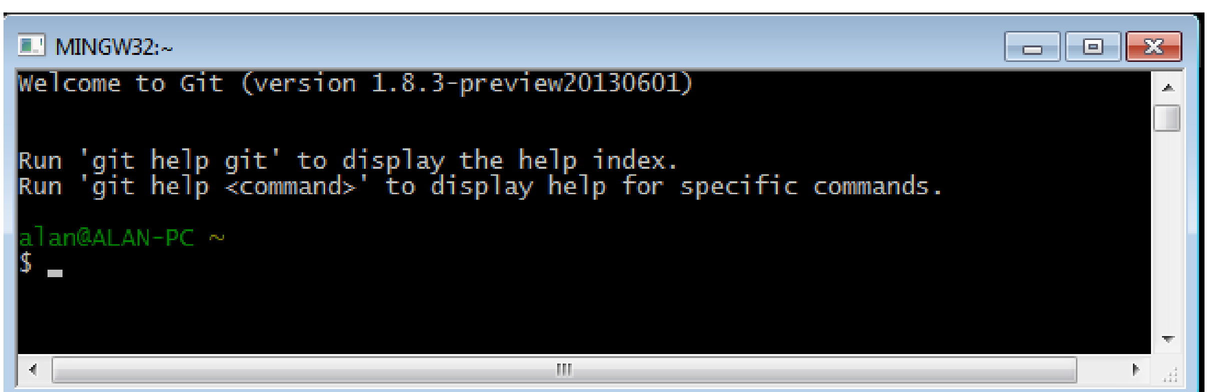


FIGURE 2.6: Git Bash Window

2.2.3 Run these commands

Git Example 2.1

```
git config --global user.name "Your Name"
git config --global user.email "your_email@whatever.com"
git config --global color.ui true
```

If you do not want to type your username and password every time you work with a remote server, you will need to install the credential helper. See the article Set Up Git for additional details on setting up the credential helper.

To confirm your username and email, type `git config --list` at the `$` prompt.

Git Example 2.2

```
git config --list  # shows your configuration


user.name=Alan Arnholt
user.email=arnholtat@appstate.edu
push.default=simple
credential.helper=osxkeychain
filter.media.clean=git-media-clean %f
filter.media.smudge=git-media-smudge %f
color.ui=true
core.repositoryformatversion=0
core.filemode=true
core.bare=false
core.logallrefupdates=true
core.ignorecase=true
remote.origin.fetch=+refs/heads/*:refs/remotes/origin/*
remote.origin.url=https://github.com/alanarnholt/STT4870.git
```

```
branch.master.remote=origin
branch.master.merge=refs/heads/master
```

From the `credential.helper` line in Git Example 2.2 on the preceding page, one can see the `credential.helper` is being used. Now that Git is set up on your computer, we provide instructions for setting up a remote repository on GitHub.

2.3 GitHub

"GitHub is a web-based hosting service for software development projects that use the Git revision control system. GitHub offers both paid plans for private repositories and free accounts for open source projects. The site was launched in 2008 by Tom Preston-Werner, Chris Wanstrath, and PJ Hyett."[1]

Qualified faculty members can request free private accounts to use with their classes. To request a discount, which actually pays the whole price for ten private repositories for qualified faculty, click on the request a discount at https://education.github.com. Qualified students can also request private accounts for their personal use. Students are given five private repositories once approved that are free of charge until the student graduates. A step-by-step guide for setting up your GitHub account for classroom use can be found at https://education.github.com/guide. Free private repositories for anyone can be created at https://bitbucket.org. Bitbucket, like GitHub, is a web based hosting service that uses the Git revision control system.

2.3.1 Creating a GitHub Account

Point your browser to https://github.com; clickon the green `Sign up for GitHub` button; type a username in the `Username` box (please use `firstlast`, for example my username is `alanarnholt`); enter your email (use your school email) in the `Email Address` box; type in your password in the `Password` box; type your password again in the `Confirm your password` box. Then, click the `Create an account` box, and you will have a GitHub account. You should

[1] http://en.wikipedia.org/wiki/GitHub

use the same name you used when you set up your `user.name` and `user.email` on your local machine.

2.3.2 Creating a GitHub Repository

In order to push your local work to a remote repository, you will first need to create the remote repository. Log in to your GitHub account; click the `New repository` button; then, give your repository a name and optionally a description (Figure 2.7.) When you finish, click the `Create repository` button, and your GitHub repository will be created. You should click in the box `Initialize this repository with a README` if you want GitHub to create a markdown README file.

