Basic Statistics and Projects in R

Reproducibility and GitHub

Christian Althaus, Alan Haynes

UNIVERSITY OF BERN

Public Health Sciences Course Program - Basic Statistics and Projects in R. Slides availab on **GitHub**.

Summary of yesterday morning

- Introduction to R and RStudio
- Objects, functions, and packages
- R projects and project management (folder structure and file names)
- Base R vs. tidyverse
- Data wrangling with the tidyverse (e.g., `dplyr`)
  - Pipes (`%>%`, `|>`)
  - Data import
  - Modifying data
  - Summarizing data
- Tables

Public Health Sciences Course Program - Basic Statistics and Projects in R. Slides availab on **GitHub**.

Summary of yesterday afternoon

- Fundamentals of data visualization
- Color scales
- Data visualization with the tidyverse (`gpplot2`)
  - `data` are mapped to `aes`thetic properties of `geom`etric objects
  - Themes
  - Combine and arrange plots (`patchwork`, `cowplot`)

Don’t forget about the online resources!

Today

Time	Duration	Topic	Content
09:00-10:30	90 min	Reproducible documents	Markdown, Quarto
10:30-11:00	30 min	*Coffee break*	Coffee, sun, fresh air
11:00-12:00	60 min	Version control and collaboration	Git/GitHub
12:00-12:30	30 min	Websites	GitHub Pages

Public Health Sciences Course Program - Basic Statistics and Projects in R. Slides availab on **GitHub**.

Reproducible documents

UNIVERSITY OF BERN

Public Health Sciences Course Program - Basic Statistics and Projects in R. Slides availab on **GitHub**.

UNIVERSITY OF BERN

Open science

> *Open Science is the conduct of science in such a way that others can collaborate and contribute, where research data, laboratory notes and other research processes are freely available, with licence terms that allow re-use, redistribution and reproduction of the research.* (**FOSTER**)

Also see the **Open Science resources** at the University of Bern.
Wanna implement these practices in your research group? Read **Ten simple rules for implementing open and reproducible research practices after attending a training course**.

Public Health Sciences Course Program - Basic Statistics and Projects in R. Slides availab on **GitHub**.

b
UNIVERSITY
OF BERN

Public Health Sciences Course Program - Basic Statistics and Projects in R. Slides availab on **GitHub**.

What is reproducibility?

UNIVERSITY OF BERN

Reproducibility:

a different analyst re-performs the analysis with
the **same code** and
the **same data** and obtains
the **same result**.

Patil, Peng, Leek (2016) https://www.biorxiv.org/content/10.1101/066803v1

Public Health Sciences Course Program - Basic Statistics and Projects in R. Slides availab on **GitHub**.

UNIVERSITY OF BERN

Repeatable: keeping everything the same but repeating the analysis - do we get the same results?

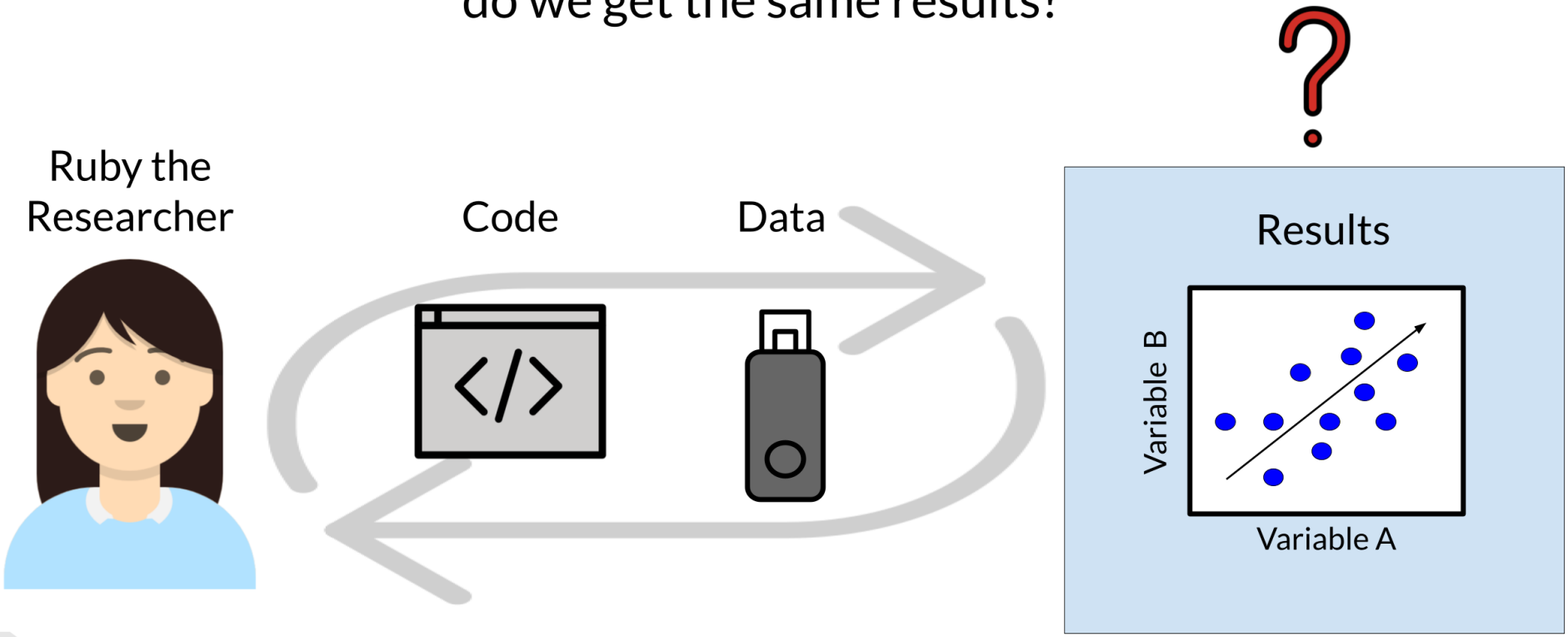


Image created by Candace Savonen using Avataars.

UNIVERSITY OF BERN

Reproducible: using the same data and analysis but in the hands of *another researcher* - do we get the same results?

