Organised, transparent and reproducible science using R, git, and drake


@dbarneche

Diego Barneche

www.diegobarneche.com

UNIVERSITY OF EXETER

Three related problems

1. **Organising stand-alone projects**
   - ✓ Where to keep different files
   - ✓ Reconciling multiple versions of files
2. **Making research reproducible**
   - ✓ Recreate outputs from a paper
   - ✓ Record entire workflow
3. **Efficient R-based workflows**
   - ✓ When to re-run (update) things?

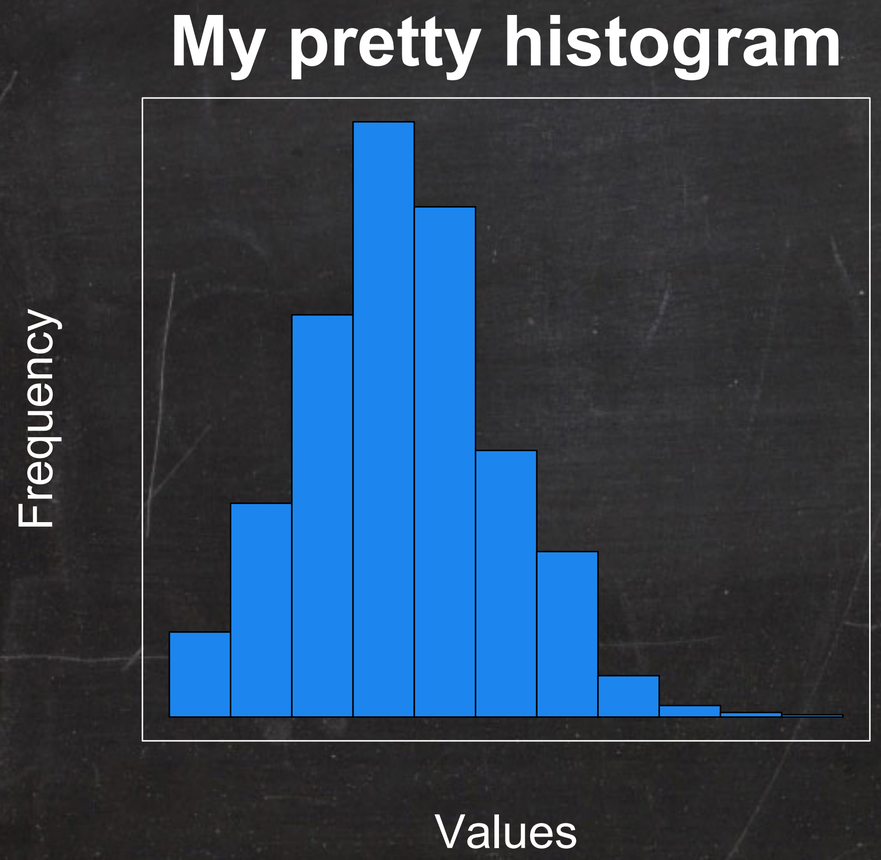
R can be Irreproducible

R can be Irreproducible

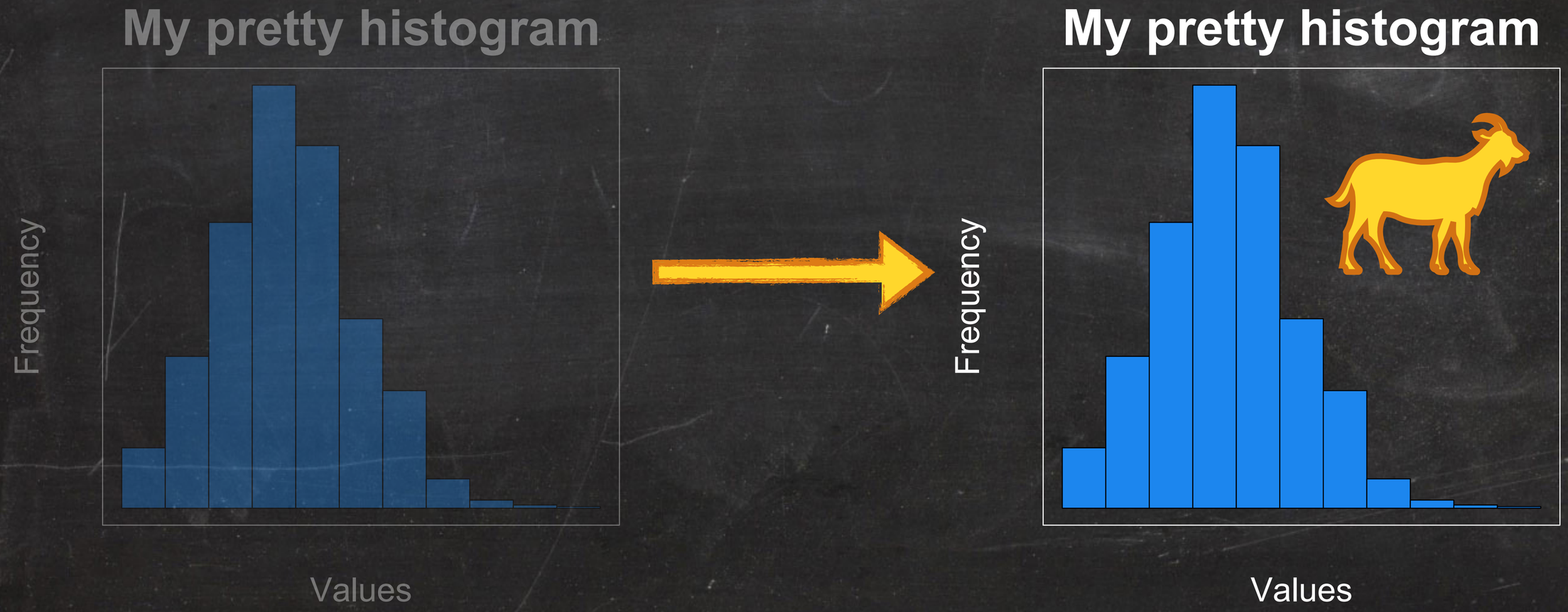
```
setwd("~/Documents/PhD/First_paper/Feb_2019/")

read.csv("~/Documents/PhD/First_paper/Feb_2019/raw_final_final.csv")
```