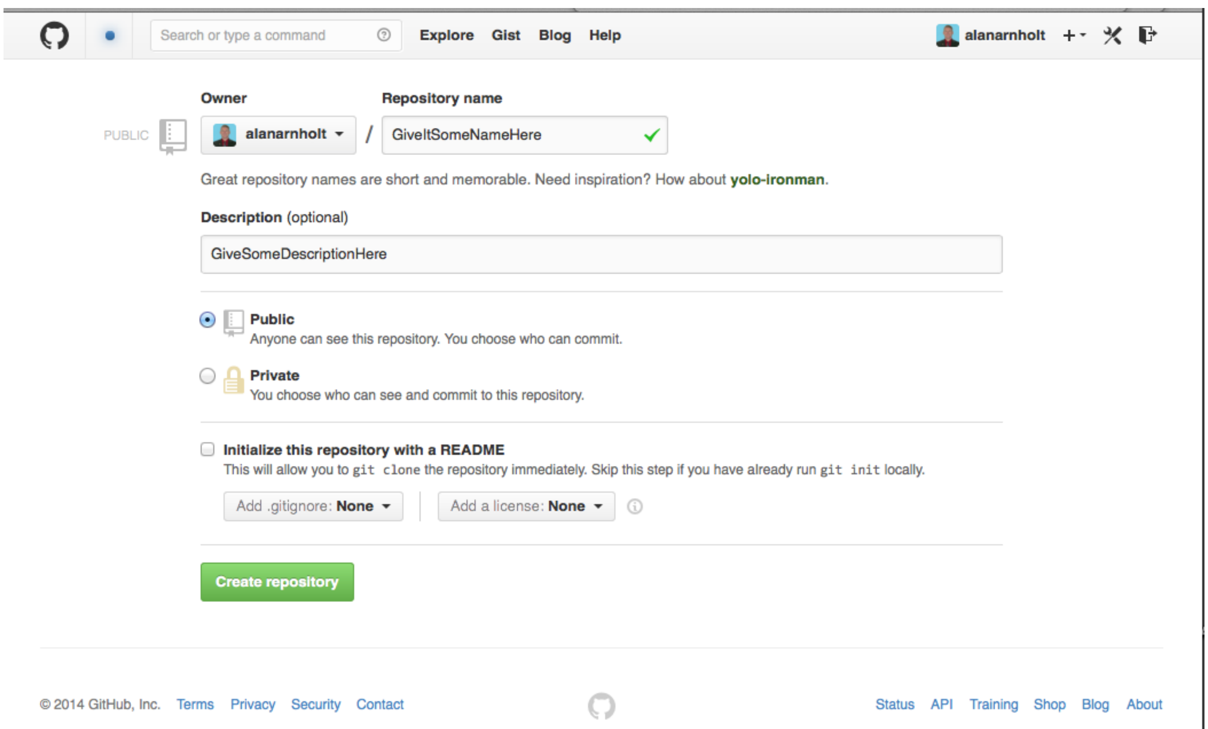


FIGURE 2.7: Create GitHub repository window

This document is stored in the repository https://github.com/alanarnholt/STT4870 in the folder https://github.com/alanarnholt/STT4870/thesis.

2.3.3 Local Repositories

Once you have your remote repository created on GitHub, you will need to create a local copy of the remote repository on your computer so that you can make changes locally. It is possible to set up a local repository using the command line or using GUI (drop, drag, etc.) commands. We start by first looking at typed commands. Then, we examine a GUI to Git.

Open either a Terminal (Mac) or Git Bash (Windows). Create a directory on your computer where you will store your copy of the GitHub (remote) repository.

```
mkdir ~/TestProject
```

The tilde (~) refers to your home directory. In other words, `~/TestProject` will create a directory called `TestProject` in your home directory. Navigate to the new directory by typing

```
cd ~/TestProject
```

Once you have a local directory with files you would like to place under version control, use the `git init` command from your working directory to track your files.

```
git init
```

Now, we are ready to point our local repository to the remote repository on GitHub by typing

```
git remote add origin https://github.com/your-user-name/TestProject.git
```

The last line needs some explanation! The `add` creates the **new remote**; the `origin` is the name for the remote; and the url is the path to the remote.

If you are working with a new repository and do not have an existing version on your computer, you need to "clone" the GitHub repo to your computer. From the working directory of your local machine, type:

```
git clone https://github.com/your-user-name/TestProject.git
```

I keep my repositories in a folder called `git_repositories` that is a subfolder of my `USERNAME` directory. If you clone a remote repository to your machine, you will not need to initialize your directory.

2.3.4 Forking a Repo

Another common way to clone a repo is by first "forking" someone else's repo. Forking a repo creates a remote (GitHub) copy of the forked repo. To work on the forked repo, you first must clone the remote fork to your local machine. When a repository is cloned, it has a default remote called `origin` that points to your fork on GitHub, not the original repository from which it was forked. This means that updates the orignal repo owner makes will not automatically be added to your forked repo. To verify that your remote (`origin`) of a forked repo is set-up properly, type

Git Example 2.3

```
git remote -v
origin  https://github.com/Your-User-Name/STT4870.git (fetch)
origin  https://github.com/Your-User-Name/STT4870.git (push)
```

The result from entering the first line of code in Git Example 2.3 should return the second and third lines with your user name in place of `Your-User-Name`.