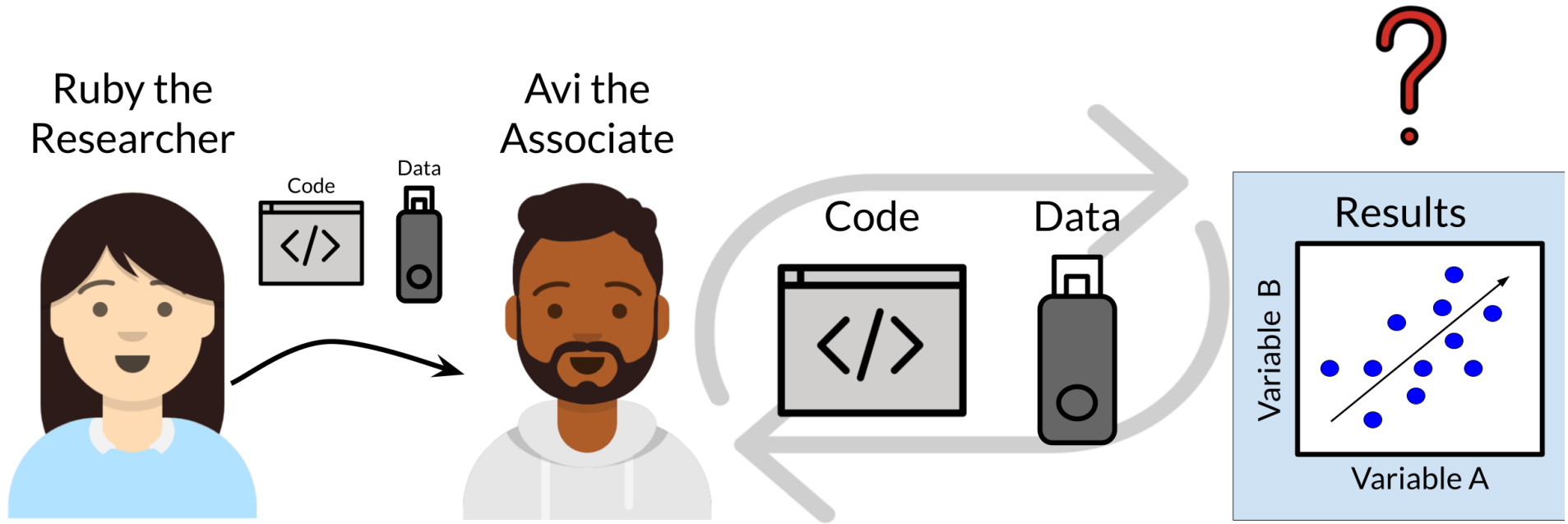
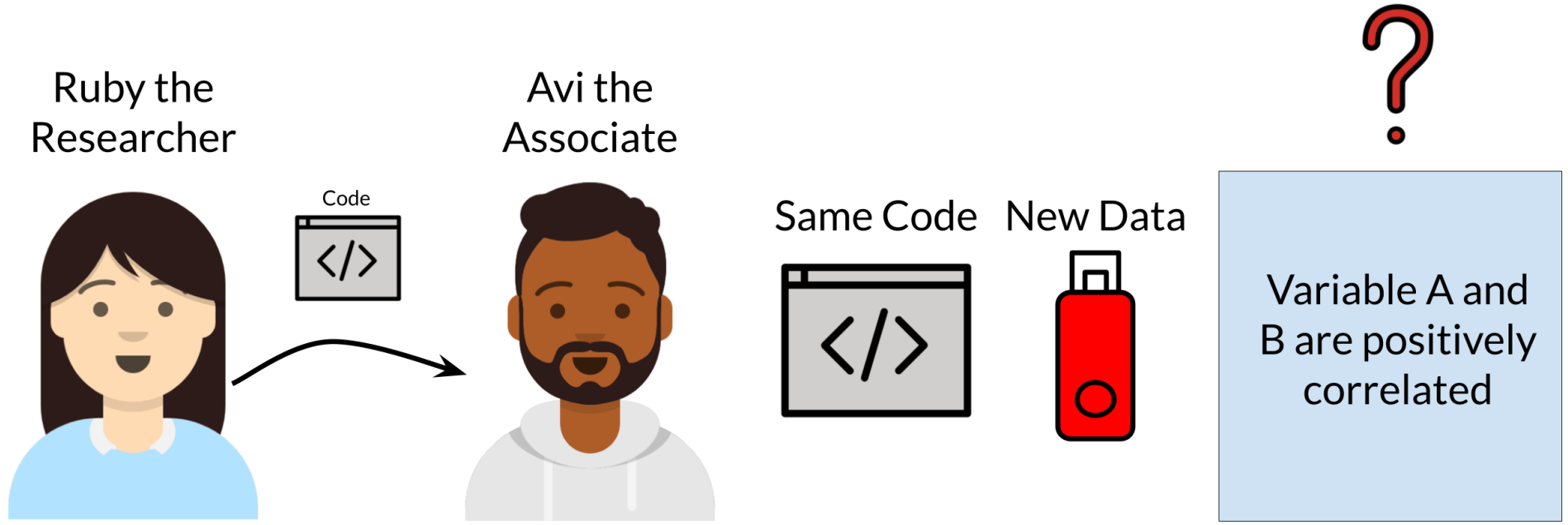


Image created by Candace Savonen using Avataars.

Public Health Sciences Course Program - Basic Statistics and Projects in R. Slides availab on **GitHub**.

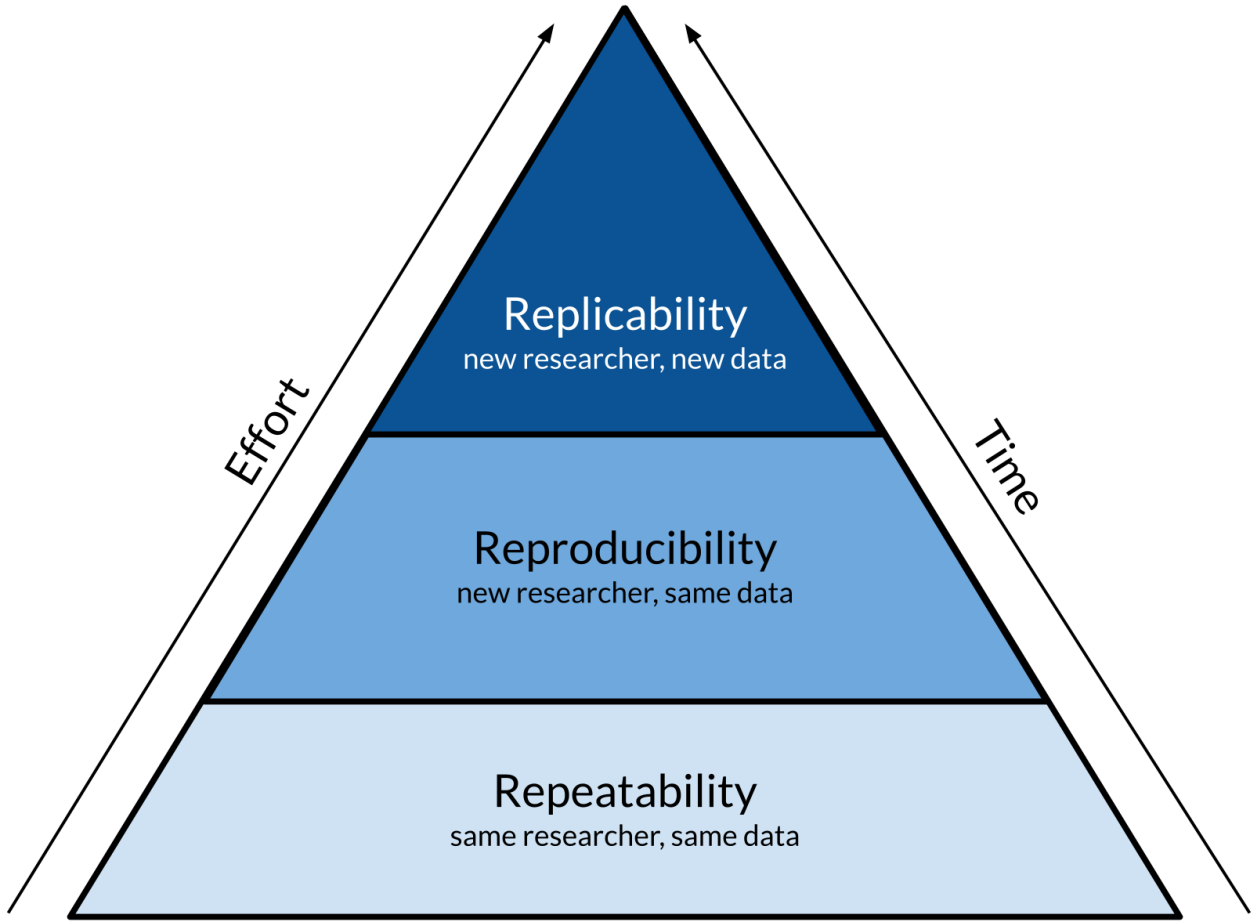
UNIVERSITY OF BERN

Replicable: with new data do we obtain the same inferences?



Public Health Sciences Course Program - Basic Statistics and Projects in R. Slides availab on **GitHub**.

