

APPALACHIAN STATE UNIVERSITY

Departmental Honors Thesis

SOME DESCRIPTIVE TITLE WITH LOGISTIC REGRESSION

Author:

CAROLINE FEHLMAN

Supervisor:

ALAN T. ARNHOLT

*Submitted to the Department of Mathematical Sciences
in partial fulfillment of the requirements for the degree of
Bachelor of Science in General Mathematics*

in the

Statistics Group
Department of Mathematical Sciences

August 2022

Approved by:

Alan T. Arnholt, Ph.D., Thesis Director
Department of Mathematical Sciences

William Cook, Ph.D., Honors Director
Department of Mathematical Sciences

Eric Marland, Ph.D., Chair
Department of Mathematical Sciences

Declaration of Authorship

I, CAROLINE FEHLMAN, declare that this thesis titled, 'SOME DESCRIPTIVE TITLE WITH LOGISTIC REGRESSION' 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:

Abstract

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

Declaration of Authorship	ii
Abstract	iii
Acknowledgements	iv
Contents	v
List of Figures	vii
List of Tables	viii
Abbreviations	ix
Physical Constants	x
Symbols	xi
1 Literature Review	1
1.1 Some Descriptive Title	1
2 Using Git	2
2.1 Downloading Git	3
2.1.1 Mac Users	3
2.1.2 Windows Users	3
2.2 Initial Setup	5
2.2.1 Mac Users	5
2.2.2 Windows Users	6

2.2.3	Run these commands	8
2.3	GitHub	9
2.3.1	Creating a GitHub Account	9
2.3.2	Creating a GitHub Repository	10
2.3.3	Local Repositories	11
2.3.4	Forking a Repo	12
2.4	Using Git with RStudio	15
2.5	So you want to collaborate?	23
3	Using R	24
4	Using xtable	27
5	Using BibT_EX	28
5.1	Bibliographies with BibT _E X	28
5.2	How to use citations	28
5.3	Generating a BibT _E X file of R packages	28
5.4	Using BibDesk	28
A	Appendix Title Here	29
	Bibliography	30

List of Figures

2.1	Git Download site	4
2.2	Files in Finder	4
2.3	Spotlight	5
2.4	Terminal window	6
2.5	Windows Start Menu	7
2.6	Git Bash Window	7
2.7	Create GitHub repository window	10
2.8	GitHub flow chart	14
2.9	New Project window	15
2.10	Create Project form Version Control window	15
2.11	Clone Git Repository window	16

List of Tables

Abbreviations

LAH List Abbreviations **Here**

Physical Constants

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

Symbols

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

For/Dedicated to/To my...

Chapter 1

Literature Review

1.1 Some Descriptive Title

The first person to consider this really interesting topic was [Gandrud \(2013\)](#).

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 harddrive. 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 latest version of Git from <http://git-scm.com/downloads>.

Figure 2.1 on the following 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 next 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)**.