To keep track of this repo, you need to add another remote named `upstream.` This can be done by typing

Git Example 2.4

```
git remote add upstream https://github.com/alanarnholt/STT4870.git
```

Typing the first line of code in Git Example 2.3 on the previous page after entering the code in Git Example 2.4 on the preceding page should return something similar to Git Example 2.5. That is, the second and third lines should have your user name where the url has `Your-User-Name`.

Git Example 2.5

```
git remote -v
origin  https://github.com/Your-User-Name/STT4870.git (fetch)
origin https://github.com/Your-User-Name/STT4870.git (push)
upstream  https://github.com/alanarnholt/STT4870.git (fetch)
upstream  https://github.com/alanarnholt/STT4870.git (push)
```

To pull in changes not present in your local repository without modifying your files, type

```
git fetch upstream
```

When you fetch the upstream repository, the upstream branches are stored in your local repository in a local branch named `upstream/master.` Next, you need to merge the changes into your local branch to bring your local branch in sync with the upstream branch without losing any local changes. Make sure you are on the master branch by typing `git checkout master.` Then, enter `git merge upstream/master.` Once your local branch is in sync with the upstream remote, you will want to push your local changes back to your forked repo on GitHub by typing `git push.`

Note that changes you make will not be made to the source repository unless the project maintainer "pulls" your changes after you make a pull request. Pull requests are a way to notify the project maintainer about changes in your fork of their respository. To initiate a pull request see Section 2.5 on page 34. A graphical representation of the two major collaboration modes is depicted in Figure 2.8 on the next page.

Another approach is to use

```
git pull upstream master
```

What is the difference?

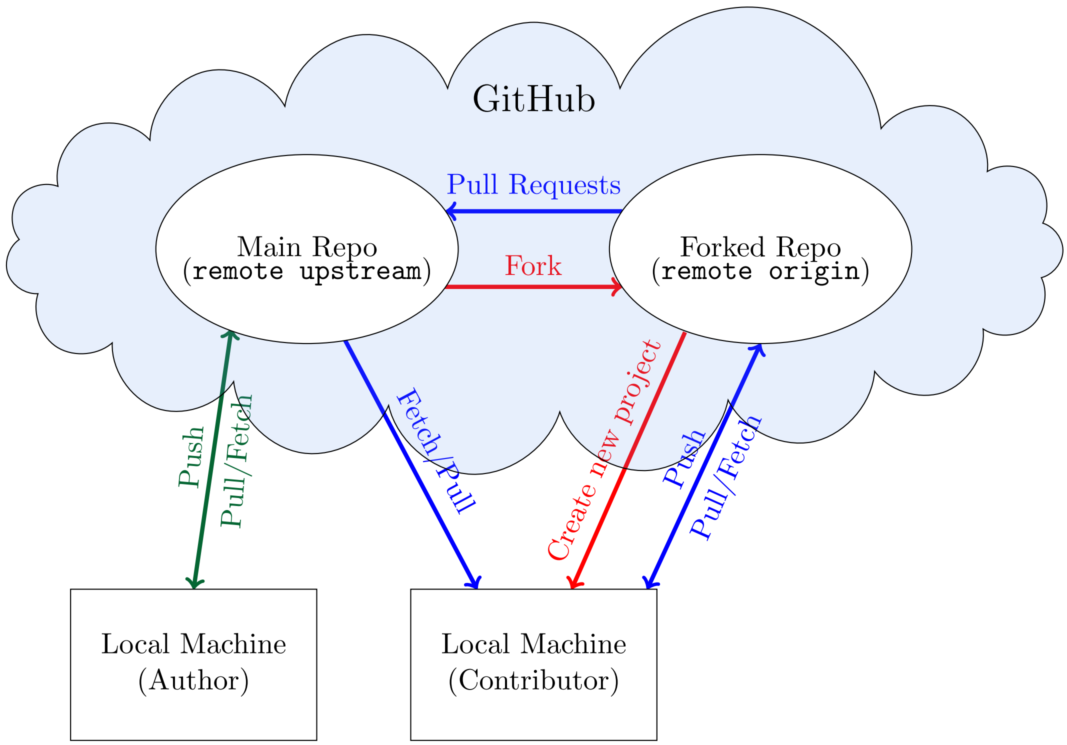


FIGURE 2.8: GitHub flow chart

2.4 Using Git with RStudio

One way to clone this repo using `RStudio` is to click on `File -> New Project` (see Figure 2.9 on the following page.) Click Version Control, and a new window such as 2.10 on the next page will appear where you will select Git. In the next window that appears, see 2.11 on the following page, enter the URL for the repository you are cloning. Enter a project name, and specify where you want the project to reside on your computer. When you are finished, click the `Create Project` button; and you will have cloned a remote repository.