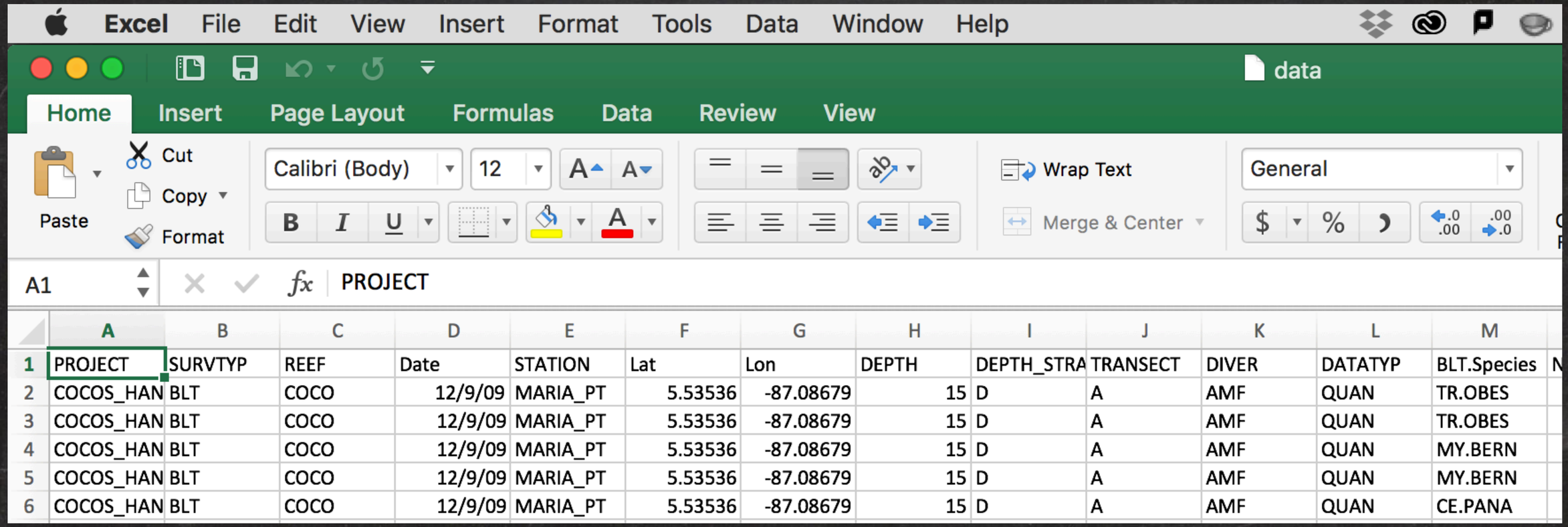
R can be Irreproducible



R can be Irreproducible



R can be Irreproducible



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	PROJECT	SURVTYP	REEF	Date	STATION	Lat	Lon	DEPTH	DEPTH_STRA	TRANSECT	DIVER	DATATYP	BLT.Species
2	COCOS_HAN	BLT	COCO	12/9/09	MARIA_PT	5.53536	-87.08679	15	D	A	AMF	QUAN	TR.OBES
3	COCOS_HAN	BLT	COCO	12/9/09	MARIA_PT	5.53536	-87.08679	15	D	A	AMF	QUAN	TR.OBES
4	COCOS_HAN	BLT	COCO	12/9/09	MARIA_PT	5.53536	-87.08679	15	D	A	AMF	QUAN	MY.BERN
5	COCOS_HAN	BLT	COCO	12/9/09	MARIA_PT	5.53536	-87.08679	15	D	A	AMF	QUAN	MY.BERN
6	COCOS_HAN	BLT	COCO	12/9/09	MARIA_PT	5.53536	-87.08679	15	D	A	AMF	QUAN	CE.PANA

Manually edit data (data.xlsx —> data_v1.xlsx)