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**

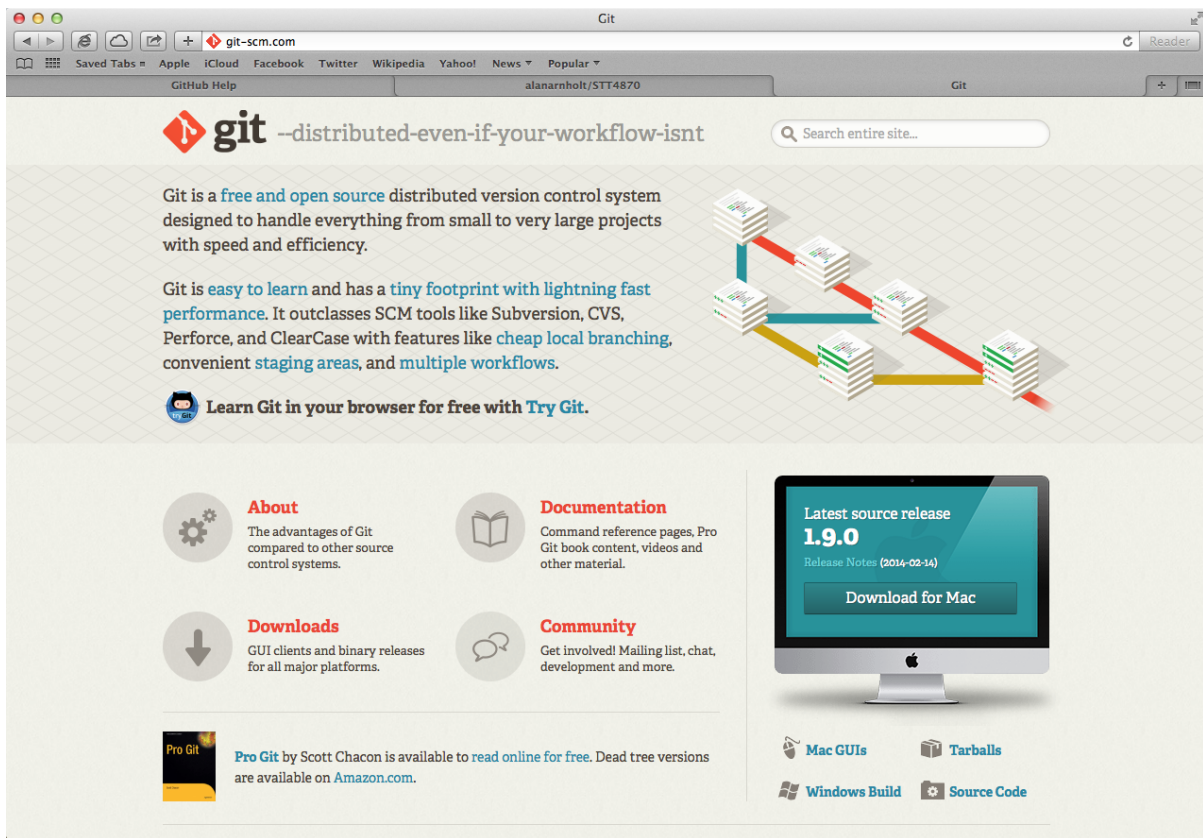


FIGURE 2.1: Git Download site

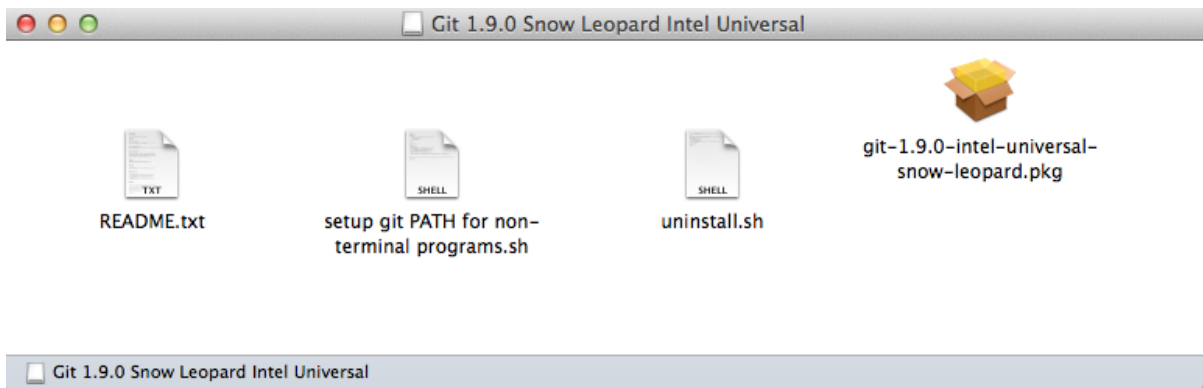


FIGURE 2.2: Files in Finder

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  
"/usr/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 8 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.)

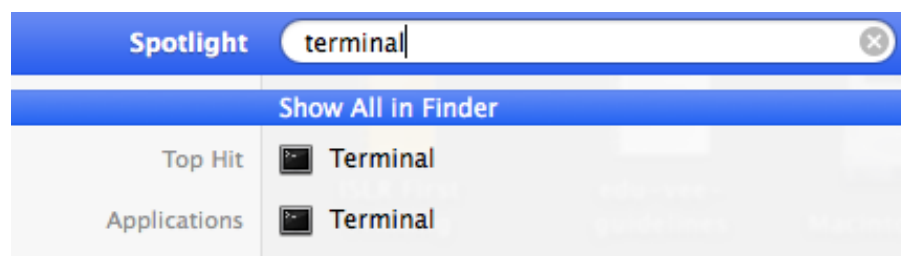


FIGURE 2.3: Spotlight

2.2.1 Mac Users

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

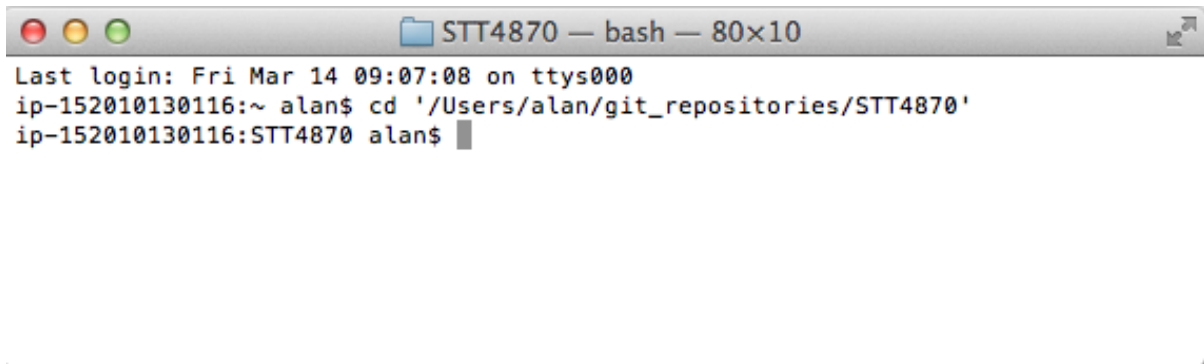


FIGURE 2.4: Terminal window

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 next page](#), a window similar to [Figure 2.6 on the following page](#) will open.