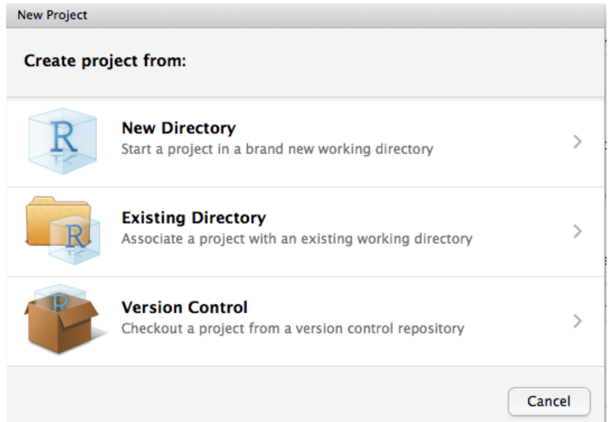


Figure 2.9: New Project window

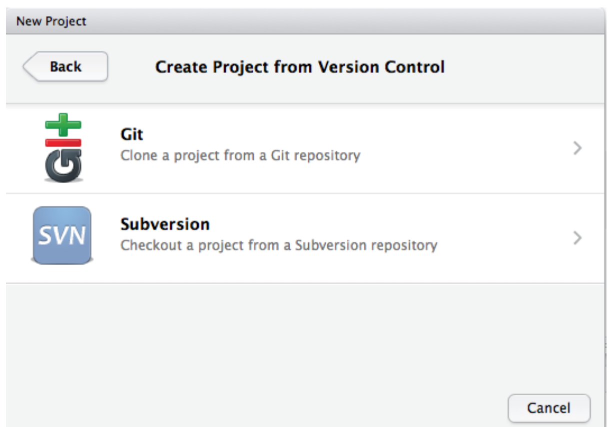


Figure 2.10: Create Project form Version Control window

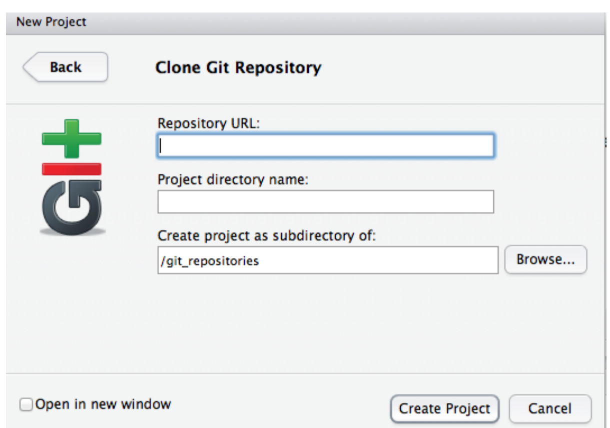


Figure 2.11: Clone Git Repository window

To check the current status of your repository type `git status` as shown in Git Example 2.6.

Git Example 2.6

```
git status
```

```
On branch master
Your branch is up-to-date with 'origin/master'.

Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

modified:   ../../slides/DeptGitSlides.md
modified:   ../Bibliography/RpksBIB.R
deleted:    ../Chapter3-tikzDictionary
modified:   ../cache/__packages
deleted:    ../figure/G1.pdf
deleted:    ../figure/G2.aux
deleted:    ../figure/G2.out
deleted:    ../figure/G2.pdf
deleted:    ../figure/G2.tikz
deleted:    ../figure/Rgraph.pdf
deleted:    ../figure/ggplot2Graphs.pdf
modified:   ../main-tikzDictionary
modified:   ../main.Rnw
modified:   ../main.bbl
modified:   ../main.lof
modified:   ../main.pdf
modified:   ../main.toc

Untracked files:
```

```
  (use "git add <file>..." to include in what will be committed)

../Bibliography/Rpkgs122414.bib

no changes added to commit (use "git add" and/or "git commit -a")
```

The `git status` command shows us what files are not staged for a commit. Before files can be committed, they must be added to the staging area. Files are added to the stating area with the command `git add file_name`. To add all files in the working directory, one can use `git add .` (The command includes the period.) Next, all files are added to the staging area, and a snapshot is taken of the commit with the message "staging all files."

```
git add .
git commit  -m "staging all files"


On branch master
Your branch is up-to-date with 'origin/master'.

Changes not staged for commit:
modified:   ../../slides/DeptGitSlides.md
modified:   ../Bibliography/RpksBIB.R
deleted:    ../Chapter3-tikzDictionary
modified:   ../cache/__packages
deleted:    ../figure/G1.pdf
deleted:    ../figure/G2.aux
deleted:    ../figure/G2.out
deleted:    ../figure/G2.pdf
deleted:    ../figure/G2.tikz
deleted:    ../figure/Rgraph.pdf
deleted:    ../figure/ggplot2Graphs.pdf
modified:   ../main-tikzDictionary
modified:   ../main.Rnw
```

```
modified:   ../main.bbl
modified:   ../main.lof
modified:   ../main.pdf
modified:   ../main.toc

Untracked files:
../Bibliography/Rpkgs122414.bib

no changes added to commit
```

