A primer in Human Cardiovascular Genetics

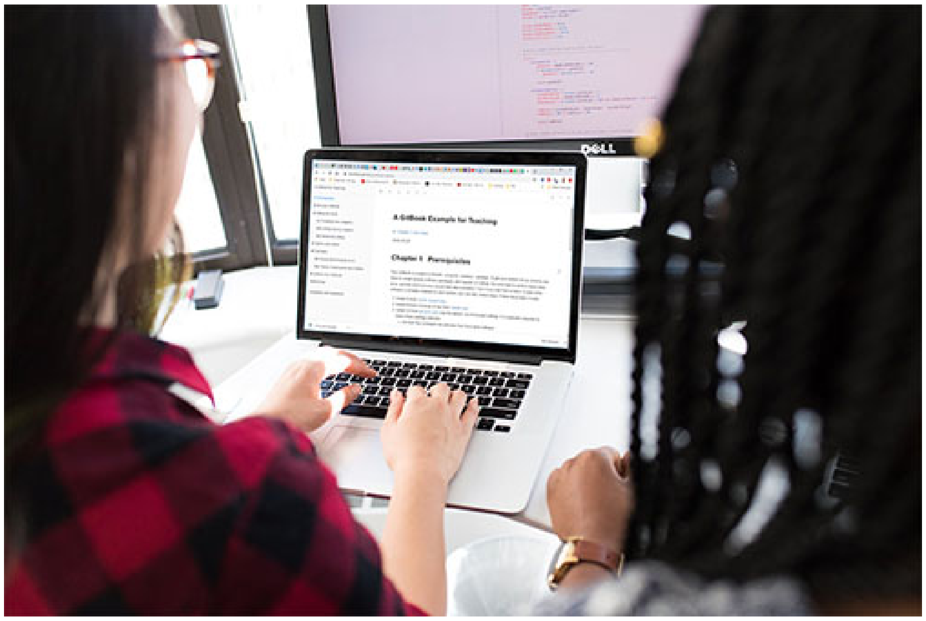
dr. Sander W. van der Laan

2022-03-24

Contents

Chapter 1

About this primer



Welcome to the course 'Genetic Epidemiology', and this practical entitled 'A primer in Human Cardiovascular Genetics'. In the next few days we will use this GitBook to perform quality control (QC), executing a genome-wide association study (GWAS), annotating the GWAS results, and performing further downstream analyses. We will use data from the first release of the *Welcome Trust Case-Control Consortium (WTCCC)* and focus on coronary artery disease (CAD).

We won't have time to perform imputation, but we will provide some pointers as to how to do this with minimal coding/scripting experience.

Part of this is based on four great Nature Protocols from the Zondervan group at the Wellcome Center Human Genetics.

1. Zondervan KT *et al. Designing candidate gene and genome-wide case-control association studies.* Nat Protoc 2007.
2. Pettersson FH *et al. Marker selection for genetic case-control association studies.* Nat Protoc 2009.
3. Anderson CA *et al. Data QC in genetic case-control association studies.* Nat Protoc 2010.
4. Clarke GM *et al. Basic statistical analysis in genetic case-control studies.* Nat Protoc 2011.

An update on the community standards of QC for GWAS can be found here:

1. Laurie CC *et al. Quality control and quality assurance in genotypic data for genome-wide association studies.* Genet Epidemiol 2010.

With respect to imputation you should also get familiar with the following two works:

1. Marchini, J. and Howie, B. *Genotype imputation for genome-wide association studies.* Nat Rev Genet 2010
2. de Bakker PIW *et al. Practical aspects of imputation-driven meta-analysis of genome-wide association studies.* Hum Mol Genet 2008.
3. Winkler TW *et al. Quality control and conduct of genome-wide association meta-analyses.* Nat Protoc 2014.

Importantly, follow this practical, but also review the questions asked in the presentation (on the screen or in the Dropbox) and try to answer them.

1.1 Team

We work with a team of enthousiastic lecturers with experience in bioinformatics, GWAS, genetic analyses, Mendelian randomisation, and epidemiology. This year the team consists of:

Charlotte Onland-Moret
*Associate Professor*N.C.Onland@umcutrecht.nl

Sander W. van der Laan
*Assistant professor*s.w.vanderlaan-2@umcutrecht.nl | swvanderlaan

Arjan Boltjes
*Postdoctoral researcher*a.boltjes@umcutrecht.nl | arjanboltjes

Joanna von Berg *PhD student*j.vonberg-24@umcutrecht.nl | JoannavonBerg
Jessica van Setten *Assistant professor*j.vansetten@umcutrecht.nl | j__vansetten

Getting started

Ready to start? Your first point of action is to prepare your system for this course (Chapter 2).