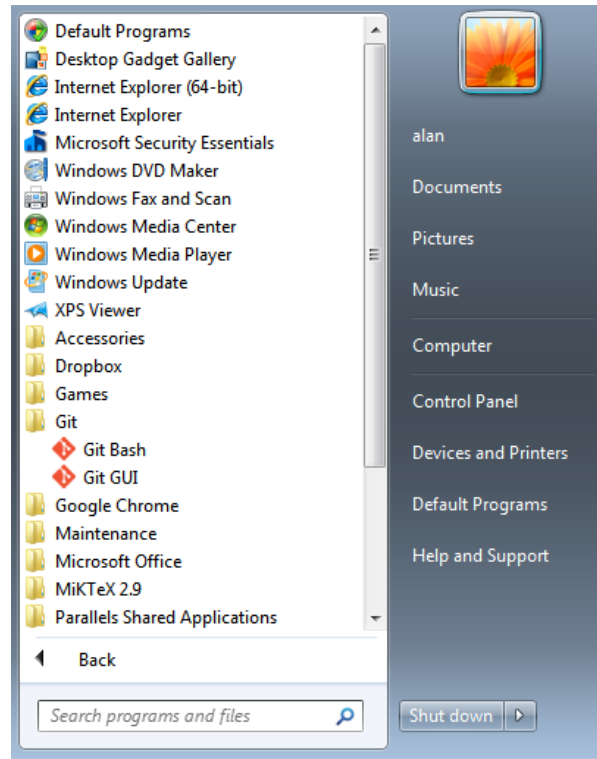


FIGURE 2.5: Windows Start Menu

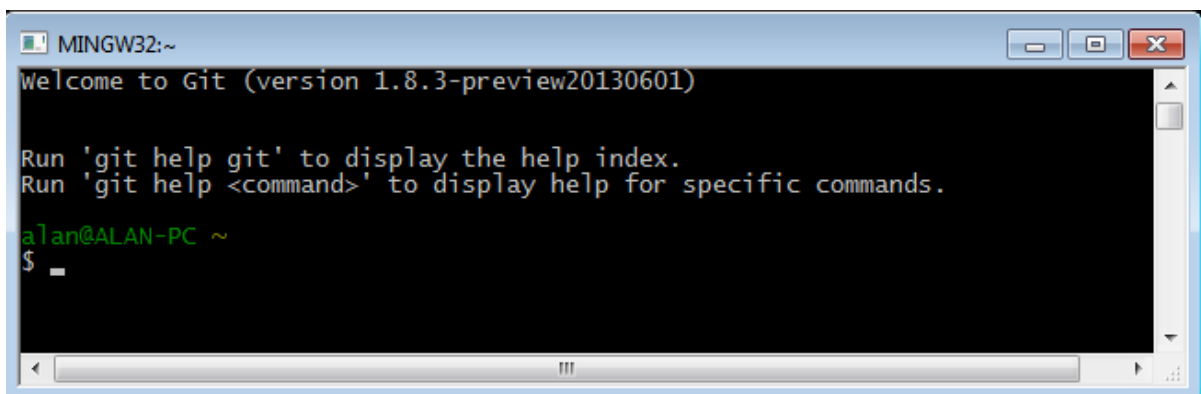


FIGURE 2.6: Git Bash Window

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

credential.helper=cache --timeout=10000000
user.email=arnholtat@appstate.edu
user.name=Alan T. Arnholt
core.repositoryformatversion=0
core.filemode=true
core.bare=false
core.logallrefupdates=true
remote.origin.url=https://github.com/alanarnholt/ThesisTemplate.git
remote.origin.fetch=+refs/heads/*:refs/remotes/origin/*
branch.main.remote=origin
branch.main.merge=refs/heads/main
```

From the `credential.helper` line in Git Example 2.2 on the previous 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>; click on 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 use the same name you used when you set up your `user.name` and `user.email` on your local machine.

¹ <http://en.wikipedia.org/wiki/GitHub>

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.

The screenshot shows the GitHub 'Create repository' interface. At the top, there's a navigation bar with the GitHub logo, a search bar, and links for 'Explore', 'Gist', 'Blog', and 'Help'. The user 'alanarnholt' is logged in. The form has two main sections: 'Owner' and 'Repository name'. The owner is 'alanarnholt' and the repository name is 'GiveltSomeNameHere' with a green checkmark. Below this, there's a hint: 'Great repository names are short and memorable. Need inspiration? How about **yolo-ironman**.' The 'Description (optional)' field contains 'GiveSomeDescriptionHere'. The visibility is set to 'Public' (radio button selected). Below this, there's a section for 'Initialize this repository with a README' which is checked. At the bottom, there are two buttons: 'Add .gitignore: None' and 'Add a license: None'. A large green 'Create repository' button is at the bottom center. The footer contains copyright information and links for 'Status', 'API', 'Training', 'Shop', 'Blog', and 'About'.

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 original 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 preceding page after entering the code in Git Example 2.4 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 repository. To initiate a pull request see Section 2.5 on page 23. A graphical representation of the two major collaboration modes is depicted in Figure 2.8.

Another approach is to use

```
git pull upstream master
```

What is the difference?

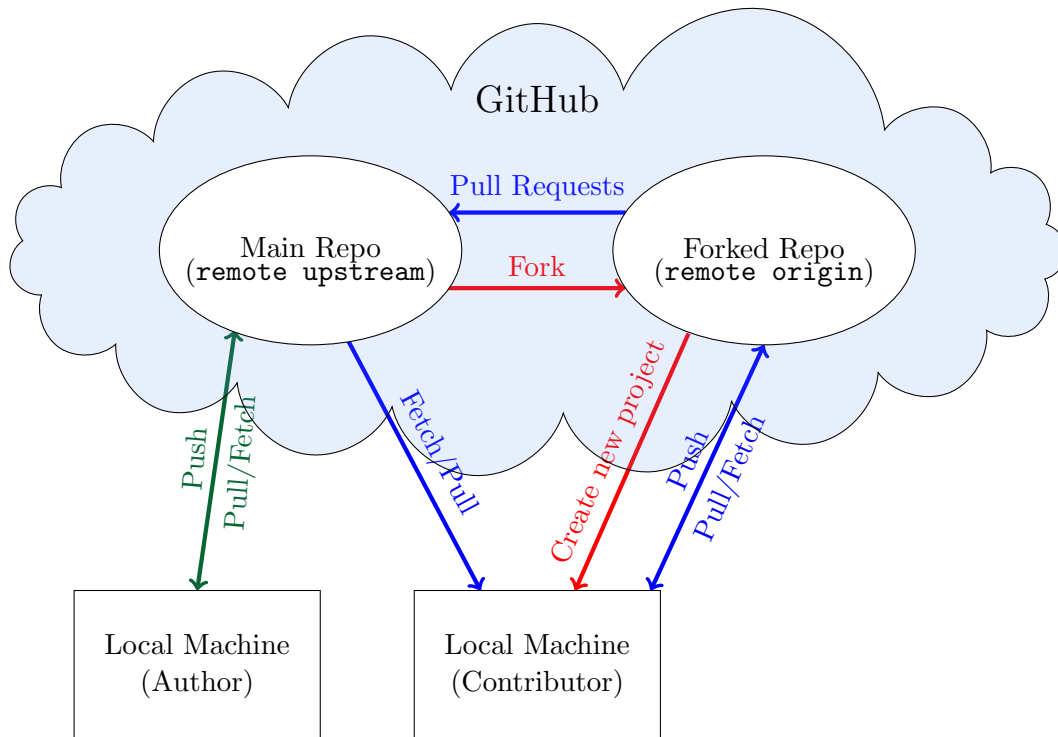


FIGURE 2.8: GitHub flow chart

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.) Click **Version Control**, and a new window such as 2.10 will appear where you will

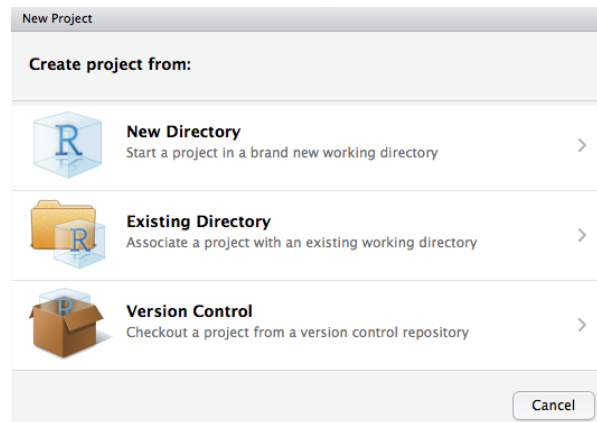


FIGURE 2.9: New Project window

select **Git**. In the next window that appears, see 2.11 on the next page, enter the URL for the

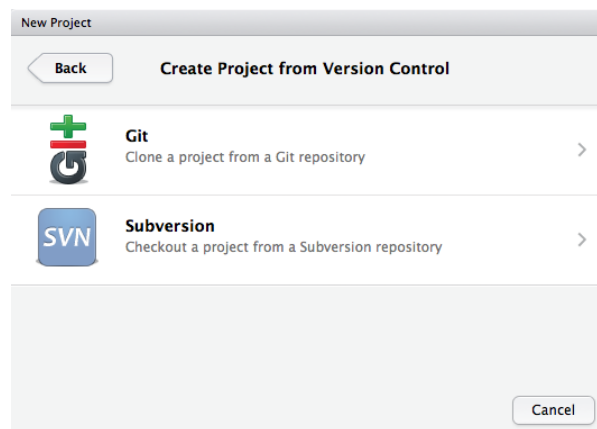


FIGURE 2.10: Create Project form Version Control window

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.

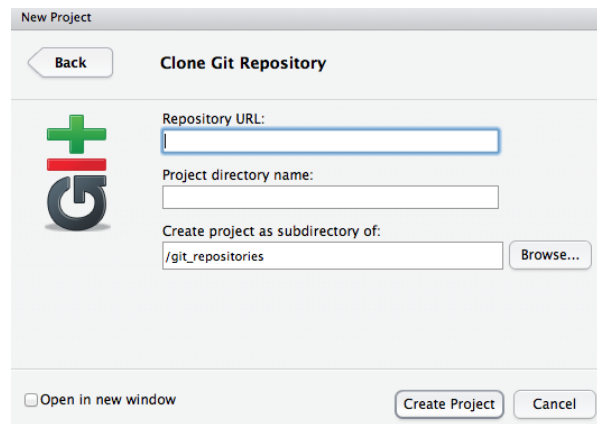


FIGURE 2.11: Clone Git Repository window

To check the current status of your repository type `git status` as shown in Git Example Git Example [2.6](#).

Git Example 2.6