Check the status after the last commit.

```
git status


On branch master
Your branch is up-to-date with 'origin/master'.

Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

modified:   ../../slides/DeptGitSlides.md
modified:   ../Bibliography/RpksBIB.R
deleted:    ../Chapter3-tikzDictionary
modified:   ../cache/__packages
deleted:    ../figure/G1.pdf
deleted:    ../figure/G2.aux
deleted:    ../figure/G2.out
deleted:    ../figure/G2.pdf
deleted:    ../figure/G2.tikz
deleted:    ../figure/Rgraph.pdf
deleted:    ../figure/ggplot2Graphs.pdf
```

```
modified:   ../main-tikzDictionary
modified:   ../main.Rnw
modified:   ../main.bbl
modified:   ../main.lof
modified:   ../main.pdf
modified:   ../main.toc

Untracked files:
  (use "git add <file>..." to include in what will be committed)

../Bibliography/Rpkgs122414.bib

no changes added to commit (use "git add" and/or "git commit -a")
```

Push changes to the remote repository.

```
git push


Everything up-to-date
```

See if there is anything left to do.

```
git status


On branch master
Your branch is up-to-date with 'origin/master'.

Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

modified:   ../../slides/DeptGitSlides.md
```

```
modified:   ../Bibliography/RpksBIB.R
deleted:    ../Chapter3-tikzDictionary
modified:   ../cache/__packages
deleted:    ../figure/G1.pdf
deleted:    ../figure/G2.aux
deleted:    ../figure/G2.out
deleted:    ../figure/G2.pdf
deleted:    ../figure/G2.tikz
deleted:    ../figure/Rgraph.pdf
deleted:    ../figure/ggplot2Graphs.pdf
modified:   ../main-tikzDictionary
modified:   ../main.Rnw
modified:   ../main.bbl
modified:   ../main.lof
modified:   ../main.pdf
modified:   ../main.toc

Untracked files:
  (use "git add <file>..." to include in what will be committed)

../Bibliography/Rpkgs122414.bib

no changes added to commit (use "git add" and/or "git commit -a")
```

Show the last three commits with

```
git log  -3


commit 78b4951ab63beda9cd648f05cbb2ca258d079490
Author: Alan Arnholt <arnholtat@appstate.edu>
Date:   Wed Dec 24 10:59:12 2014 -0500
```

```
    staging all files

commit 73b0b65cf89352bc62ae3802428b19743d9bacee
Author: Alan Arnholt <arnholtat@appstate.edu>
Date:   Wed Dec 24 10:44:53 2014 -0500

    staging all files

commit 98f8fb3714072878a56ba5e09ed31c39bcd4315d
Author: Alan Arnholt <arnholtat@appstate.edu>
Date:   Sat Jun 7 16:20:18 2014 -0400

    problem last time with pdf missing gggplot2 package
```

That was ugly. Let us try some formatting.

```
git log --pretty=oneline -3


78b4951ab63beda9cd648f05cbb2ca258d079490 staging all files
73b0b65cf89352bc62ae3802428b19743d9bacee staging all files
98f8fb3714072878a56ba5e09ed31c39bcd4315d problem last time with pdf missing gggplot2packag
```

The previous output was too brief to suit me. Let us try some further formatting.

```
git log --pretty=format:"%h %ad- %s [%an]" -3


78b4951 Wed Dec 24 10:59:12 2014 -0500- staging all files [Alan Arnholt]
73b0b65 Wed Dec 24 10:44:53 2014 -0500- staging all files [Alan Arnholt]
98f8fb3 Sat Jun 7 16:20:18 2014 -0400- problem last time with pdf missing gggplot2 package
```

Maybe even some statistics?

```
git log --pretty=format:"%h %ad- %s [%an]" -3 --stat

78b4951 Wed Dec 24 10:59:12 2014 -0500- staging all files [Alan Arnholt]
 thesis/Chapters/Chapter2.Rnw | 2 +-
 thesis/Chapters/Chapter3.Rnw | 2 +-
 2 files changed, 2 insertions(+), 2 deletions(-)

73b0b65 Wed Dec 24 10:44:53 2014 -0500- staging all files [Alan Arnholt]
 thesis/Chapters/Chapter3.Rnw | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)

98f8fb3 Sat Jun 7 16:20:18 2014 -0400- problem last time with pdf missing gggplot2 package
 thesis/figure/G2.pdf            | Bin 49710 -> 49710 bytes
 thesis/figure/G2.tikz           |   2 +-
 thesis/figure/Rgraph.pdf        | Bin 5489 -> 5493 bytes
 thesis/figure/ggplot2Graphs.pdf | Bin 36900 -> 36913 bytes
 thesis/main-tikzDictionary      | Bin 0 -> 2300 bytes
 thesis/main.lof                 |   2 +-
 thesis/main.pdf                 | Bin 897094 -> 938422 bytes
 thesis/main.toc                 |  16 ++++++++--------
 8 files changed, 10 insertions(+), 10 deletions(-)
```