Chapter 2

Prerequisites

2.1 Linux, macOS, and Windows

Most programs made to execute genetic studies are developed for the Unix environment, *e.g.* Linux and macOS. Therefore, they may not work as intended on window's environment. Fortunately, Windows now allow users to install a linux subsystem within Windows 10 and you can find the detail guide here.
However, we highly recommend to 1) either install a linux subsystem on your Windows computer (*e.g.* a virtual machine with Ubuntu could work), or 2) switch to macOS in combination with homebrew as this will give you advantage of a powerful computer with a userfriendly interface and all the flexibility to use Unix-based programs for your work.

> For this practical we use a Windows laptop with Ubuntu on a VirtualMachine. Therefore every command is intended for Linux/macOS.

2.2 Programs you need

You need few programs for this practical or for your (future) genetic epidemiology work for that matter.

Program	Link	Description
PLINK	[https:/	/www.cog-genomics.org/plink2/](https://www.cog-genomics.org/plink2/){target="_blank"} PLINK is a free, open-source genetic analysis toolset, designed to perform a range of basic data parsing and quality control, as well as basic and large-scale analyses in a computationally efficient manner.
R	[https:/	/cran.r-project.org](https://cran.r-project.org/){target="_blank"} Progam to perform statistical analysis and visualisations.
RStudio	[https:/	/www.rstudio.com](https://www.rstudio.com){target="_blank"} A userfriendly R-wrap-around for code editing, debugging, analyses, and visualisation

Table 1: Programs needed for genetic epidemiology.

All genetic analyses can be done in PLINK, even on your laptop, but with large datasets, e.g. UK-biobank size, it's better to switch to a high-performance computing cluster like we have at the Uithof.

Nowadays, a lot of people also use programs like SNPTEST, BOLT-LMM, and GCTA as alternatives to run GWAS and downstream analyses, *e.g.* heritability estimation, Fst-calculation, etc.

Mendelian randomisation can be done either with the SMR or GSMR function from GCTA or with R-packages, like `TwoSampleMR`.

2.3 The Terminal

For all the above programs, except RStudio, you will need the `Terminal`. This comes with every major operating system. And you will (start to) make your own scripts. The benefit of using scripts is that each step is clearly layed out and easily replicable, thus allows for greater reproducibility, easier troubleshooting and can be run on high-performance computer clusters.

Download the data you need to your Desktop: LINK.

Now open the terminal, it should be on the left in the toolbar as a little black computer-monitor-like icon. Mac uses can type `command + space` and type `terminal`, a terminal screen should open.

> From now on we will use little code blocks like the example to indicate a code you should type/copy-paste and hit enter. If a code is followed by a comment, it is indicated by a # - you don't need to copy-paste and execute this.

```
CODE BLOCK

CODE BLOCK # some comment here
```

You can navigate around the computer through the terminal by typing `cd <path>`; `cd` stands for "change directory" and means "some_file_directory_you_want_to_go_to".

```
# For Linux/macOS Users
cd ~ # will bring you to your home directory
cd ../ # will bring you to the parent directory (up one level)
cd XXX # will bring you to the XXX directory
```

Let's navigate to the folder you just downloaded.

```
cd ~/Desktop/practical
```

Let's check out what is inside the directory, by listing (`ls`) its contents.

```
ls -lh

# For Linux/macOS Users
ls -l # shows files as list
ls -lh # shows files as list with human readable format
ls -lt # shows the files as list sorted by time edited
ls -lS # shows the files as list sorted by size
```

Adding the flags `-lh` will get you the contents of a directory in a list (`-l`) and make the size 'human-readable' (`-h`).

You can also count the number of files.

```
ls | wc -l
```

2.4 Installing some R packages

I tested this VirtualMachine and everything should be fine, except some libraries weren't there. We need to install them.

To be able to install certain `r`-packages, we need to install some Linux (Ubuntu) software. Type the following:

```
sudo apt-get install libcurl4 libcurl4-openssl-dev -y

sudo apt-get install libssl-dev
```

Now close the terminal window - really making sure that the terminal-program has quit.

Open a new terminal window and open `r` by simply typing `R`. You should install the following packages, and then you're good to go!