```
git status
```

```
On branch main
```

```
Your branch is up to date with 'origin/main'.
```

```
Changes not staged for commit:
```

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

```
  (use "git restore <file>..." to discard changes in working directory)
```

```
modified:   ../../../../.gitignore
```

```
Untracked files:
```

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

```
Experimental/Xtables.pdf
```

```
figure/
```

```
../Figures/
```

```
../figure/
```

```
../main.pdf
```

```
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 staging 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"

[main ecb5013] staging all files
6 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 ThesisTemplate/thesis/Chapters/Experimental/Xtables.pdf
create mode 100644 ThesisTemplate/thesis/Chapters/figure/G1.pdf
create mode 100644 ThesisTemplate/thesis/Chapters/figure/G2-1.pdf
create mode 100644 ThesisTemplate/thesis/Chapters/figure/G2.pdf
create mode 100644 ThesisTemplate/thesis/Chapters/figure/Rgraph-1.pdf
create mode 100644 ThesisTemplate/thesis/Chapters/figure/ggplot2Graphs-1.pdf
```

Check the status after the last commit.

```
git status

On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

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

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

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

Push changes to the remote repository.

```
git push

To https://github.com/alanarnholt/ThesisTemplate.git
45761b1..ecb5013  main -> main
```

See if there is anything left to do.

```
git status

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

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

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

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

Show the last three commits with

```
git log -3

commit ecb50138f1845d7c4c943b7318560a862851c459
```



```
Author: Alan T. Arnholt <arnholtat@appstate.edu>
```

```
Date: Thu Aug 25 11:07:13 2022 -0400
```

```
    staging all files
```

```
commit 45761b174332970e51f2ad658b96c58dd95128be
```

```
Author: Alan T. Arnholt <arnholtat@appstate.edu>
```

```
Date: Thu Aug 25 11:02:19 2022 -0400
```

```
    tweaks
```

```
commit fec78c8c22dc7d0ebfba5e996b0b716785062594
```

```
Author: Alan T. Arnholt <arnholtat@appstate.edu>
```

```
Date: Thu Aug 25 10:53:52 2022 -0400
```

```
    tweaks
```

That was ugly. Let us try some formatting.

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

```
ecb50138f1845d7c4c943b7318560a862851c459 staging all files
```

```
45761b174332970e51f2ad658b96c58dd95128be tweaks
```

```
fec78c8c22dc7d0ebfba5e996b0b716785062594 tweaks
```

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

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

```
ecb5013 Thu Aug 25 11:07:13 2022 -0400- staging all files [Alan T. Arnholt]
```

```
45761b1 Thu Aug 25 11:02:19 2022 -0400- tweaks [Alan T. Arnholt]
```

```
fec78c8 Thu Aug 25 10:53:52 2022 -0400- tweaks [Alan T. Arnholt]
```

Maybe even some statistics?

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

```
ecb5013 Thu Aug 25 11:07:13 2022 -0400- staging all files [Alan T. Arnholt]
.../thesis/Chapters/Experimental/Xtables.pdf      | Bin 0 -> 135612 bytes
ThesisTemplate/thesis/Chapters/figure/G1.pdf      | Bin 0 -> 4788 bytes
ThesisTemplate/thesis/Chapters/figure/G2-1.pdf    | Bin 0 -> 4749 bytes
ThesisTemplate/thesis/Chapters/figure/G2.pdf      | Bin 0 -> 4696 bytes
ThesisTemplate/thesis/Chapters/figure/Rgraph-1.pdf | Bin 0 -> 5363 bytes
.../thesis/Chapters/figure/ggplot2Graphs-1.pdf   | Bin 0 -> 39621 bytes
6 files changed, 0 insertions(+), 0 deletions(-)
```