R can be Irreproducible

Undocumented dependencies

R can be Reproducible

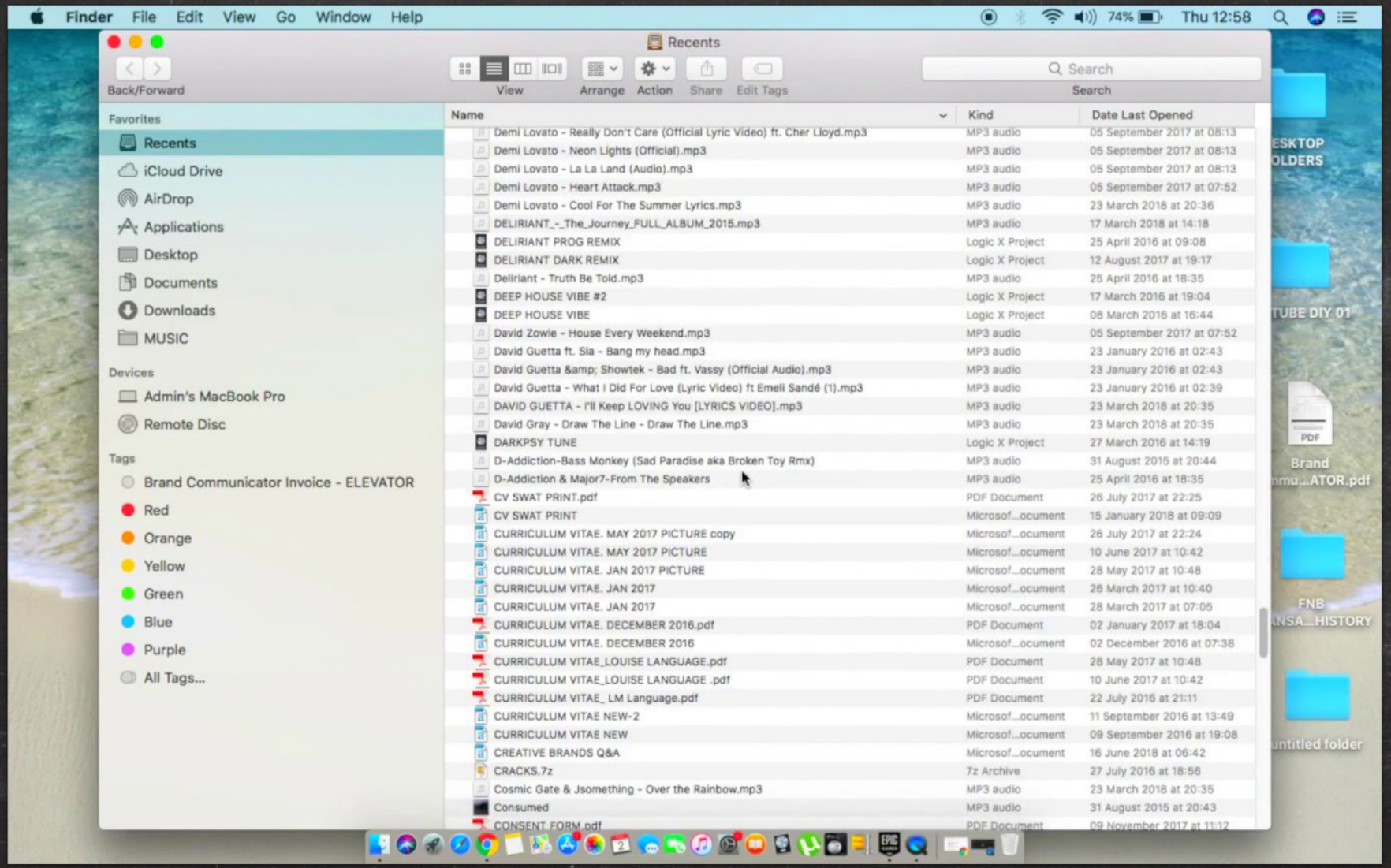
Don’t do any of these things

R can be Reproducible

Today's content

1. Project set up
2. Version control with git
3. GitHub
4. Simple reproducible reports
5. Drake

The Disaster



A niceR solution

```
proj/
|-- R/
|-- data/
|-- output/
|-- |-- data/
|-- |-- figures/
|-- doc/
|-- analysis.R
```

A niceR solution

```
proj/
|-- R/
|-- data/
|-- output/
|-- |-- data/
|-- |-- figures/
|-- doc/
|-- analysis.R
```

The **R** directory contains various files with function definitions (but only function definitions—no code that actually runs).

A niceR solution

```
proj/
|-- R/
|-- data/
|-- output/
|-- |-- data/
|-- |-- figures/
|-- doc/
|-- analysis.R
```

The **data** directory contains data used in the analysis. This is treated as *read only*; in particular the R files are never allowed to write to the files in here. Depending on the project, these might be .csv files, a database, and the directory itself may have subdirectories.

A niceR solution

```
proj/
|-- R/
|-- data/
|-- output/
|-- |-- data/
|-- |-- figures/
|-- doc/
|-- analysis.R
```

The **output/data** directory contains simulation output, processed datasets, logs, or other processed things. The **output/figures** directory contains the output figures generated by your code. Altogether the output directory *only contains generated files*; that is, I should always be able to delete the contents and regenerate them.

A niceR solution

```
proj/
|-- R/
|-- data/
|-- output/
|-- |-- data/
|-- |-- figures/
|-- doc/
|-- analysis.R
```

The **doc** directory contains the paper. The RMarkdown file type can pick up figures directly made by R. With Word you'll have to paste them in yourself as the figures update.

A niceR solution

```
proj/
|-- R/
|-- data/
|-- output/
|-- |-- data/
|-- |-- figures/
|-- doc/
|-- analysis.R
```

In this set up, **analysis.R** is the R script that actually *does* things in the project root. For very simple projects, you might drop the R directory, perhaps replacing it with a single file **analysis-functions.R** which you `source()` within the .R file.