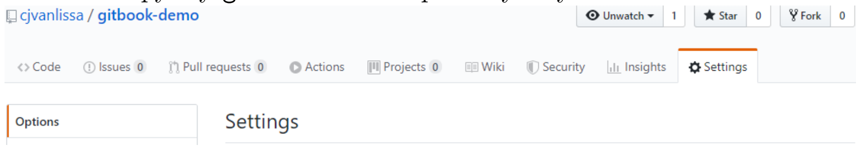
```
install.packages(c("httr", "usethis", "data.table", "devtools", "qqman", "CMplot", "ti

library(c("httr", "usethis", "data.table", "devtools", "qqman", "CMplot", "tibble", "p
devtools::install_github("kassambara/ggpubr")
library("ggpubr")
```

All in all this may take some time, good moment to relax, review your notes, stretch your legs, or take a coffee.

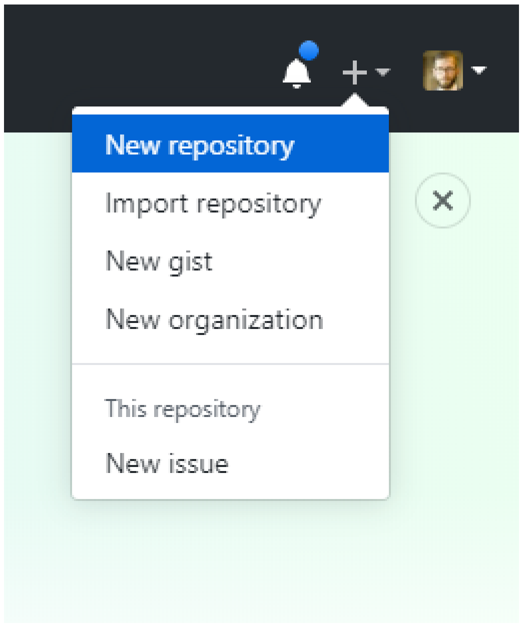
Chapter 3

Get your GitBook

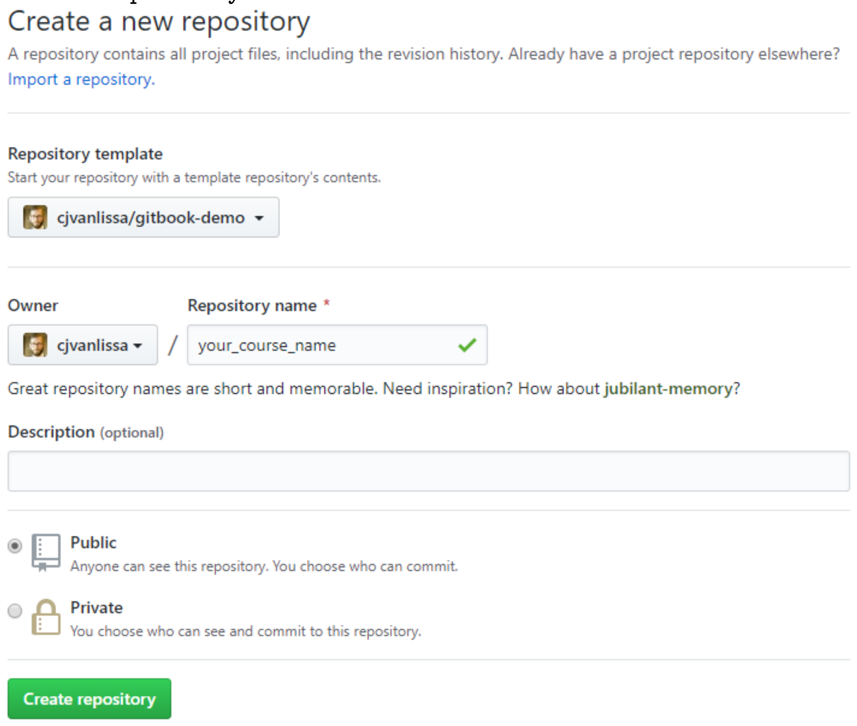
To get your GitBook, you should follow these steps:

1. Go to https://github.com/cjvanlissa/gitbook-demo
2. In the top right of the page, click `Fork`.
   This will copy my `gitbook-demo` repository to your GitHub account.
   

3. My repository is now copied to your account. It is a template repository, which means that you can create a *new repository* based on this one.
4. Create a new repository for your own GitBook. Create one for a course you've been wanting to update. In the top-right corner of the GitHub website, click the + icon, and select "New repository":



5. In the dialog, select the `gitbook-demo` as "Repository template", and give the repository an appropriate name for your course. Then, press `Create repository`:



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Repository template

Start your repository with a template repository's contents.

cjvanlissa/gitbook-demo

Owner: cjvanlissa / Repository name *: your_course_name

Great repository names are short and memorable. Need inspiration? How about jubilant-memory?