```
45761b1 Thu Aug 25 11:02:19 2022 -0400- tweaks [Alan T. Arnholt]
ThesisTemplate/thesis/main.Rnw | 1 +
1 file changed, 1 insertion(+)
```

```
fec78c8 Thu Aug 25 10:53:52 2022 -0400- tweaks [Alan T. Arnholt]
.gitignore | 20 +
ThesisTemplate/Git/GitOne.Rmd | 340 +++
ThesisTemplate/Git/GitOne.html | 565 +++++
ThesisTemplate/Git/GitOne.md | 452 ++++
.../ADD_37d613968d89bfc0fc573bd5f4d62239.RData | Bin 0 -> 355 bytes
...GITsetup_1e85974789856654fd6170253d9fcbe2.RData | Bin 0 -> 199 bytes
.../LOGP2_faf1a7fcf18137324c7c98647d730ea6.RData | Bin 0 -> 333 bytes
.../LOGP3_ad807c34590b81f628e6cb6032102919.RData | Bin 0 -> 974 bytes
.../LOGP_85b7cbcb21a72630e78bd3b750ea156f.RData | Bin 0 -> 329 bytes
.../LOG_087011b0f8f69f16e2c2e5605086c615.RData | Bin 0 -> 408 bytes
.../PUSH_4a4f909daa6525f7416b0e0df208f325.RData | Bin 0 -> 121 bytes
.../Rgraph_8ff4435448e1c897d42aaa6cf87ae77b.RData | Bin 0 -> 293 bytes
.../STATUS2_8ac7feaeff926fc5b62fe3cb6b50f859.RData | Bin 0 -> 580 bytes
.../STATUS3_12105d35cf1bc9dd32a9a4e2704fe895.RData | Bin 0 -> 729 bytes
.../STATUS_c084c907b84567e22073c1b7e27f7062.RData | Bin 0 -> 537 bytes
```

```

ThesisTemplate/Git/cache/__packages | 3 +
...t2Graphs_72d1a1d504100a2f2362cf955bc751c2.RData | Bin 0 -> 197 bytes
.../verify_2d584a4c99894e4c308ba44967920be7.RData | Bin 0 -> 498 bytes
ThesisTemplate/Git/figure/Rgraph.png | Bin 0 -> 17781 bytes
ThesisTemplate/Git/figure/ggplot2Graphs.png | Bin 0 -> 32780 bytes
ThesisTemplate/Git/images/CreateGitHub.png | Bin 0 -> 151971 bytes
ThesisTemplate/Git/images/CreateGitRepo.png | Bin 0 -> 91309 bytes
ThesisTemplate/Git/images/GitBashWindow.png | Bin 0 -> 18113 bytes
ThesisTemplate/Git/images/GitDownload.png | Bin 0 -> 450948 bytes
ThesisTemplate/Git/images/MacGitDownload.png | Bin 0 -> 42135 bytes
ThesisTemplate/Git/images/NewProject.png | Bin 0 -> 51988 bytes
.../Git/images/ProjectVersionControl.png | Bin 0 -> 44534 bytes
ThesisTemplate/Git/images/SGitDownload.png | Bin 0 -> 463619 bytes
ThesisTemplate/Git/images/Spotlight.png | Bin 0 -> 19473 bytes
ThesisTemplate/Git/images/TerminalWindow.png | Bin 0 -> 23152 bytes
ThesisTemplate/Git/images/VersionControl.png | Bin 0 -> 34971 bytes
ThesisTemplate/Git/images/WindowsGitBash.png | Bin 0 -> 53795 bytes
.../Git/images/WindowsProgramGitBash.png | Bin 0 -> 53795 bytes
ThesisTemplate/thesis/Appendices/AppendixA.Rnw | 9 +
.../thesis/Appendices/AppendixTemplate.Rnw | 9 +
.../thesis/Bibliography/Bibliography.bib | 86 +
ThesisTemplate/thesis/Bibliography/Rpkgs031914.bib | 303 +++
ThesisTemplate/thesis/Bibliography/Rpkgs122414.bib | 506 +++++
ThesisTemplate/thesis/Bibliography/RpksBIB.R | 13 +
ThesisTemplate/thesis/MissingPackages/booktabs.sty | 182 ++
ThesisTemplate/thesis/MissingPackages/caption.sty | 693 ++++++
ThesisTemplate/thesis/MissingPackages/fancyhdr.sty | 485 +++++
ThesisTemplate/thesis/MissingPackages/listings.sty | 2237 ++++++
ThesisTemplate/thesis/MissingPackages/natbib.sty | 724 ++++++
ThesisTemplate/thesis/MissingPackages/rotating.sty | 199 ++
ThesisTemplate/thesis/MissingPackages/setspace.sty | 546 +++++
.../thesis/MissingPackages/subfigure.sty | 218 ++

```