Repeatability, reproducibility, and replicability



Based off of a figure from Essawy et al, 2020 https://doi.org/10.1016/j.envsoft.2020.104753

In between repeatability and reproducibility, there is also ‘runnability’ (same researcher, new machine).

Reproducibility with R(Studio)

- Make the management of the reproducibility of your project a top priority!
- Use R projects (`.Rproj`).
- Use a folder structure that meets the needs of your project (e.g., **https://github.com/ISPMBern/project-template**).
- Use sequential numbers and descriptive names for your files names (e.g., `02_analysis.R`).
- Use a main or master script (e.g., `00_main.R`) from which you can `source` other scripts:

```r
# Main R script that sources all subsequent R scripts

source(here("R/01_cleaning.R")) # source() reads R code from a file
source(here("R/02_analysis.R"))
source(here("R/03_plotting.R"))
```

- Distinguish between `raw` and `processed` data. Never, ever change your raw data!
- Use Markdown files (`README.md`) to provide additional information.

Public Health Sciences Course Program - Basic Statistics and Projects in R. Slides availab on **GitHub**.

What is Markdown?

Markdown is a lightweight markup language for creating formatted text using a plain-text editor.
John Gruber (long-time Mac aficionados may know him from his blog **Daring Fireball**) created the Markdown language in 2004, with **Aaron Swartz** (creater of `atx`) acting as a beta tester. Gruber had the goal of enabling people “to write using an easy-to-read and easy-to-write plain text format, optionally convert it to structurally valid XHTML (or HTML).”
Nowadays, there are many different flavors of Markdown, e.g., **GitHub Flavored Markdown**, which makes it the primary choice to communicate important information about your project/repository.

Markdown syntax - Headings

Markdown Syntax	Output
`# Header 1`	# Header 1
`## Header 2`	## Header 2
`### Header 3`	### Header 3
`#### Header 4`	#### Header 4
`##### Header 5`	##### Header 5
`###### Header 6`	###### *Header 6*

Markdown syntax - Text formatting

UNIVERSITY OF BERN

Markdown Syntax	Output
`*italics* and **bold**`	*italics* and **bold**
`superscript^2^ / subscript~2~`	superscript^2 / subscript$_2$
`~~strikethrough~~`	~~strikethrough~~
`` `verbatim code` ``	`verbatim code`

Public Health Sciences Course Program - Basic Statistics and Projects in R. Slides availab on **GitHub**.

Markdown syntax - Lists

Markdown Syntax	Output

`* unordered list`
`  + sub-item 1`
`  + sub-item 2`
`  - sub-sub-item 1`
	• unordered list
	  ▪ sub-item 1
	  ▪ sub-item 2
	  ○ sub-sub-item 1

`*  item 2`
 	
`  Continued (indent 4 spaces)`
	• item 2
	  Continued (indent 4 spaces)

`1. ordered list`
`2. item 2`
`  i) sub-item 1`
`  A.  sub-sub-item 1`
	1. ordered list
	2. item 2
	  i. sub-item 1
	  a. sub-sub-item 1

`(a) A list whose numbering` `continues after`	1. A list whose numbering

Markdown Syntax	Output
`(@)  an interruption`	continues after 2. an interruption

Markdown syntax - Links and images

UNIVERSITY OF BERN

You can use different types of (hyper)links.

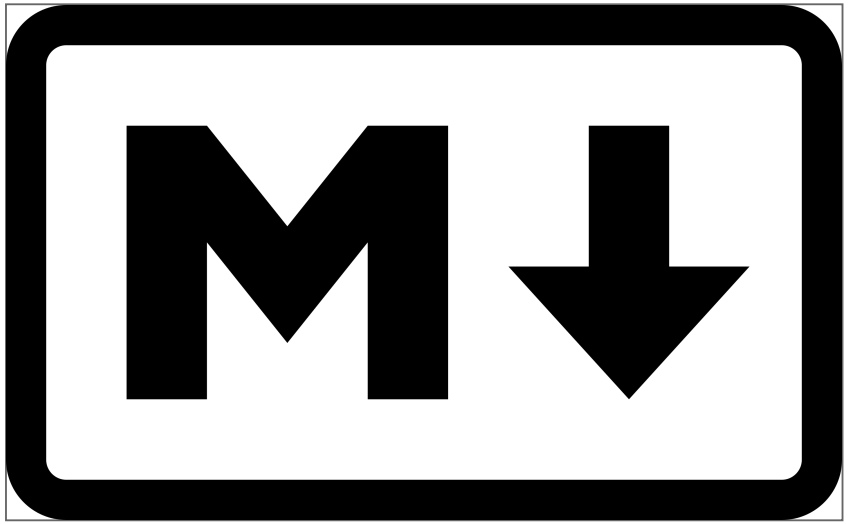
Markdown

```
You can embed [named hyperlinks](https://quarto.org/
direct URL's like <https://quarto.org/>, and links t
[other places](#markdown-syntax---lists) in the docu
The syntax is similar for adding images:

![Markdown logo](figures/markdown.png)
```

Output

You can embed **named hyperlinks**, direct URL's like **https://quarto.org/**, and links to **other places** in the document. The syntax is similar for embedding an inline image:



UNIVERSITY OF BERN

.

Public Health Sciences Course Program - Basic Statistics and Projects in R. Slides availab on **GitHub**.

Markdown syntax - Tables

```
| Right | Left | Default | Center |
|------:|:-----|---------|:------:|
|    12 | 12   |      12 |   12   |
|   123 | 123  |     123 |  123   |
|     1 | 1    |       1 |   1    |
```

Right	Left	Default	Center
12	12	12	12
123	123	123	123
1	1	1	1

Markdown syntax - Quotes

> Let us change our traditional attitude to the construction of programs: Instead of imagining that our main task is to instruct a computer what to do, let us concentrate rather on explaining to human beings what we want a computer to do. - Donald Knuth on Literate Programming[1]

```
1 > Let us change our traditional attitude to the construction of programs: Inste
2 imagining that our main task is to instruct a computer what to do, let us conce
3 rather on explaining to human beings what we want a computer to do. - Donald Kn
4 Literate Programming^[Knuth (1984, Comput J)](https://doi.org/10.1093/comjnl/27
```