2.5 So you want to collaborate?

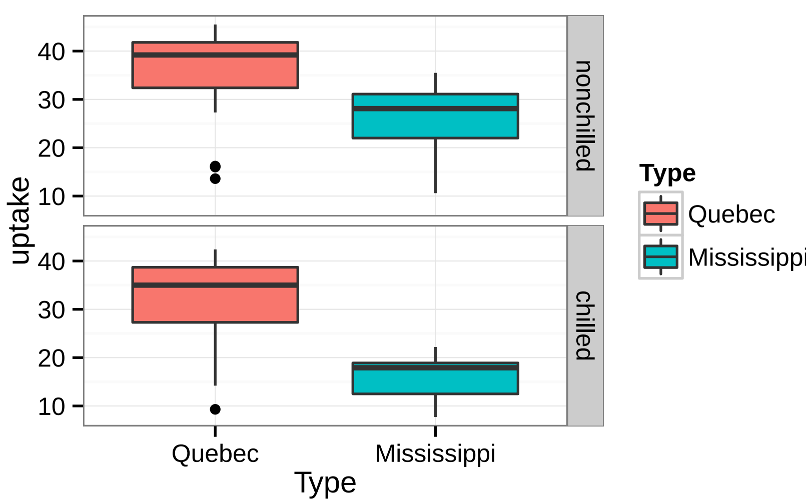
At this point, you have forked a repo and would like to contribute to someone's project. A great place to start is by reading https://help.github.com/articles/using-pull-requests.

Chapter 3

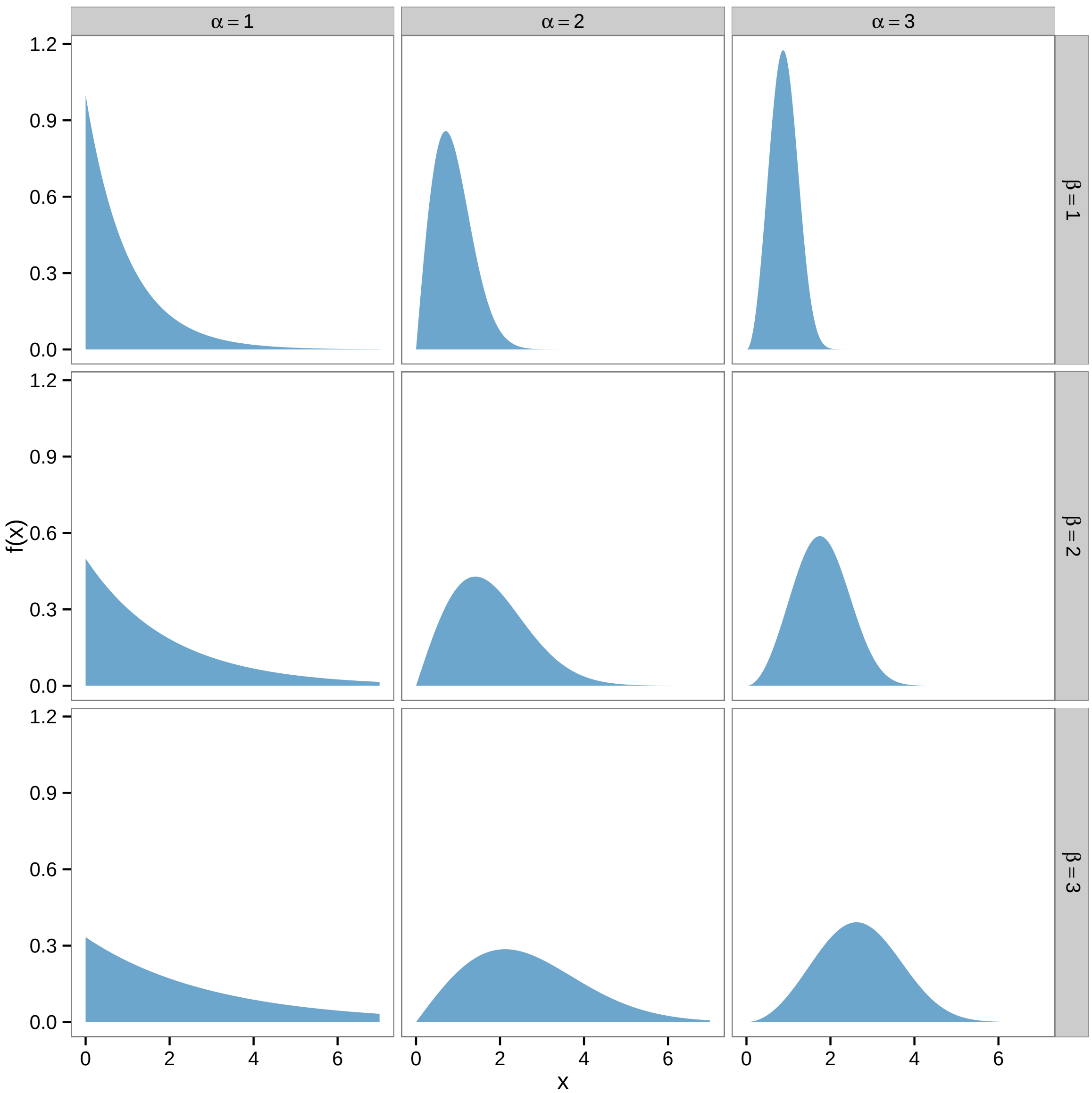
Using R

Now, just to show how cool this is, we will mix in a little R. First, consider the following graph where the R code that creates the graph is shown to the reader.

```
library(ggplot2)
ggplot(data = CO2, aes(x = Type, y = uptake, fill = Type)) +
    geom_boxplot() + facet_grid(Treatment ~ .) + theme_bw()
```



I love graphs! The following graph created with `ggplot2` (Wickham and Chang, 2014b) uses Greek letters in the facet panels. The `R` Code used to create the graph is not shown in the final document. The code is hidden using the argument `echo = FALSE` in the `R` code chunk.