```
ThesisTemplate/thesis/MissingPackages/vmargin.sty | 576 +++++
ThesisTemplate/thesis/Primitives/Electron.doc    | Bin 0 -> 34304 bytes
ThesisTemplate/thesis/Thesis.cls                  | 1116 ++++++++
ThesisTemplate/thesis/cache/__packages            | 8 +
ThesisTemplate/thesis/jss.bst                     | 1631 ++++++++
ThesisTemplate/thesis/lstpatch.sty                | 1 +
ThesisTemplate/thesis/main.Rnw                    | 408 ++++
ThesisTemplate/thesis/main.bbl                    | 434 ++++
ThesisTemplate/thesis/vector.sty                  | 1 +
56 files changed, 11765 insertions(+)
```

2.5 So you want to collaborate?

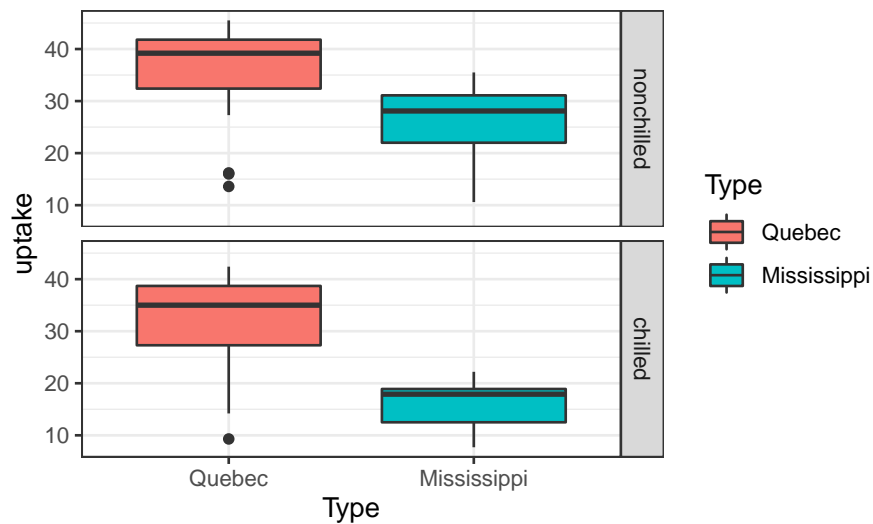
At this point, you have forked a repo and would like to contribute to someone's project. A great place to start is by reading <https://help.github.com/articles/using-pull-requests>.

Chapter 3

Using R

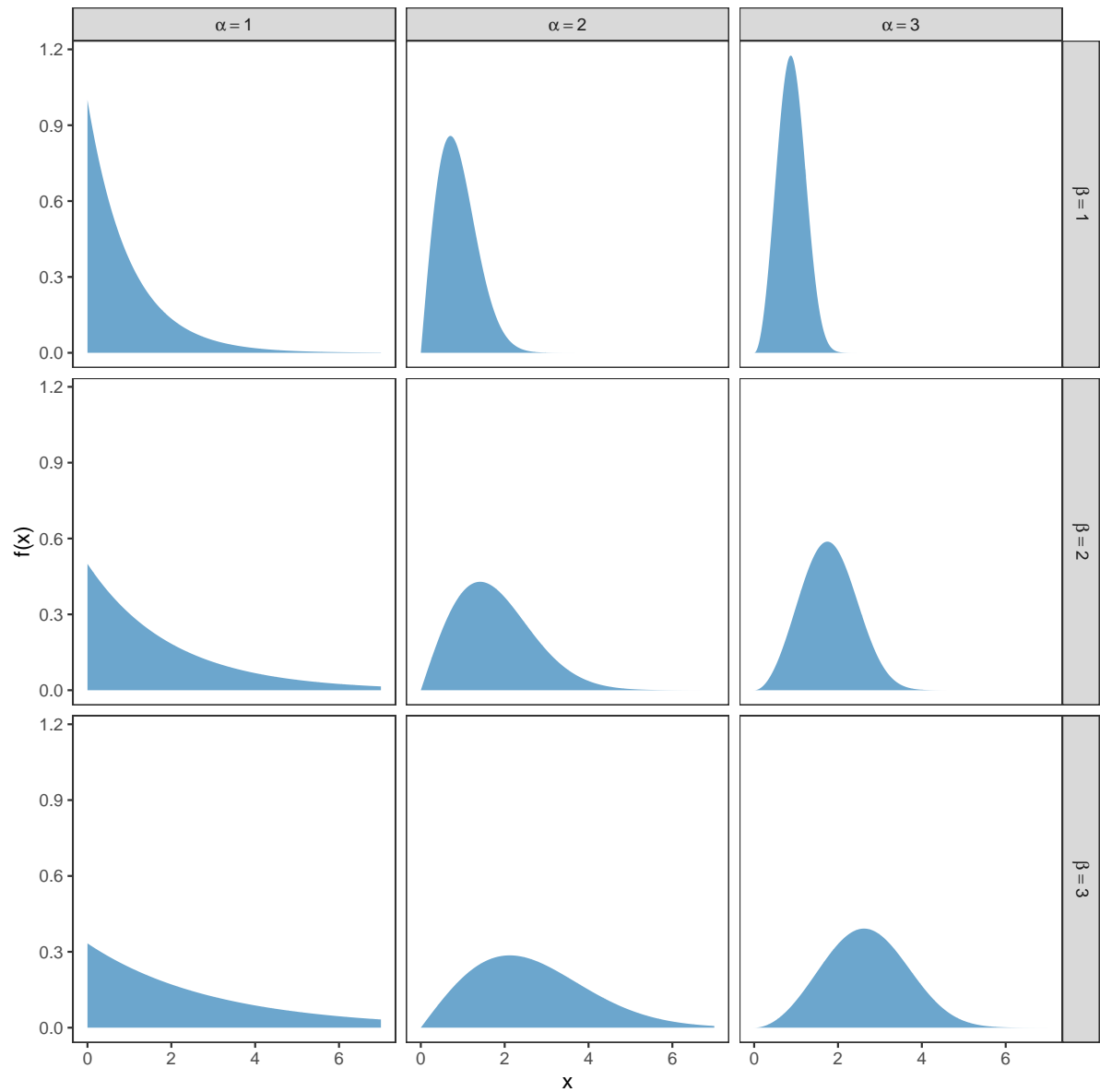
Now, just to show how cool this is, we will mix in a little R. First, consider the following graph where the R code that creates the graph is shown to the reader.

```
library(ggplot2)
ggplot(data = C02, aes(x = Type, y = uptake, fill = Type)) +
  geom_boxplot() + facet_grid(Treatment ~ .) + theme_bw()
```



I love graphs! The following graph created with `ggplot2` (Wickham, Chang, Henry, Pedersen, Takahashi, Wilke, Woo, Yutani, and Dunnington, 2022a) 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.



See R Code [3.1 on the next page](#) 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
```

Inline R expressions are created with `\Sexpr{}`. For example, the p -value from R Code 3.1 is 0.8343083.

Chapter 4

Using xtable

Chapter 5

Using BibT_EX

5.1 Bibliographies with BibT_EX

5.2 How to use citations

5.3 Generating a BibT_EX file of R packages

5.4 Using BibDesk

Appendix A

Appendix Title Here

Write your Appendix content here.