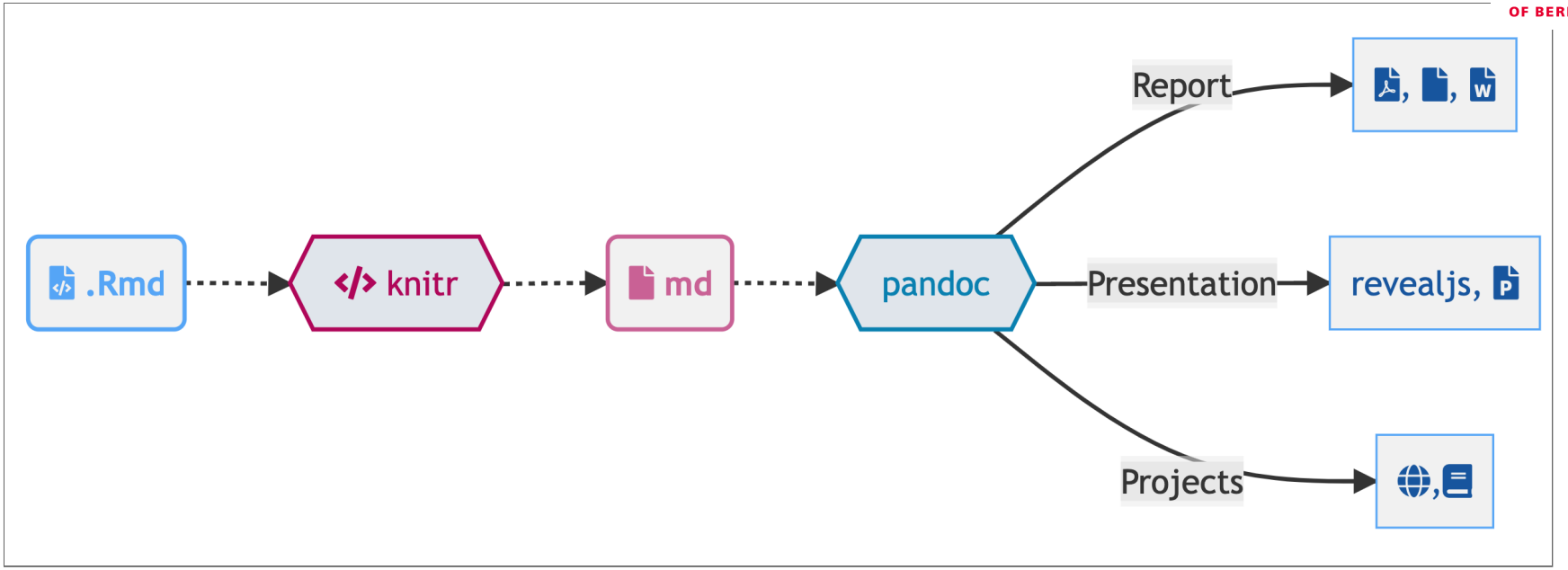
R Markdown

UNIVERSITY OF BERN

Public Health Sciences Course Program - Basic Statistics and Projects in R. Slides availab on **GitHub**.

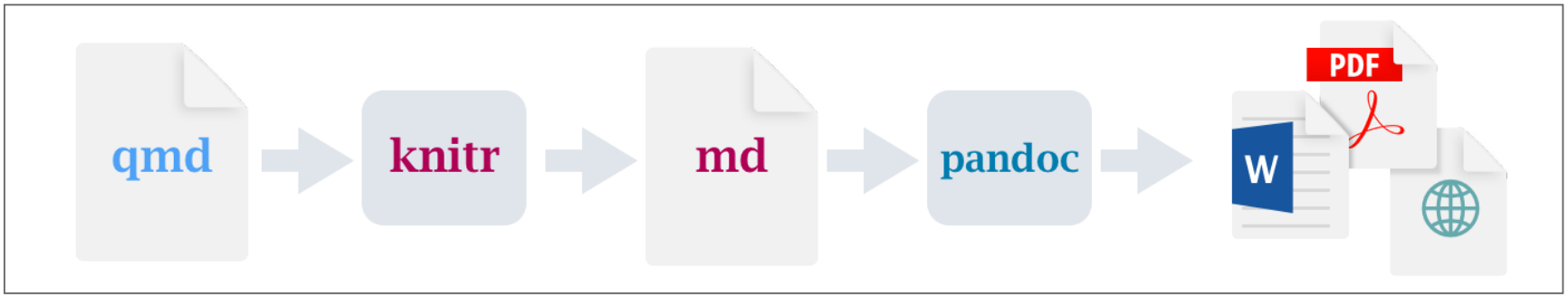
Public Health Sciences Course Program - Basic Statistics and Projects in R. Slides availab on **GitHub**.

Flowchart for R Markdown



So what is Quarto?

R Markdown has been around for roughly a decade and was fundamentally built for R. That's why RStudio (now Posit) developed Quarto, a next-generation, R Markdown-like open-source scientific and technical publishing system built on Pandoc.
Quarto is as friendly to Python, Julia, Observable JavaScript, and Jupyter notebooks as it is to R. It's not a language-specific library, but an external software application.
If you start creating reproducible documents, just use Quarto.



From QMD to HTML

````
---
title: "ggplot2 demo"
author: "Norah Jones"
date: "5/22/2021"
format:
  html:
    fig-width: 8
    fig-height: 4
    code-fold: true
---

## Air Quality

@fig-airquality further explores the impact of temperatu
  on ozone level.

```{r}
#| label: fig-airquality
#| fig-cap: Temperature and ozone level.
#| warning: false

library(ggplot2)
ggplot(airquality, aes(Temp, Ozone)) +
  geom_point() +
  geom_smooth(method = "loess")
```
````

UNIVERSITY OF BERN

Public Health Sciences Course Program - Basic Statistics and Projects in R. Slides availab on **GitHub**.

ggplot2 demo

Norah Jones

May 22nd, 2021

Air Quality

Figure 1 further explores the impact of temperature on ozone level.

▶ Code

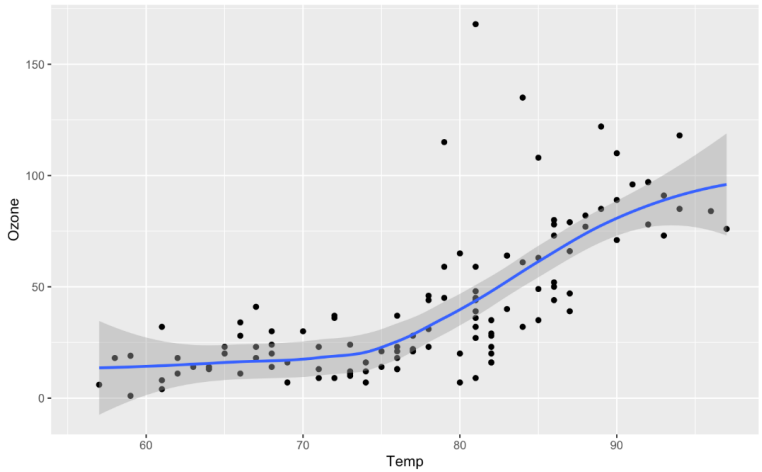


Figure 1: Temperature and ozone level.

UNIVERSITY OF BERN

Public Health Sciences Course Program - Basic Statistics and Projects in R. Slides availab on **GitHub**.

Anatomy of a code chunk

````
```{r}
#| label: car-cyl
#| echo: false
mtcars %>%
  distinct(cyl)
```
````

- Has 3x backticks on each end ```
- Place engine (`r`) between curly braces `{r}`
- Place options underneath, behind the `#|` (hashpipe): `#| option1: value`

Code chunks are controllable

These are just some examples. There are a lot more **options**.

Option	Description
`fig-height: 4`	Plots generated from this chunk will have a height of 4 inches.
`fig-width: 6`	Plots generated from this chunk will have a width of 6 inches.
`dpi: 150`	Plots generated will have a dots per inch (pixel density) of 150
`echo: false`	Code will not be echoed (i.e., not shown)
`eval: false`	Nothing will be evaluated, but code still be printed
`cache: true`	Results will be cached, and chunk will not be run in subsequent renders, unless code is changed.
`message: false`	No messages will be printed

Option	Description
`warning: false`	No warnings will be printed
`include: false`	No ouputs/echo/messages/etc wi returned

Demo

Let’s create a reproducible HTML report!

UNIVERSITY OF BERN

Public Health Sciences Course Program - Basic Statistics and Projects in R. Slides availab on **GitHub**.

Further online material

- **Tutorial: Hello, Quarto**
- rstudio::conf 2022 Workshop: **Getting Started with Quarto**

Exercise: Reproducible report

Not, it's your turn!

1. Open your R project from yesterday.
2. Create a new Quarto document (choose HTML). Think about the best place to save your `.qmd` file.
3. Write a short report that generates one of your figures from yesterday (or even better, a new figure).
4. Render and enjoy your (first?) reproducible HTML report!