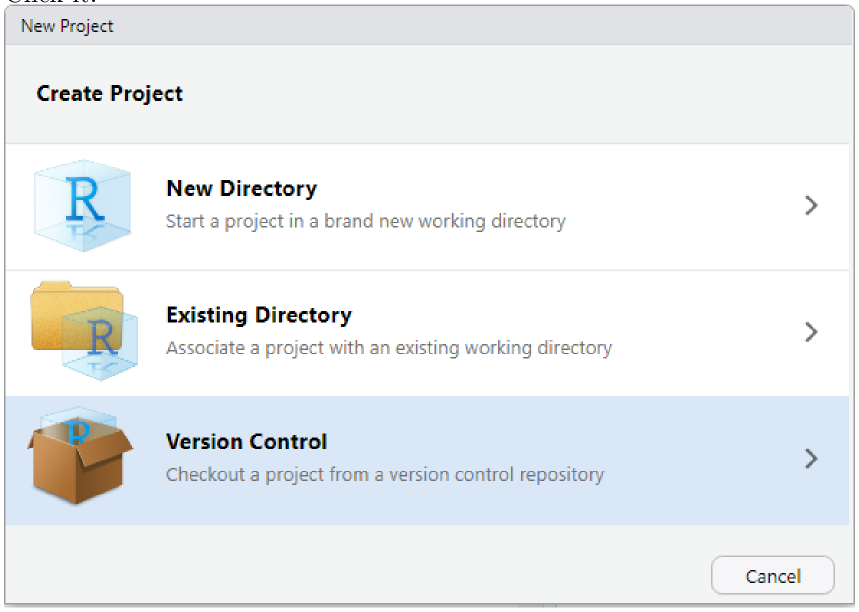
Description (optional)

[✓ Public] Anyone can see this repository. You choose who can commit.

Private You choose who can see and commit to this repository.

Create repository

6. Now, go back to Rstudio on your computer. In Rstudio, click `File > New Project`. A dialog will open. If you set up Rstudio with Git correctly, the dialog should have an option to create a new project from Version control. Click it:



New Project

Create Project

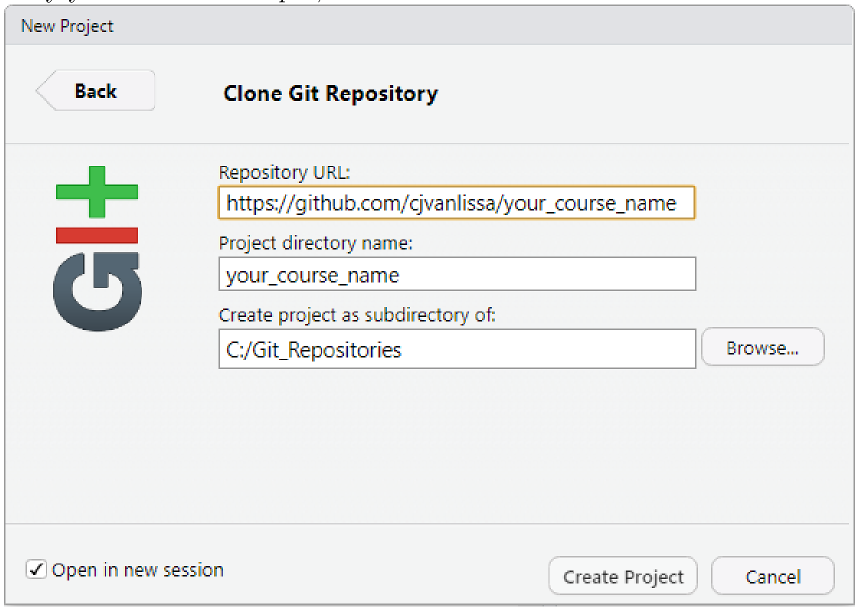
New Directory
Start a project in a brand new working directory