The graphs shown so far have fonts that do not match the font of the thesis. To create graphs that have the same font as the thesis, use the `dev = "tikz"` option in the R code chunk.

Figure 3.1 uses the `dev = "tikz"` option so that the fonts in the graph are in agreement with the document fonts.

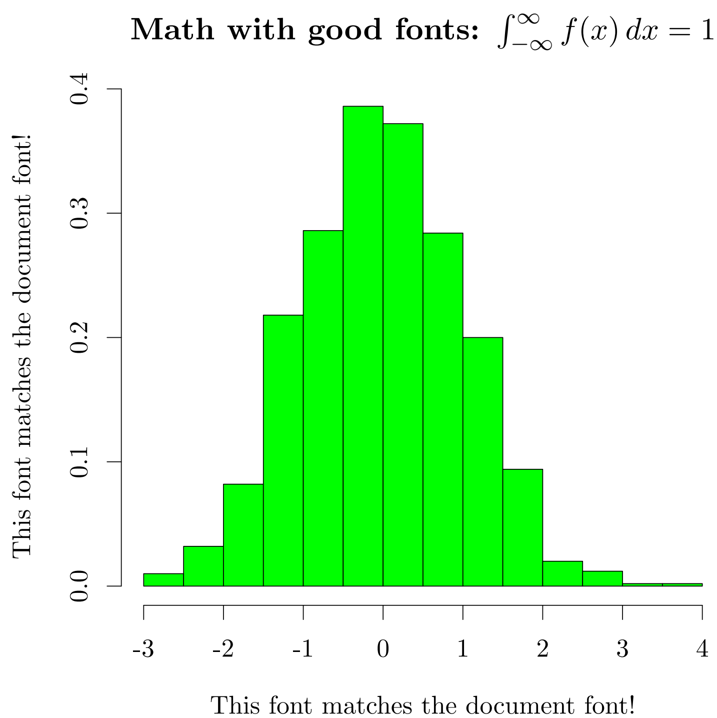


FIGURE 3.1: Something goes here

See R Code 3.1 which labels some `R` Code.

R Code 3.1

```
set.seed(13)
stuff <- rnorm(100, 100, 10)
qs <- qnorm(seq(0, 1, by = 0.1), 100, 10)
OB <- cut(stuff, breaks = qs)
T1 <- xtabs(~OB)
OBS <- as.vector(T1)
EXP <- rep(10, 10)
```

```
X2 <- sum((OBS - EXP)^2/EXP)
X2
```

```
[1] 5
```

```
pvalue <- pchisq(X2, 9, lower = FALSE)
pvalue
```

```
[1] 0.8343083
```

Chapter 4

Using `xtable`

Chapter 5

Using BibTEX

5.1 Bibliographies with BibTEX

5.2 How to use citations

5.3 Generating a BibTEX file of R packages

5.4 Using BibDesk

Appendix A

Appendix Title Here

Write your Appendix content here.

Bibliography

Arnholt, A. T. (2014). *PASWR2: Probability and Statistics with R, Second Edition.* R package version 1.0.

Arnold, A. S., J. S. Wilson, and M. G. Boshier (1998, March). A simple extended-cavity diode laser. *Review of Scientific Instruments 69*(3), 1236–1239.

Auguie, B. (2012). *gridExtra: functions in Grid graphics.* R package version 0.9.1.

Boettiger, C. (2014). *knitcitations: Citations for knitr markdown files.* R package version 1.0.5.

Canty, A. and B. Ripley (2014). *boot: Bootstrap Functions (originally by Angelo Canty for S).* R package version 1.3-13.

Dahl, D. B. (2014). *xtable: Export tables to LaTeX or HTML.* R package version 1.7-4.

Dragulescu, A. A. (2014a). *xlsx: Read, write, format Excel 2007 and Excel 97/2000/XP/2003 files.* R package version 0.5.7.

Dragulescu, A. A. (2014b). *xlsxjars: Package required POI jars for the xlsx package.* R package version 0.6.1.

Faraway, J. (2014). *faraway: Functions and datasets for books by Julian Faraway.* R package version 1.0.6.

Francois, R. (2013). *bibtex: bibtex parser.* R package version 0.3-6.

Friendly, M. (2014). *vcdExtra: vcd extensions and additions.* R package version 0.6-3.

Gandrud, C. (2013). *Reproducible Research with R and RStudio.* CRC Press.