One last thing…

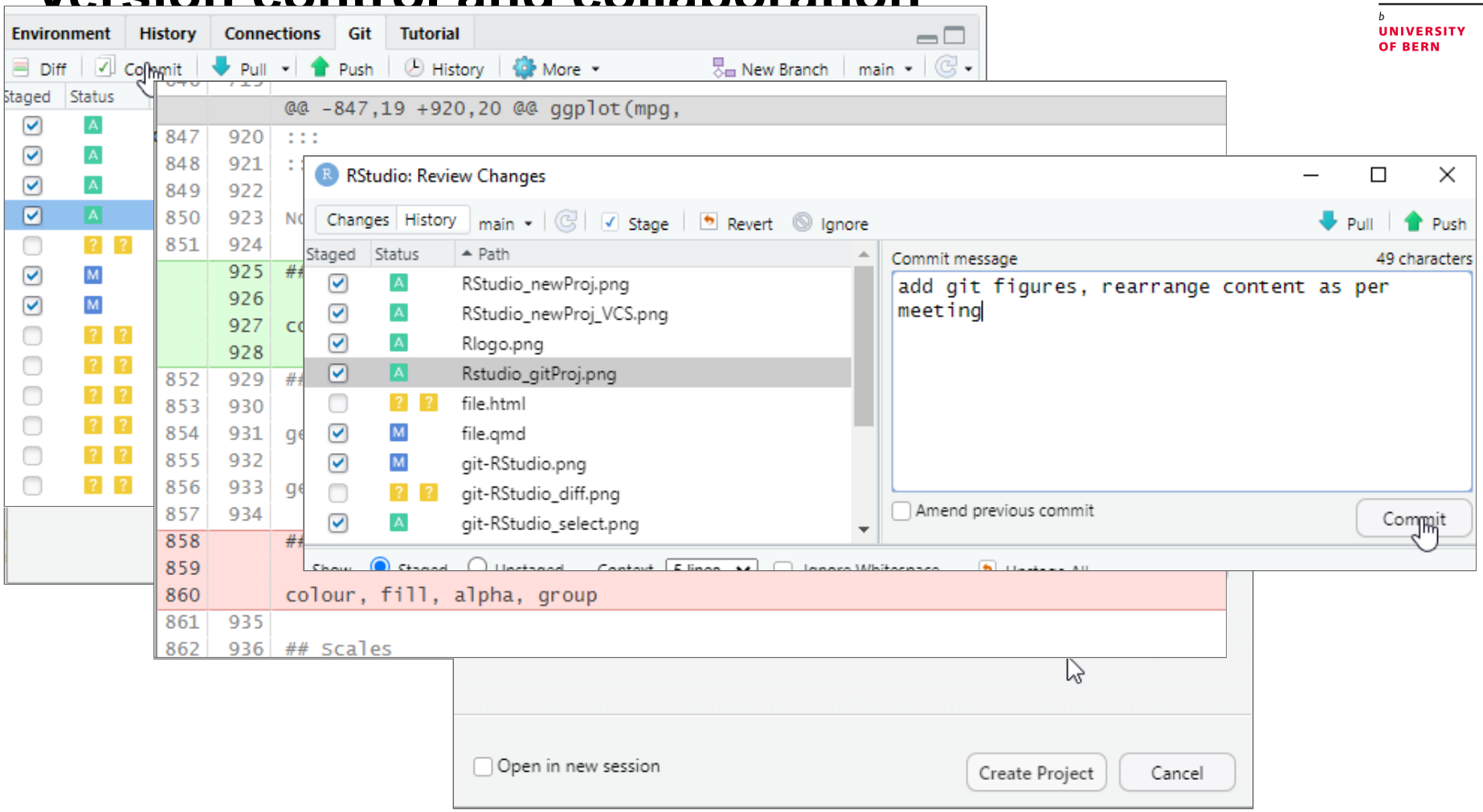
Sometimes, your analysis might depend on specific versions of R (rarely) or packages (more frequent). Regular updates of packages may break existing code. To solve this potential problem, you can use tools to ensure that you (or your collaborators, including you in 5 years time) can reproduce your environment and rerun your analysis:

- `renv` produces a lockfile with all information required to replicate the environment.
- `groundhog` installs the version of a package that existed at a particular point in time. It is slightly more lightweight than `renv`.

Finally, you can add `sessionInfo()` to your reports, which lists the used versions of R and packages.

1. **Knuth (1984) Computers and Typesetting**

Version control and collaboration



Public Health Sciences Course Program - Basic Statistics and Projects in R. Slides availab on **GitHub**.

Version control

```
file.R

file1.R

file1-final.R

file1-final2.R

file1-final2-2023-02.R

file1-final2-2023-02_final.R

...
```

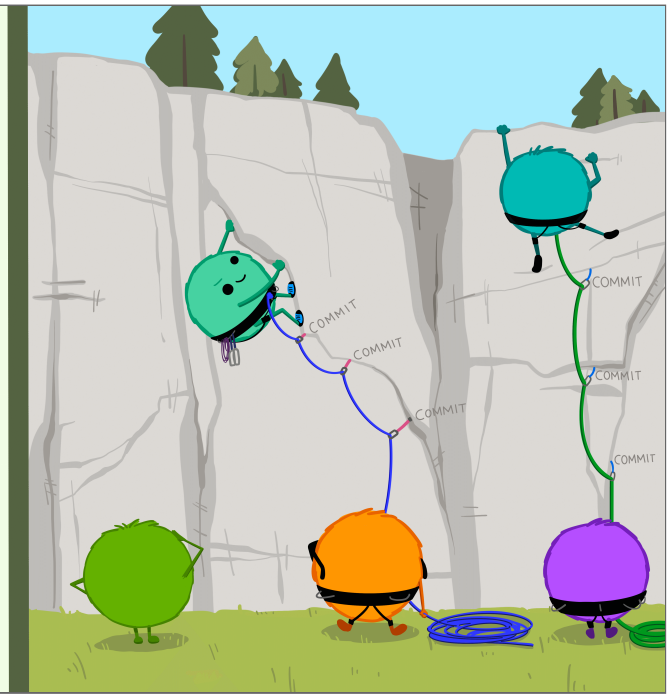
Version control systems allow you to retain a single file, while also preserving the history of the file.
Git and SVN are (probably) the most well known. We will be working with Git and GitHub.

Git

Git is a version control system.

> Using a Git commit is like using anchors and other protection when climbing…**if you make a mistake, you can't fall past the previous commit**.
>
> Commits are also helpful to others, because **they show your journey, not just the destination**.
>
> – HADLEY WICKHAM & JENNY BRYAN

Wickham & Bryan, RPackages (https://r-packages.org/preface.html)



It's not always trivial to use, but **Happy Git and GitHub for the useR** is a super resource especially for R people.

Illustrations from the Openscapes blog GitHub for supporting, contributing, and failing safely by Allison Horst and Julia Lowndes

Public Health Sciences Course Program - Basic Statistics and Projects in R. Slides availab on **GitHub**.