Existing Directory
Associate a project with an existing working directory

Version Control
Checkout a project from a version control repository

Cancel

7. In the next dialog window, you should copy the URL of the GitHub repository you created in *Step 5*, like so:



New Project

Back **Clone Git Repository**

Repository URL:
https://github.com/cjvanlissa/your_course_name

Project directory name:
your_course_name

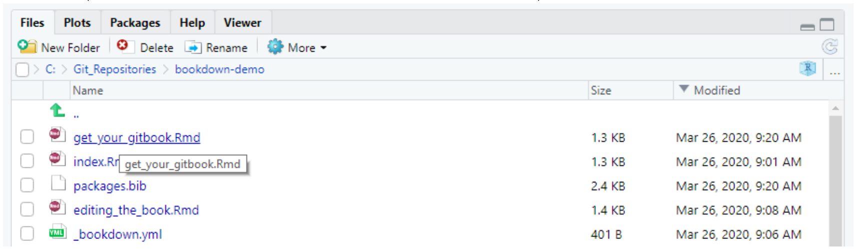
Create project as subdirectory of:
C:/Git_Repositories Browse...

✓ Open in new session

Create Project Cancel

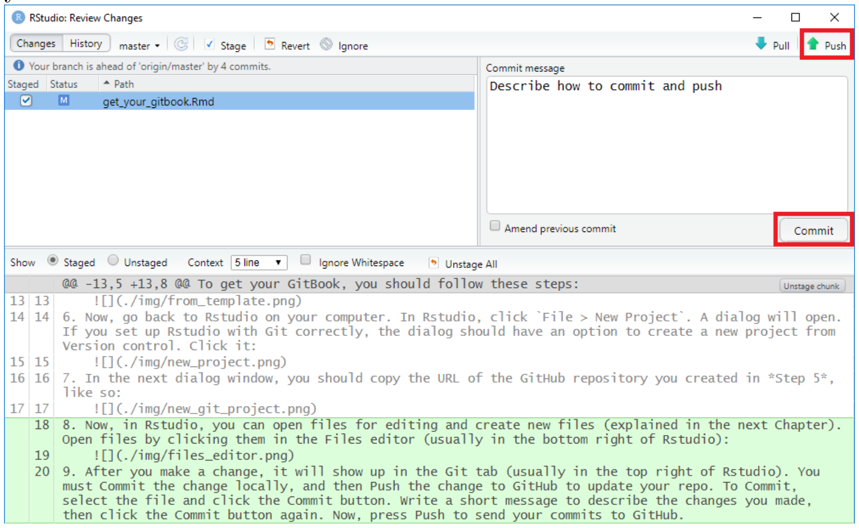
8. Now, in Rstudio, you can open files for editing and create new files (ex-

plained in the next Chapter). Open files by clicking them in the Files editor (usually in the bottom right of Rstudio):



Name	Size	Modified
..		
get_your_gitbook.Rmd	1.3 KB	Mar 26, 2020, 9:20 AM
index.Rmd	1.3 KB	Mar 26, 2020, 9:01 AM
packages.bib	2.4 KB	Mar 26, 2020, 9:20 AM
editing_the_book.Rmd	1.4 KB	Mar 26, 2020, 9:08 AM
_bookdown.yml	401 B	Mar 26, 2020, 9:06 AM

9. After you make a change, it will show up in the Git tab (usually in the top right of Rstudio). You must Commit the change locally, and then Push the change to GitHub to update your repo. To Commit, select the file and click the Commit button. Write a short message to describe the changes you made, then click the Commit button again. Now, press Push to send your commits to GitHub.



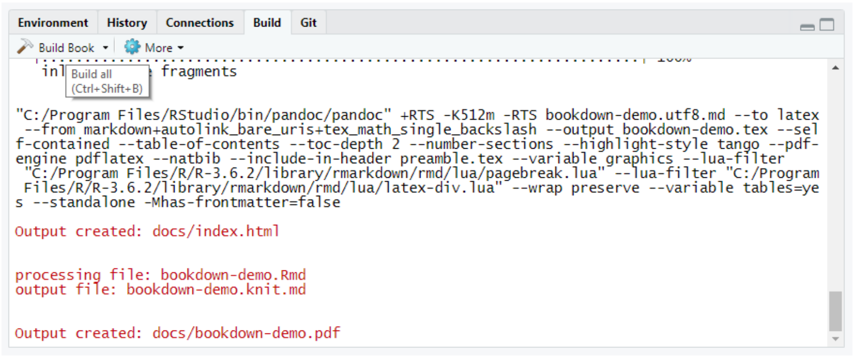
Your branch is ahead of 'origin/master' by 4 commits.