A niceR solution

```
proj/
|-- R/
|-- data/
|-- output/
|-- |-- data/
|-- |-- figures/
|-- doc/
|-- analysis.R
```

```
library(some_package)
library(some_other_package)
source("R/functions.R")
source("R/utilities.R")
```

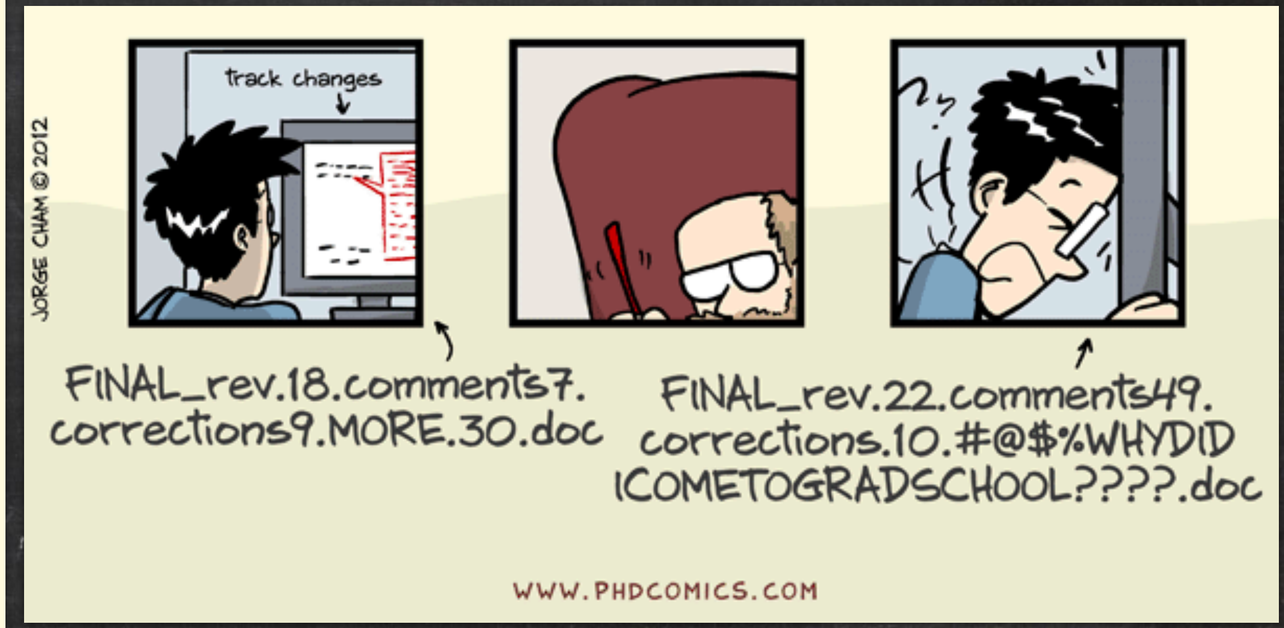
...followed by the code that loads the data, cleans it up, runs the analysis and generates the figures.

A niceR solution

```
gapminder/
|-- R/analysis.R
|-- data/gapminder-FiveYearData.csv
|-- output/
|-- |-- data/
|-- |-- figures/
|-- doc/
|-- analysis.R
|-- gapminder.Rproj
```

Task

Version control



Version control

- Code is doing odd things now and didn't used to.
- Deleted some code and want to get it back.
- Show your supervisor what you did last week.
- See what your collaborators wrote last week.
- Get the previous version of MS back.
- Experiment and try different strategies.
- Have an audit-able project history.

Version control

You might already be using some form of version control

```
## My file (c) John Snow
## Created: 2018/10/04
## Modified: 2019/04/04
```

Version control

You might already be using some form of version control

```
## My file (c) John Snow
## Created: 2018/10/04
## Modified: 2019/04/04
```

- **Repetitive and boring**
- **Difficult to extract the information easily**
- **No checking on the contents of the fields**