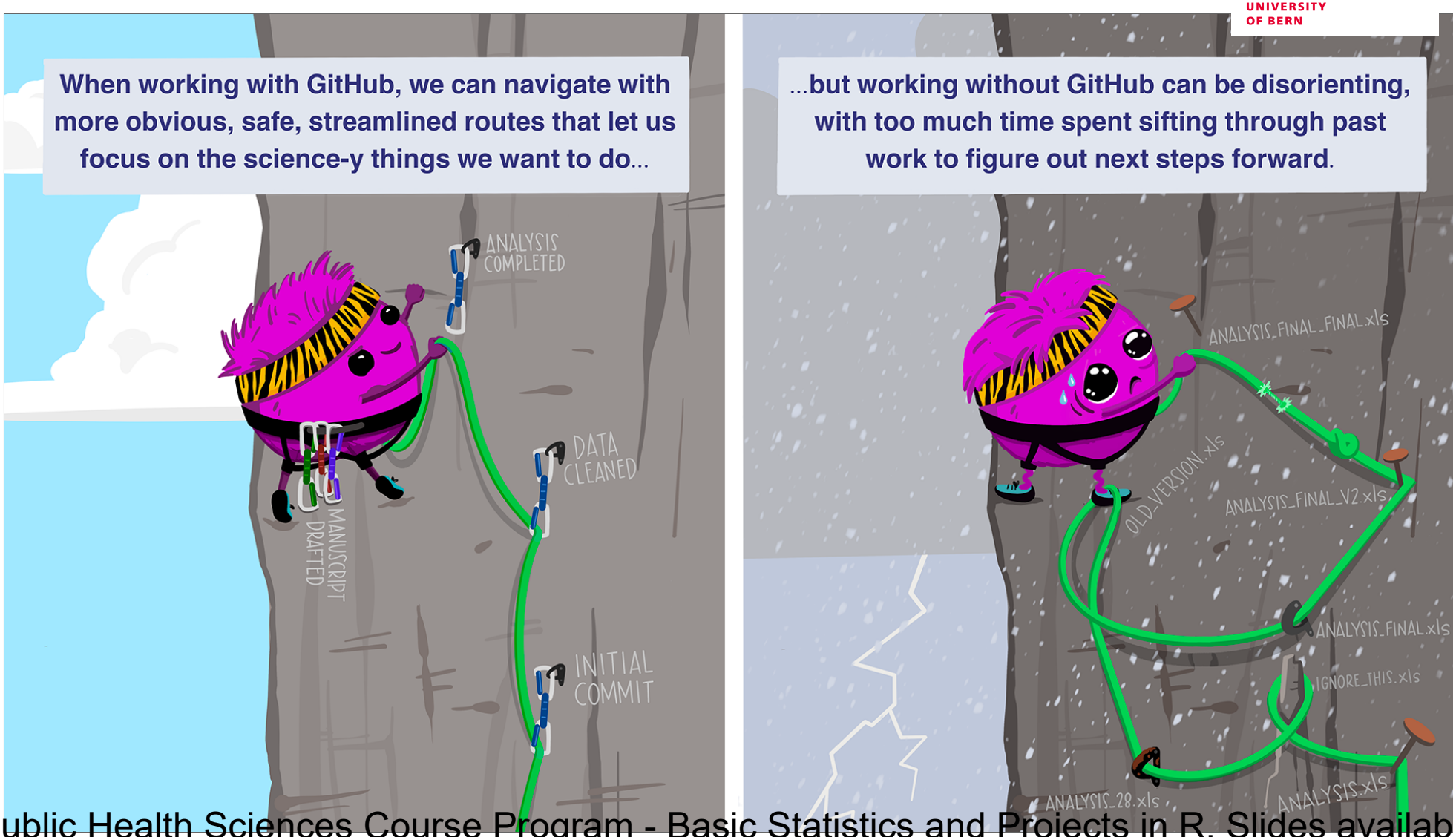
GitHub

GitHub is a website that integrates Git and serves as an online version of your repository. It can also host websites, check R packages, etc.

Illustrations from the Openscapes blog GitHub for supporting, contributing, and failing safely by Allison Horst and Julia Lowndes

Public Health Sciences Course Program - Basic Statistics and Projects in R. Slides availab on **GitHub**.

GitHub



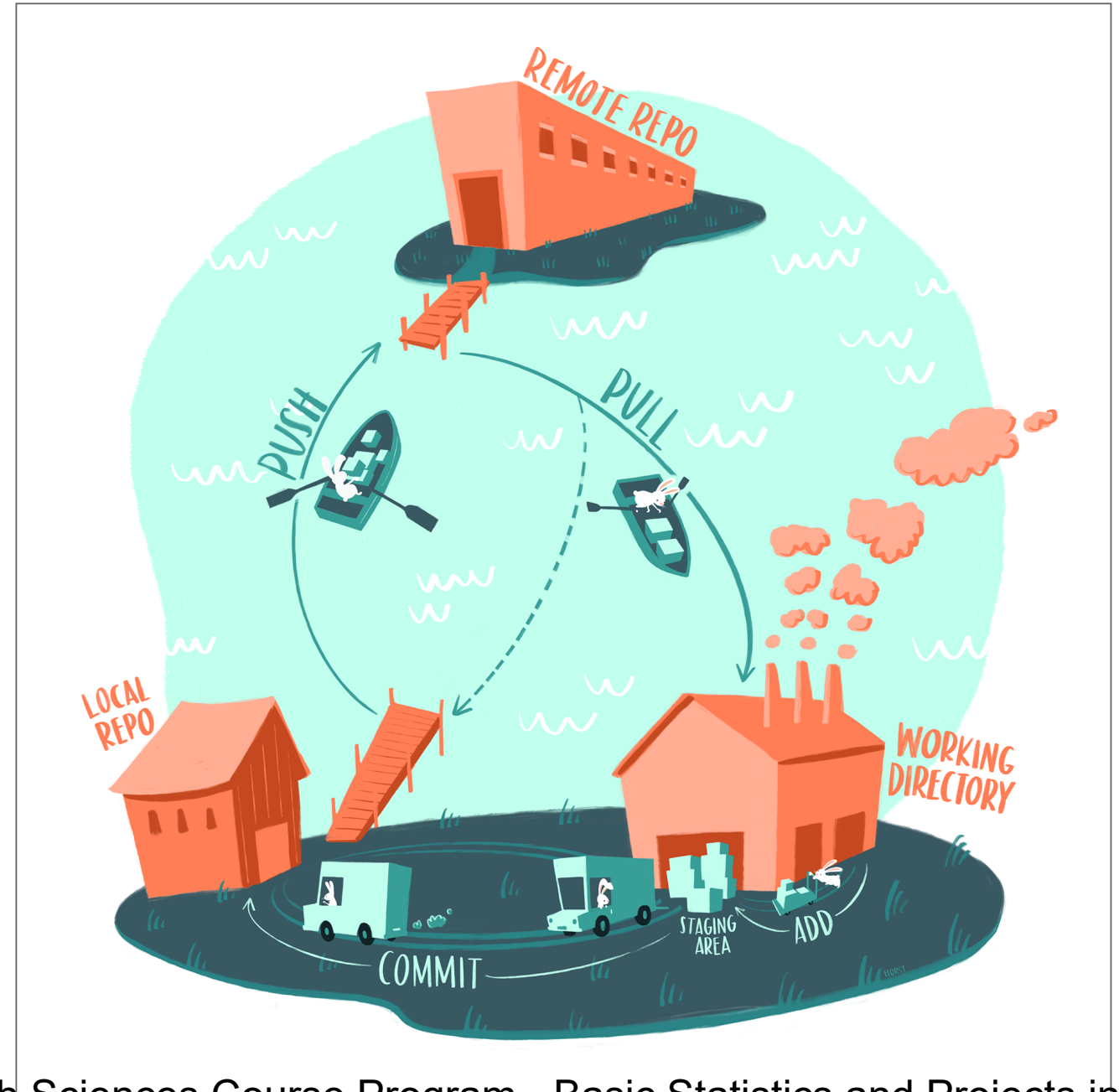
Public Health Sciences Course Program - Basic Statistics and Projects in R. Slides availab on **GitHub**.

Illustrations from the Openscapes blog GitHub for supporting, contributing, a
failing safely by Allison Horst and Julia Lowndes

The Git(Hub) process

UNIVERSITY OF BERN

Public Health Sciences Course Program - Basic Statistics and Projects in R. Slides availab on **GitHub**.