Staged	Status	Path
✓	M	get_your_gitbook.Rmd

Commit message: Describe how to commit and push

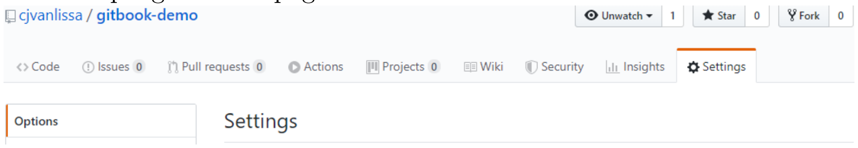
```
@@ -13,5 +13,8 @@ To get your GitBook, you should follow these steps:
    ![](./img/from_template.png)
6. Now, go back to Rstudio on your computer. In Rstudio, click `File > New Project`. A dialog will open.
If you set up Rstudio with Git correctly, the dialog should have an option to create a new project from
Version control. Click it:
    ![](./img/new_project.png)
7. In the next dialog window, you should copy the URL of the GitHub repository you created in *Step 5*,
like so:
    ![](./img/new_git_project.png)
8. Now, in Rstudio, you can open files for editing and create new files (explained in the next Chapter).
Open files by clicking them in the Files editor (usually in the bottom right of Rstudio):
    ![](./img/files_editor.png)
9. After you make a change, it will show up in the Git tab (usually in the top right of Rstudio). You
must Commit the change locally, and then Push the change to GitHub to update your repo. To Commit,
select the file and click the Commit button. Write a short message to describe the changes you made,
then click the Commit button again. Now, press Push to send your commits to GitHub.
```

10. To render your book as a GitBook, you must “Build” it. In the top-right panel of Rstudio, you see a “Build” tab. In this tab, simply click the “Build Book” button to build your book. You should see a lot of rendering messages, until a window pops up with your brand new GitBook. If you get errors at this stage, you probably made a mistake in preparing your system (see the previous Chapter).

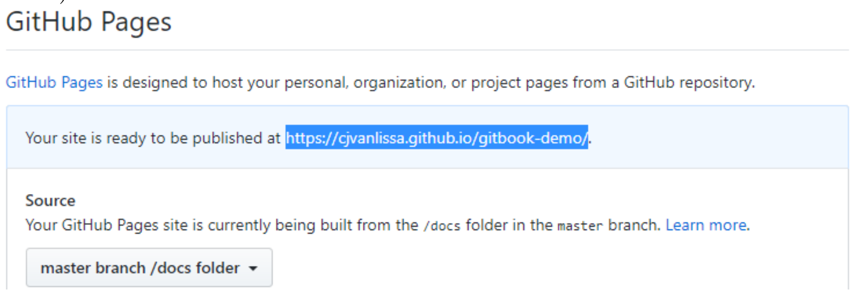


```
   inl        fragments

"C:/Program Files/RStudio/bin/pandoc/pandoc" +RTS -K512m -RTS bookdown-demo.utf8.md --to latex
 --from markdown+autolink_bare_uris+tex_math_single_backslash --output bookdown-demo.tex --sel
f-contained --table-of-contents --toc-depth 2 --number-sections --highlight-style tango --pdf-
engine pdflatex --natbib --include-in-header preamble.tex --variable graphics --lua-filter
 "C:/Program Files/R/R-3.6.2/library/rmarkdown/rmd/lua/pagebreak.lua" --lua-filter "C:/Program
 Files/R/R-3.6.2/library/rmarkdown/rmd/lua/latex-div.lua" --wrap preserve --variable tables=ye
s --standalone -Mhas-frontmatter=false

Output created: docs/index.html


processing file: bookdown-demo.Rmd
output file: bookdown-demo.knit.md

Output created: docs/bookdown-demo.pdf
```

11. Building the book generated a lot of new files in the `./docs` directory. This directory contains the website files for your GitBook. Open the Git tab again, verify that the `./docs` directory is listed, and Commit and Push all of these new files as described in *Step 9*.
12. There is only one last remaining task: To publish your GitBook on GitHub pages. Once you do this, any change to the `./docs` folder that you push to GitHub will lead to an immediate update of your GitBook website. Go back to the GitHub page for your Repository. Click on the `Settings` tab on the top right of the page:



13. On the Settings page, scroll all the way down until you reach a section called `GitHub Pages`. There, under the "Source" heading, click the word `None`, and select `master branch /docs folder`. When you select it, the page will update, and if you scroll back down to the `GitHub Pages` section, you will see the URL where your GitBook is published. The first time, it will take a few minutes for your GitBook to come online. When you publish updates to the GitBook however (simply by following *Step 11* again), the update will be near-instantaneous. The Pages section should now look like this (and that is hopefully the link where you found this book):


GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

Your site is ready to be published at https://cjvanlissa.github.io/gitbook-demo/.