Gandrud, C. (2014). *repmis: A collection of miscellaneous tools for reproducible research with R.* R package version 0.3.3.

Genz, A., F. Bretz, T. Miwa, X. Mi, and T. Hothorn (2014). *mvtnorm: Multivariate Normal and t Distributions.* R package version 1.0-2.

Grätzer, G. and R. Schöpf (2007). *More math into latex*, Volume 4. Springer.

Graves, S., H.-P. Piepho, and L. S. with help from Sundar Dorai-Raj (2012). *multcompView: Visualizations of Paired Comparisons.* R package version 0.1-5.

Gross, J. and bug fixes by Uwe Ligges (2012). *nortest: Tests for Normality.* R package version 1.0-2.

Harrell, Jr., F. E. (2014). *Hmisc: Harrell Miscellaneous.* R package version 3.14-6.

Hawthorn, C. J., K. P. Weber, and R. E. Scholten (2001, December). Littrow configuration tunable external cavity diode laser with fixed direction output beam. *Review of Scientific Instruments 72*(12), 4477–4479.

Hothorn, T., F. Bretz, and P. Westfall (2014). *multcomp: Simultaneous Inference in General Parametric Models.* R package version 1.3-8.

Hothorn, T., K. Hornik, M. A. van de Wiel, and A. Zeileis (2014). *coin: Conditional Inference Procedures in a Permutation Test Framework.* R package version 1.0-24.

Lumley, T. (2009). *leaps: regression subset selection.* R package version 2.9.

Meyer, D., E. Dimitriadou, K. Hornik, A. Weingessel, and F. Leisch (2014). *e1071: Misc Functions of the Department of Statistics (e1071), TU Wien.* R package version 1.6-4.

Meyer, D., A. Zeileis, and K. Hornik (2014). *vcd: Visualizing Categorical Data.* R package version 1.3-2.

Pinheiro, J., D. Bates, and R-core (2014). *nlme: Linear and Nonlinear Mixed Effects Models.* R package version 3.1-118.

Qiu, Y. and Y. Xie (2014). *highr: Syntax highlighting for R.* R package version 0.4.

R Core Team (2014a). *foreign: Read Data Stored by Minitab, S, SAS, SPSS, Stata, Systat, Weka, dBase, ...* R package version 0.8-61.

R Core Team (2014b). *R: A Language and Environment for Statistical Computing.* Vienna, Austria: R Foundation for Statistical Computing.

Rinker, T. (2014). *reports: Assist the Workflow of Writing Academic Articles and Other Reports.* R package version 0.1.4.

Ripley, B. (2014). *MASS: Support Functions and Datasets for Venables and Ripley's MASS.* R package version 7.3-35.

RStudio (2012). *rstudio: Tools and Utilities for RStudio.* R package version 0.98.1091.

RStudio, Inc. (2014). *shiny: Web Application Framework for R.* R package version 0.10.2.2.

Sarkar, D. (2014). *lattice: Lattice Graphics.* R package version 0.20-29.

Sarkar, D. and F. Andrews (2013). *latticeExtra: Extra Graphical Utilities Based on Lattice.* R package version 0.6-26.

Sharpsteen, C. and C. Bracken (2013). *tikzDevice: R Graphics Output in LaTeX Format.* R package version 0.7.0.

Spector, P. (2008). *Data manipulation with R.* Springer.

Technology, T. and LLC. (2013). *shinyAce: Ace editor bindings for Shiny.* R package version 0.1.0.

Temple Lang, D. (2014). *RCurl: General network (HTTP/FTP/...) client interface for R.* R package version 1.95-4.5.

Wand, M. (2014). *KernSmooth: Functions for kernel smoothing for Wand & Jones (1995).* R package version 2.23-13.

Wickham, H. (2012). *gtable: Arrange grobs in tables.* R package version 0.1.2.

Wickham, H. (2014a). *httr: Tools for Working with URLs and HTTP.* R package version 0.6.0.

Wickham, H. (2014b). *plyr: Tools for splitting, applying and combining data.* R package version 1.8.1.

Wickham, H. (2014c). *reshape2: Flexibly Reshape Data: A Reboot of the Reshape Package.* R package version 1.4.1.

Wickham, H. and W. Chang (2014a). *devtools: Tools to make developing R code easier.* R package version 1.6.1.

Wickham, H. and W. Chang (2014b). *ggplot2: An implementation of the Grammar of Graphics.* R package version 1.0.0.

Wickham, H., P. Danenberg, and M. Eugster (2014). *roxygen2: In-source Documentation for R.* R package version 4.1.0.

Wieman, C. E. and L. Hollberg (1991, January). Using diode lasers for atomic physics. *Review of Scientific Instruments 62*(1), 1–20.

Xie, Y. (2013). *Dynamic Documents with R and knitr.* CRC Press.

Xie, Y. (2014a). *formatR: Format R Code Automatically.* R package version 1.0.

Xie, Y. (2014b). *knitr: A General-Purpose Package for Dynamic Report Generation in R.* R package version 1.8.