REMOTE REPO
PUSH
PULL
LOCAL REPO
WORKING DIRECTORY
STAGING AREA
ADD
COMMIT

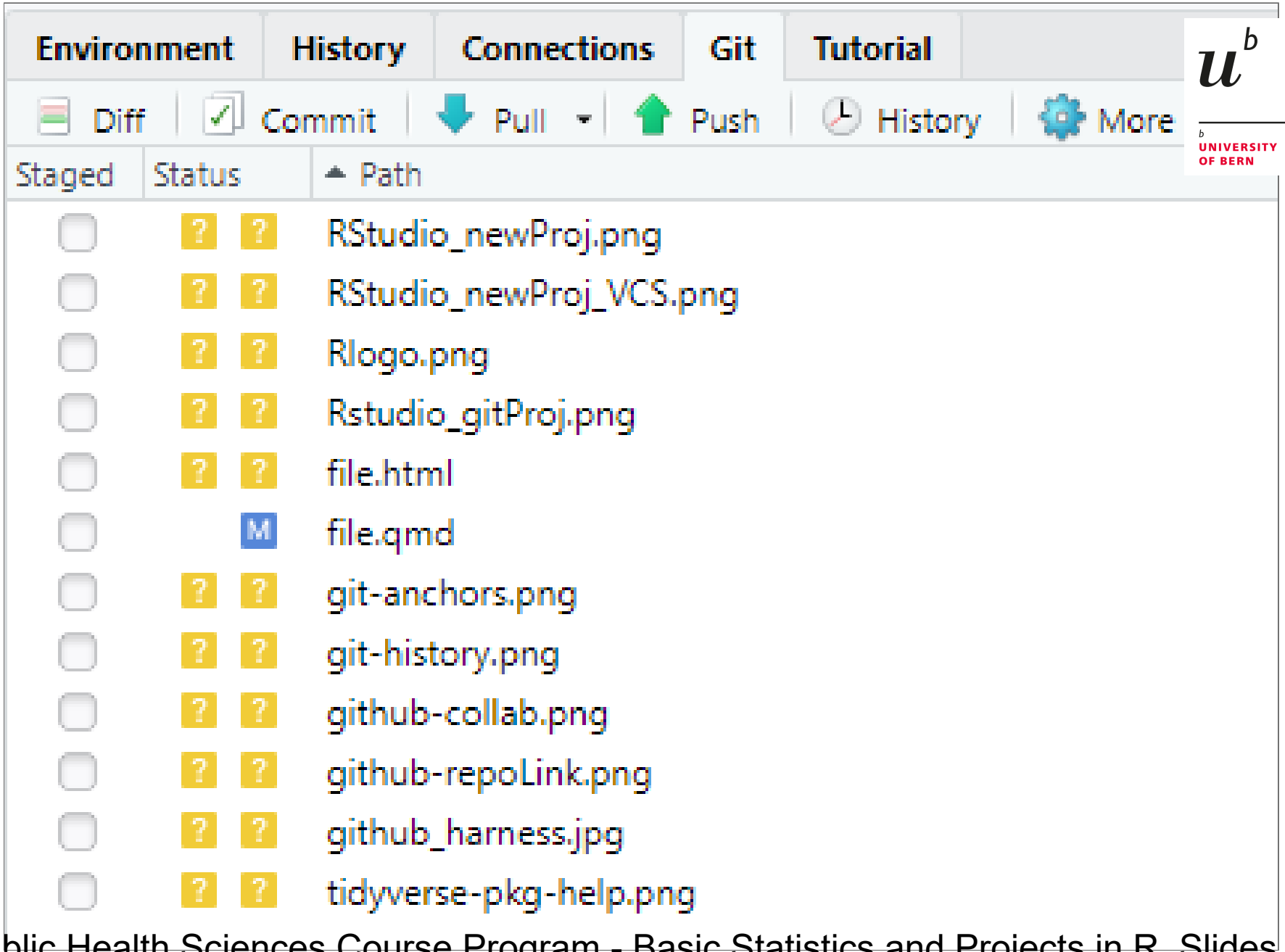
Artwork by @allison_horst

UNIVERSITY OF BERN

Public Health Sciences Course Program - Basic Statistics and Projects in R. Slides availab on **GitHub**.

Working with GitHub in practice

`git` is a command line tool - type commands into the command prompt.
RStudio offers a way to work with Git (or SVN) repositories.



UNIVERSITY OF BERN

Public Health Sciences Course Program - Basic Statistics and Projects in R. Slides availab on **GitHub**.

It can do the most important things, but not everything (terminal use resolve
though).
GitHub also offers a nice desktop app (offering easier authentication).
The following slides assume a GitHub-first approach. This would be my recommended approach.

UNIVERSITY OF BERN

0. Introduce your computer to GitHub

Install Git (from e.g. **https://gitforwindows.org/** or by installing Xcode on macOS) if you don't yet have a Git installation on your system.
In R, install the `usethis` and `gitcreds` packages (`install.packages(c("usethis", "gitcreds"))`).
Type `usethis::create_github_token()` into the R console.
An internet browser window should open with a GitHub page. Accept the default settings and click "Generate token". Leave the window open.
Back in R, type `gitcreds::gitcreds_set()` into the console.
Copy and paste the token from the GitHub window into the R console.
You should now be able to work with your GitHub account from within R.

1. Clone the repository

UNIVERSITY OF BERN

Public Health Sciences Course Program - Basic Statistics and Projects in R. Slides availab on **GitHub**.

2. Make changes

- Open files
- Modify as necessary
- Save changes
- Create new files
- …

3. (Add and) commit

UNIVERSITY OF BERN

Public Health Sciences Course Program - Basic Statistics and Projects in R. Slides availab on **GitHub**.

4. Push back to GitHub

Click the green button to move your committed content to GitHub.

Worthwhile going to your repository to check that it's there, at least the first time.

No GitHub repository yet, but a local project

UNIVERSITY OF BERN

You have your files, but there isn't a related project on GitHub (yet)…
You will not have a Git pane in RStudio…
Initialize Git with `usethis::use_git()` (install the `usethis` R package if you don't already have it). RStudio will/should restart and the Git pane should now be visible.
Commit your changes as before (Git pane, select the files to commit, write your message and click commit).
Connect the project to GitHub via `usethis::use_github()`.
This will create a GitHub repository in your account with the same name as the folder and pushes your commits to date to GitHub.

Your turn!!

Make a Git repository out of the folder you've been using the last 2 days.

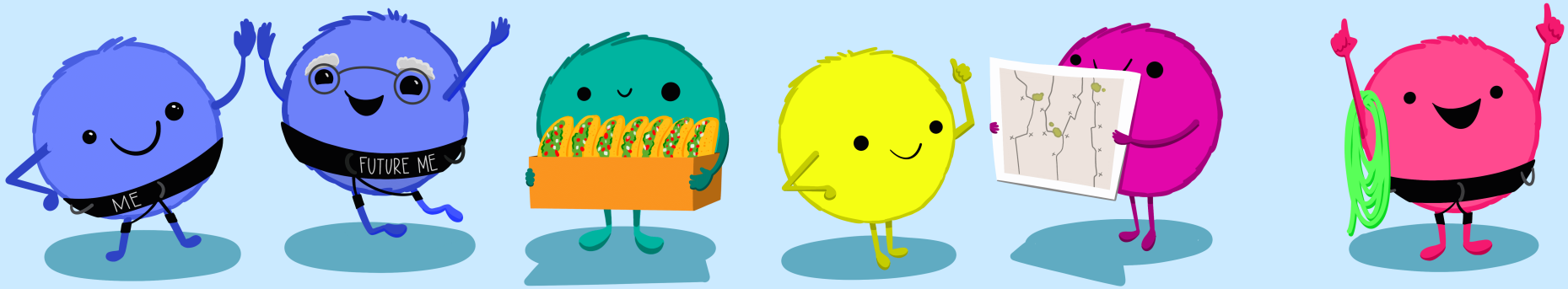
- initialize a Git repository in the folder (`usethis::use_git()`)
- commit your files using appropriate commit messages
- transfer your repository to GitHub (`usethis::use_github()`)
- check that your content is on GitHub
- make a new change and make sure that you know how to get it to GitHub

If you have time, try the GitHub-first approach. Make a new repository on GitHub based on **https://github.com/ISPMBern/project-template** (click "use this template" and "create a new repository"), then clone it to your computer, make a change, and get it back to GitHub.