Source

Your GitHub Pages site is currently being built from the `/docs` folder in the `master` branch. Learn more.

master branch /docs folder

Chapter 4

Editing the book

The contents of the book are written in **RMarkdown**. You can use any formatting code that Pandoc's Markdown supports, e.g., a math equation $a^2 + b^2 = c^2$. Moreover, you can include chunks of R-code, like this:

The results of these chunks can be rendered to the GitBook:

```
## [1] "This is an R-command!"
```

To edit the book, you can change the text in the `.Rmd` files. Each Rmd file should contain **one and only one** chapter. A chapter is defined by the first-level heading `#`, e.g., `# Editing the book`.

Any sub-headings within the chapter are indicated with several `#` signs, e.g., `##` (level 2) and `###` (level 3).

4.1 Creating new chapters

To create a new chapter, you must follow two steps: 1) Create the file, and 2) Include it in the list of chapters.

First, to create the file for a new chapter in Rstudio, click `File > New File > Text file`. At the top of the file, write your chapter heading, as explained above. Then, click `File > Save`. Save the file as `.Rmd`, without spaces in the file name, e.g.: `editing_the_book.Rmd`.

Second, to include it in the list of chapters, open the file `_bookdown.yml` (click it in the Files explorer in the bottom right of Rstudio). This file has a list of `.Rmd` files to be included in the book. In this example, the list looks like this:

```
tmp <- readLines("_bookdown.yml")
cat(tmp[grep("^rmd_files", tmp):grep("references\\.Rmd", tmp)], sep = "\n")
```

rmd_files: [“index.Rmd”, “prerequisites.Rmd”, “get_your_gitbook.Rmd”, “editing_the_book.Rmd”, “figures_tables.Rmd”, “examples.Rmd”, “open_educational.Rmd”, “use_in_course.Rmd”, “licenses.Rmd”, “references.Rmd”]

Insert the file name of your new chapter in the desired position in this list.

4.2 Linking across chapters

You can label chapter and section titles using `{#label}` after them. The labels can be used as cross-references. For example, we can link to Chapter 5. If you do not manually label chapters, there will be automatic labels anyway, e.g., Chapter 6.

4.3 Advanced editing

The convenient Rmarkdown Cheat Sheet by Rstudio covers most of the knowledge required for advanced Rmarkdown editing. You can print it out and stick it to your wall!

Chapter 5

Figures and tables

Figures and tables with captions will be placed in `figure` and `table` environments, respectively.

```r
par(mar = c(4, 4, .1, .1))
plot(pressure, type = 'b', pch = 19)
```

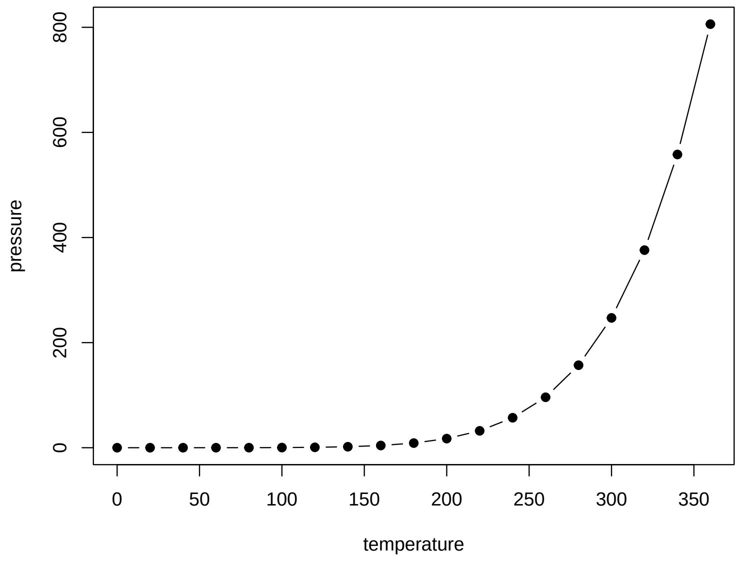


Figure 5.1: Here is a nice figure!

Reference a figure by its code chunk label with the `fig:` prefix, e.g., see Figure 5.1. Similarly, you can reference tables generated from `knitr::kable()`, e.g., see Table 5.1.

Table 5.1: Here is a nice table!