Bibliography

- Arnholt AT (2021). *PASWR2: Probability and Statistics with R, Second Edition*.
 <https://github.com/alanarnholt/PASWR2>.
- Arnold AS, Wilson JS, Boshier MG (1998). “A Simple Extended-Cavity Diode Laser.”
 Review of Scientific Instruments, **69**(3), 1236–1239. URL
 <http://link.aip.org/link/?RSI/69/1236/1>.
- Auguie B (2017). *gridExtra: Miscellaneous Functions for "Grid" Graphics*. R package version
 2.3, URL <https://CRAN.R-project.org/package=gridExtra>.
- Boettiger C (2021). *knitcitations: Citations for Knitr Markdown Files*. R package version
 1.0.12, URL <https://github.com/cboettig/knitcitations>.
- Canty A, Ripley B (2021). *boot: Bootstrap Functions (Originally by Angelo Canty for S)*. R
 package version 1.3-28, URL <https://CRAN.R-project.org/package=boot>.
- Chang W, Cheng J, Allaire J, Sievert C, Schloerke B, Xie Y, Allen J, McPherson J, Dipert A,
 Borges B (2022). *shiny: Web Application Framework for R*. R package version 1.7.2, URL
 <https://shiny.rstudio.com/>.
- Dahl DB, Scott D, Roosen C, Magnusson A, Swinton J (2019). *xtable: Export Tables to LaTeX
 or HTML*. R package version 1.8-4, URL <http://xtable.r-forge.r-project.org/>.
- Davison AC, Hinkley DV (1997). *Bootstrap Methods and Their Applications*. Cambridge
 University Press, Cambridge. ISBN 0-521-57391-2, URL
 <http://statwww.epfl.ch/davison/BMA/>.
- Dragulescu A, Arendt C (2020). *xlsx: Read, Write, Format Excel 2007 and Excel
 97/2000/XP/2003 Files*. R package version 0.6.5, URL
 <https://github.com/colearendt/xlsx>.

- Dragulescu AA (2014). *xlsxjars: Package required POI jars for the xlsx package*. R package version 0.6.1, URL <https://CRAN.R-project.org/package=xlsxjars>.
- Faraway J (2016). *faraway: Functions and Datasets for Books by Julian Faraway*. R package version 1.0.7, URL <http://people.bath.ac.uk/jjf23/>.
- Francois R (2020). *bibtex: Bibtex Parser*. R package version 0.4.2.3, URL <https://github.com/romainfrancois/bibtex>.
- Friendly M (2022). *vcdExtra: vcd Extensions and Additions*. R package version 0.8-0, URL <https://friendly.github.io/vcdExtra/>.
- Gandrud C (2013). *Reproducible Research with R and RStudio*. CRC Press.
- Gandrud C (2016). *repmis: Miscellaneous Tools for Reproducible Research*. R package version 0.5, URL <http://cran.r-project.org/package=repmis>.
- Genz A, Bretz F (2009). *Computation of Multivariate Normal and t Probabilities*. Lecture Notes in Statistics. Springer-Verlag, Heidelberg. ISBN 978-3-642-01688-2.
- Genz A, Bretz F, Miwa T, Mi X, Hothorn T (2021). *mvtnorm: Multivariate Normal and t Distributions*. R package version 1.1-3, URL <http://mvtnorm.R-forge.R-project.org>.
- Grätzer G, Schöpf R (2007). *More math into latex*, volume 4. Springer.
- Graves S, Piepho HP, with help from Sundar Dorai-Raj LS (2019). *multcompView: Visualizations of Paired Comparisons*. R package version 0.1-8, URL <https://CRAN.R-project.org/package=multcompView>.
- Gross J, Ligges U (2015). *nortest: Tests for Normality*. R package version 1.0-4, URL <https://CRAN.R-project.org/package=nortest>.
- Harrell Jr FE (2022). *Hmisc: Harrell Miscellaneous*. R package version 4.7-1, URL <https://hbiostat.org/R/Hmisc/>.
- Hawthorn CJ, Weber KP, Scholten RE (2001). “Littrow Configuration Tunable External Cavity Diode Laser with Fixed Direction Output Beam.” *Review of Scientific Instruments*, **72**(12), 4477–4479. URL <http://link.aip.org/link/?RSI/72/4477/1>.

- Hothorn T, Bretz F, Westfall P (2008a). “Simultaneous Inference in General Parametric Models.” *Biometrical Journal*, **50**(3), 346–363.
- Hothorn T, Bretz F, Westfall P (2022). *multcomp: Simultaneous Inference in General Parametric Models*. R package version 1.4-20, URL <https://CRAN.R-project.org/package=multcomp>.
- Hothorn T, Hornik K, van de Wiel MA, Zeileis A (2006). “A Lego system for conditional inference.” *The American Statistician*, **60**(3), 257–263. doi:10.1198/000313006X118430.
- Hothorn T, Hornik K, van de Wiel MA, Zeileis A (2008b). “Implementing a class of permutation tests: The coin package.” *Journal of Statistical Software*, **28**(8), 1–23. doi:10.18637/jss.v028.i08.
- Hothorn T, Winell H, Hornik K, van de Wiel MA, Zeileis A (2021). *coin: Conditional Inference Procedures in a Permutation Test Framework*. R package version 1.4-2, URL <http://coin.r-forge.r-project.org>.
- Lumley T (2020). *leaps: Regression Subset Selection*. R package version 3.1, URL <https://CRAN.R-project.org/package=leaps>.
- Meyer D, Dimitriadou E, Hornik K, Weingessel A, Leisch F (2022a). *e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien*. R package version 1.7-11, URL <https://CRAN.R-project.org/package=e1071>.
- Meyer D, Zeileis A, Hornik K (2006). “The Strucplot Framework: Visualizing Multi-Way Contingency Tables with vcd.” *Journal of Statistical Software*, **17**(3), 1–48. doi:10.18637/jss.v017.i03.
- Meyer D, Zeileis A, Hornik K (2022b). *vcd: Visualizing Categorical Data*. R package version 1.4-10, URL <https://CRAN.R-project.org/package=vcd>.
- Nijs V, Fang F, Trestle Technology, LLC, Allen J (2022). *shinyAce: Ace Editor Bindings for Shiny*. R package version 0.4.2, URL <https://CRAN.R-project.org/package=shinyAce>.
- Pinheiro J, Bates D, R Core Team (2022). *nlme: Linear and Nonlinear Mixed Effects Models*. R package version 3.1-159, URL <https://svn.r-project.org/R-packages/trunk/nlme/>.