Collaborating on Git(Hub)

UNIVERSITY OF BERN

> **Collaboration is the most compelling reason to manage a project with Git and GitHub**. My definition of collaboration includes hands-on participation by multiple people, including your past and future self, as well as an asymmetric model, in which some people are active makers and others only read or review.
>
> -JENNY BRYAN

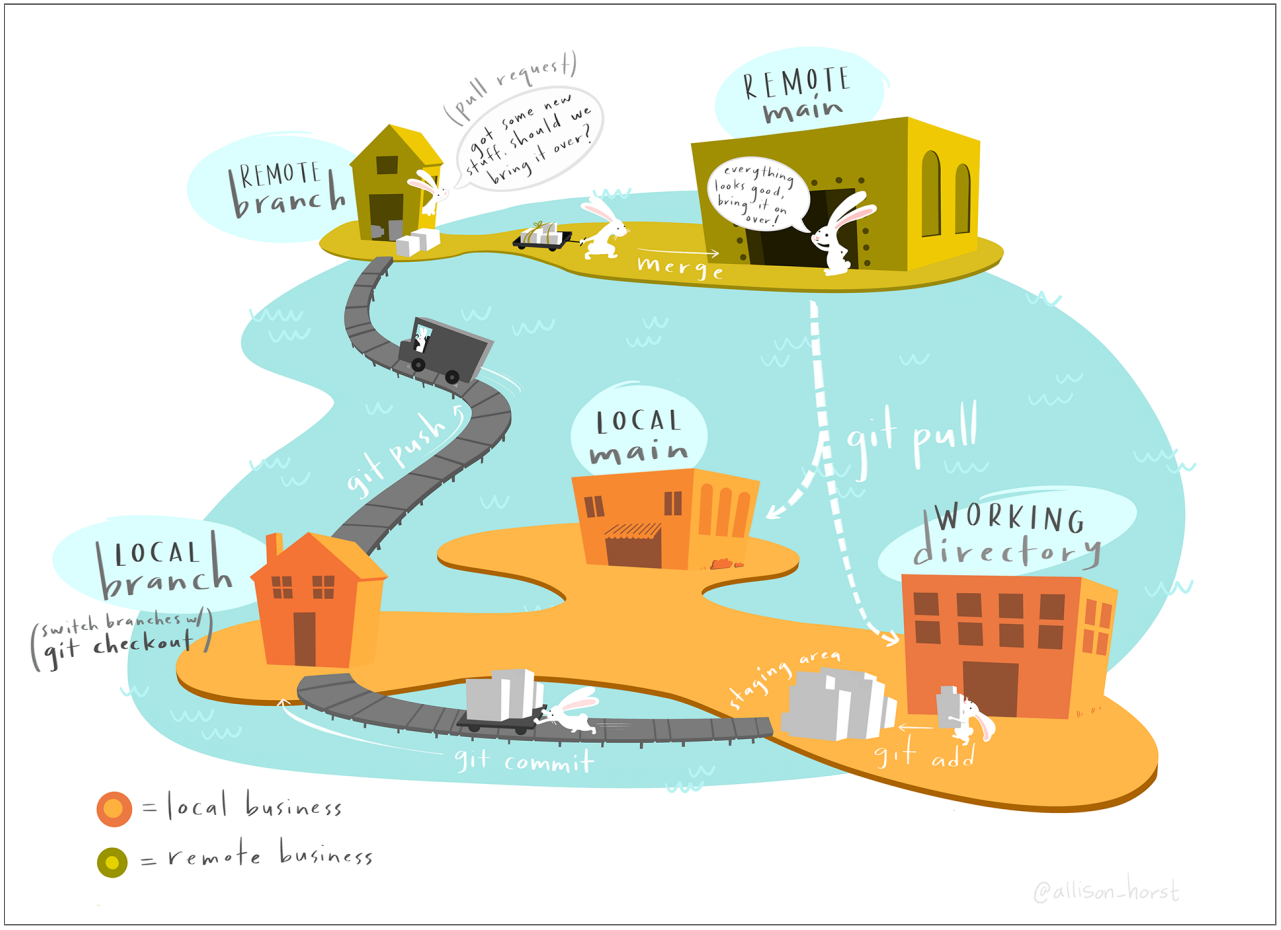


Bryan, J. 2017. Excuse me, do you have a moment to talk about version control? PeerJ Preprints. 5:e3159v2. DOI: 10.7287/peerj.preprints.3159v2

Public Health Sciences Course Program - Basic Statistics and Projects in R. Slides availab on **GitHub**.

Illustrations from the Openscapes blog GitHub for supporting, contributing, a
failing safely by Allison Horst and Julia Lowndes

Collaborating on Git(Hub)



Each collaborator

- forks the repository
- clones their fork

Public Health Sciences Course Program - Basic Statistics and Projects in R. Slides availab on GitHub

- works on their parts
- pushes their work
  to their fork
- requests that the
  main repository
  pulls the changes
  into itself
  Artwork by @allison_horst

UNIVERSITY OF BERN

Collaborating on Git(Hub) - keeping up to date

Good practice to pull from the main repository occasionally (especially when it's a very active repository).
RStudio doesn't have a built-in way to do that though, nor does GitHub Studio - you have to use the command line…
Configure a "remote"… in the command line (e.g. the terminal tab in RStudio).

```
git remote add ispm https://github.com/ISPMBern/projects-in-R.git
git pull ispm main
```

If others are working on related things (e.g. they need the output from your parts), it might be useful to set a remote to their fork so that you can pull their changes and check that your work is still compatible.

Collaborating on Git(Hub) - conflicts

Conflicts occur when two commits (modifications) change the same line(s) to different things.
Git does not know what to do about it, so it needs your help.
Git will issue an error when you try to pull highlighting the issue.
Go to the offending file and find the conflict. It'll look something like this:

```
If you have questions, please
<<<<<<< HEAD
open an issue
=======
ask your question in IRC.
>>>>>>> branch-a
```

Change the file such that it is correct - it might be the top (the remote), it might be the bottom (your version) or some combination of the two.
Commit the file(s) to complete the merge.

Websites

UNIVERSITY OF BERN

Public Health Sciences Course Program - Basic Statistics and Projects in R. Slides availab on **GitHub**.

GitHub Pages

GitHub Pages are public webpages hosted and published through GitHub. You can use GitHub Pages to share results from your research project, host a blog, or even share your résumé.
With the **Quickstart** from GitHub Pages, you can create a user site at **yourusername.github.io**. Let's have a look!

Exercise

Today

1. Take a data set (could be yours, a data set from `medicaldata` or tidytuesday, or something else).
2. Analyze/explore the data including at least one data visualization.
3. Push a short, reproducible HTML report to your GitHub repository.

   The analysis does not need to be anything sophisticated. Just find something interesting in the data and tell a story about it. It does need to be reproducible though (i.e., derived from `.qmd`)!

   Final assesment

   For the final assessment by the end of the course, you will extend your report with additional statistical analyses. Send the link of the final HTML report (don't forget to make your repository public!) to **christian.althaus@unibe.ch** and **ben.spycher@unibe.ch** (Subject: Assessment BSPR course) by 16 June 2023. Optional: Instead of sending us the link to the HTML file in your GitHub Repository (e.g., **https://github.com/calthaus/BSPR-exercises/blob/main/products/reports/report.html**), you better send us the link

Public Health Sciences Course Program > Basic Statistics and Projects in R. Slides availab on **GitHub**.

to the HTML file on your GitHub Pages site (e.g., **https://calthaus.github.io/BSPR-exercises/products/reports/report.htm**

UNIVERSITY OF BERN