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.0	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa
4.6	3.4	1.4	0.3	setosa
5.0	3.4	1.5	0.2	setosa
4.4	2.9	1.4	0.2	setosa
4.9	3.1	1.5	0.1	setosa
5.4	3.7	1.5	0.2	setosa
4.8	3.4	1.6	0.2	setosa
4.8	3.0	1.4	0.1	setosa
4.3	3.0	1.1	0.1	setosa
5.8	4.0	1.2	0.2	setosa
5.7	4.4	1.5	0.4	setosa
5.4	3.9	1.3	0.4	setosa
5.1	3.5	1.4	0.3	setosa
5.7	3.8	1.7	0.3	setosa
5.1	3.8	1.5	0.3	setosa

```r
knitr::kable(
  head(iris, 20), caption = 'Here is a nice table!',
  booktabs = TRUE
)
```

You can write citations, too. For example, we are using the **bookdown** package (Xie, 2022) in this sample book, which was built on top of R Markdown and **knitr** (Xie, 2015).

Chapter 6

Examples

Here are some examples of other GitBooks for courses; if you want to have your GitBook added to the list, please send a Pull Request (here's how to send a pull request).

6.1 Doing Meta-Analysis in R

http://cjvanlissa.github.io/Doing-Meta-Analysis-in-R

A GitBook on doing meta-analysis in R, based on the book 'Doing Meta-Analysis in R', by Mathias Harrer, Pim Cuijpers, & David Ebert, and adapted to focus on the metafor package, and exploring heterogeneity using metaforest. The original can be found here: https://bookdown.org/MathiasHarrer/Doing_Meta_Analysis_in_R/

6.2 Theory Construction and Statistical Modeling

http://cjvanlissa.github.io/TCSM

A GitBook for the course *"Theory Construction and Statistical Modeling"*, with some interesting code, for example: Blocks of answers to the tutorial questions that can be collapsed and expanded.

Chapter 7

Open Educational Resources

UNESCO defines Open Educational Resources as *teaching, learning and research materials in any medium – digital or otherwise – that reside in the public domain or have been released under an open license that permits no-cost access, use, adaptation and redistribution by others with no or limited restrictions.*

Open Educational resources can help lighten the workload on individual teachers, who can collaborate with the development of high-quality open access resources, instead of having to develop their own proprietary materials from scratch. Moreover, Open Educational resources are inclusive, lowering the barrier to knowledge acquisition for learners around the world, and enabling lifelong learning for those outside academia.

Many universities support Open Educational Resources. Here are just a few (feel free to send a pull request with other relevant resources).

- **OER Commons**: A freely accessible online library of open educational resources.
- **Utrecht University Figshare**: Open learning objects from Utrecht University.
- **Johns Hopkins University OCW**: Open public health courses and materials.
- **University of Pittsburgh OER**: Big List of Open Educational Resources.
- **MERLOT**: Online learning and support materials and content creation tools, led by an international community of educators, learners and researchers.

Chapter 8

Compatibility with existing systems

Many universities offer digital platforms for learning. You might wish to embed your GitBook within these existing systems. Here are two ways in which you might do that. Currently, this section only discusses BlackBoard, but the same principles should apply to other platforms.

8.1 Add a hyperlink

You can add a link to your GitBook in the BlackBoard course menu by following this tutorial.

8.2 Embed the whole book

You can add a Blank Page to your BlackBoard course menu, and fill that page with a full-size "iframe" - a web page within the web page. This tutorial explains how to do it. It is possible that your university is blocking this feature, however.

Chapter 9

License your GitBook

In the spirit of Open Science, it is good to think about making your course materials Open Source. That means that other people can use them. In principle, if you publish materials online without license information, you hold the copyright to those materials. If you want them to be Open Source, you must include a license. It is not always obvious what license to choose.

The Creative Commons licenses are typically suitable for course materials. This GitBook, for example, is licensed under CC-BY 4.0. That means you can use and remix it as you like, but you must credit the original source.

If your project is more focused on software or source code, consider using the GNU GPL v3 license instead.

You can find more information about the Creative Commons Licenses here. Specific licenses that might be useful are:

- CC0 ("No Rights Reserved"), everybody can do what they want with your work.
- CC-BY 4.0 ("Attribution"), everybody can do what they want with your work, but they must credit you. Note that this license may not be suitable for software or source code!

For compatibility between CC and GNU licenses, see this FAQ.

Bibliography

Xie, Y. (2015). *Dynamic Documents with R and knitr*. Chapman and Hall/CRC, Boca Raton, Florida, 2nd edition. ISBN 978-1498716963.

Xie, Y. (2022). *bookdown: Authoring Books and Technical Documents with R Markdown*. R package version 0.25.