- Pinheiro JC, Bates DM (2000). *Mixed-Effects Models in S and S-PLUS*. Springer, New York. doi:10.1007/b98882.
- R Core Team (2018). *foreign: Read Data Stored by Minitab, S, SAS, SPSS, Stata, Systat, Weka, dBase, ...* R package version 0.8-71, URL <https://CRAN.R-project.org/package=foreign>.
- R Core Team (2019). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.
- Rinker T (2014). *reports: Assist the Workflow of Writing Academic Articles and Other Reports*. R package version 0.1.4, URL <http://trinker.github.com/reports/>.
- Rinker TW (2013). *reports: Package to assist in report writing*. University at Buffalo/SUNY, Buffalo, New York. Version 0.1.4, URL <http://github.com/trinker/reports>.
- Ripley B (2022). *MASS: Support Functions and Datasets for Venables and Ripley's MASS*. R package version 7.3-58.1, URL <http://www.stats.ox.ac.uk/pub/MASS4/>.
- Sarkar D (2008). *Lattice: Multivariate Data Visualization with R*. Springer, New York. ISBN 978-0-387-75968-5, URL <http://lmdvr.r-forge.r-project.org>.
- Sarkar D (2021). *lattice: Trellis Graphics for R*. R package version 0.20-45, URL <http://lattice.r-forge.r-project.org/>.
- Sarkar D, Andrews F (2022). *latticeExtra: Extra Graphical Utilities Based on Lattice*. R package version 0.6-30, URL <http://latticeextra.r-forge.r-project.org/>.
- Sharpsteen C, Bracken C (2020). *tikzDevice: R Graphics Output in LaTeX Format*. R package version 0.12.3.1, URL <https://github.com/daqana/tikzDevice>.
- Spector P (2008). *Data manipulation with R*. Springer.
- Temple Lang D (2022). *RCurl: General Network (HTTP/FTP/...) Client Interface for R*. R package version 1.98-1.8, URL <https://CRAN.R-project.org/package=RCurl>.
- Venables WN, Ripley BD (2002). *Modern Applied Statistics with S*. Fourth edition. Springer, New York. ISBN 0-387-95457-0, URL <https://www.stats.ox.ac.uk/pub/MASS4/>.

- Wand M (2021). *KernSmooth: Functions for Kernel Smoothing Supporting Wand & Jones (1995)*. R package version 2.23-20, URL <https://CRAN.R-project.org/package=KernSmooth>.
- Wickham H (2007). “Reshaping Data with the reshape Package.” *Journal of Statistical Software*, **21**(12), 1–20. URL <http://www.jstatsoft.org/v21/i12/>.
- Wickham H (2011). “The Split-Apply-Combine Strategy for Data Analysis.” *Journal of Statistical Software*, **40**(1), 1–29. URL <https://www.jstatsoft.org/v40/i01/>.
- Wickham H (2016). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. ISBN 978-3-319-24277-4. URL <https://ggplot2.tidyverse.org>.
- Wickham H (2020). *reshape2: Flexibly Reshape Data: A Reboot of the Reshape Package*. R package version 1.4.4, URL <https://github.com/hadley/reshape>.
- Wickham H (2022a). *httr: Tools for Working with URLs and HTTP*. R package version 1.4.4, URL <https://CRAN.R-project.org/package=httr>.
- Wickham H (2022b). *plyr: Tools for Splitting, Applying and Combining Data*. R package version 1.8.7, URL <https://CRAN.R-project.org/package=plyr>.
- Wickham H, Chang W, Henry L, Pedersen TL, Takahashi K, Wilke C, Woo K, Yutani H, Dunnington D (2022a). *ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics*. R package version 3.3.6, URL <https://CRAN.R-project.org/package=ggplot2>.
- Wickham H, Danenberg P, Csárdi G, Eugster M (2022b). *roxygen2: In-Line Documentation for R*. R package version 7.2.1, URL <https://CRAN.R-project.org/package=roxygen2>.
- Wickham H, Hester J, Chang W, Bryan J (2022c). *devtools: Tools to Make Developing R Packages Easier*. R package version 2.4.4, URL <https://CRAN.R-project.org/package=devtools>.
- Wickham H, Pedersen TL (2019). *gtable: Arrange Grobs in Tables*. R package version 0.3.0, URL <https://github.com/r-lib/gtable>.
- Wieman CE, Hollberg L (1991). “Using Diode Lasers for Atomic Physics.” *Review of Scientific Instruments*, **62**(1), 1–20. URL <http://link.aip.org/link/?RSI/62/1/1>.

- Xie Y (2013). *Dynamic Documents with R and knitr*. CRC Press.
- Xie Y (2014). “knitr: A Comprehensive Tool for Reproducible Research in R.” In V Stodden, F Leisch, RD Peng (eds.), *Implementing Reproducible Computational Research*. Chapman and Hall/CRC. ISBN 978-1466561595, URL <http://www.crcpress.com/product/isbn/9781466561595>.
- Xie Y (2015). *Dynamic Documents with R and knitr*. 2nd edition. Chapman and Hall/CRC, Boca Raton, Florida. ISBN 978-1498716963, URL <https://yihui.org/knitr/>.
- Xie Y (2022a). *formatR: Format R Code Automatically*. R package version 1.12, URL <https://github.com/yihui/formatR>.
- Xie Y (2022b). *knitr: A General-Purpose Package for Dynamic Report Generation in R*. R package version 1.39, URL <https://yihui.org/knitr/>.
- Xie Y, Qiu Y (2021). *highr: Syntax Highlighting for R Source Code*. R package version 0.9, URL <https://github.com/yihui/highr>.
- Zeileis A, Meyer D, Hornik K (2007). “Residual-based Shadings for Visualizing (Conditional) Independence.” *Journal of Computational and Graphical Statistics*, **16**(3), 507–525. doi:10.1198/106186007X237856.