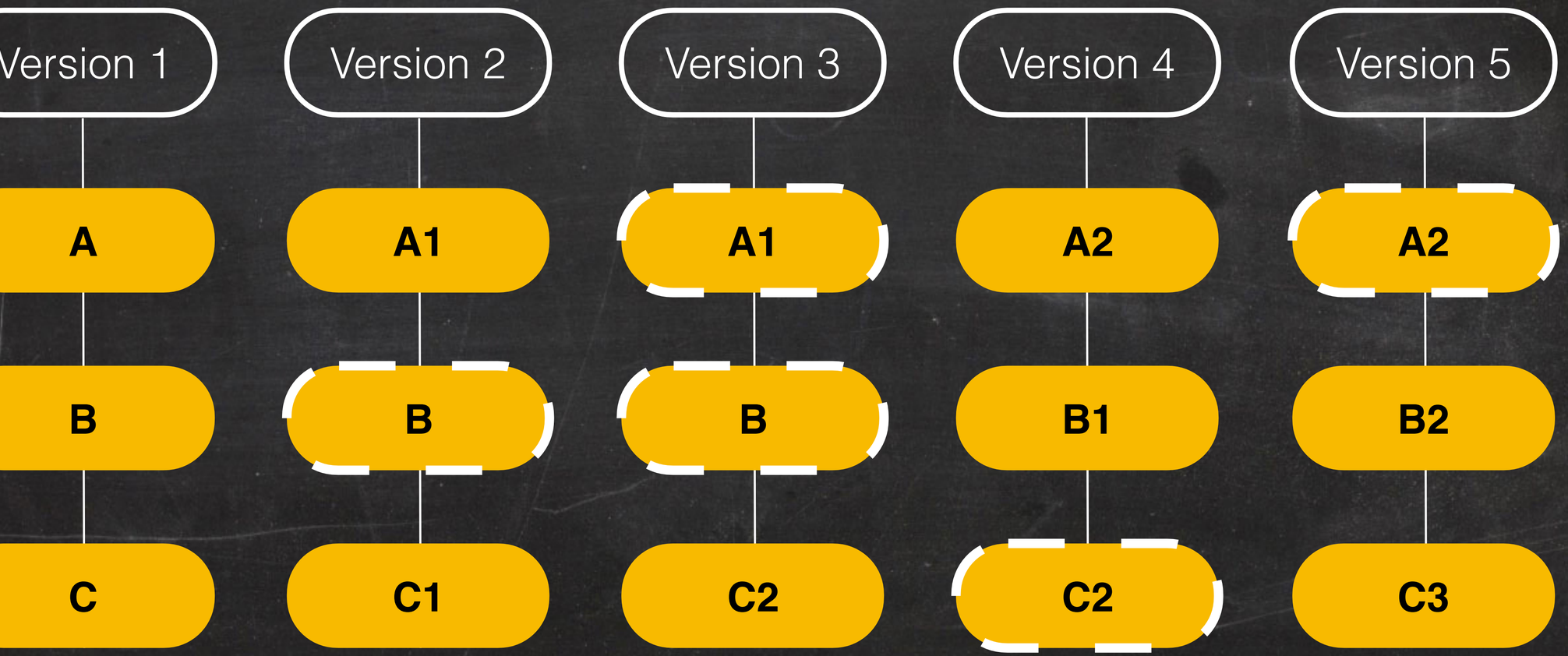
Version control

You might already be using some form of version control

```
## My file (c) John Snow
## Created: 2018/10/04
## Modified: 2019/04/04
```

- **Repetitive and boring**
- **Difficult to extract the information easily**
- **No checking on the contents of the fields**

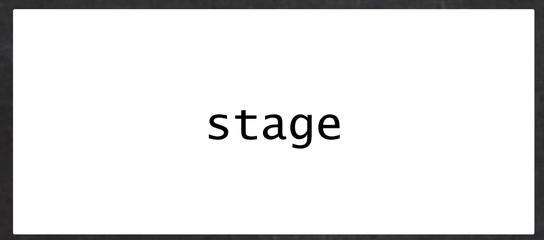
Version control



Version control

just a folder on your computer

$ git add



get files ready to commit

$ git commit

the permanent history of changes

Acknowledgements

@dbarneche

This material is not in anyway original

It was heavily based on the nice R code blog and The Carpentries Foundation

Special thanks to:

JJ Valletta
TJ McKinley
Charlotte Brand

for setting up, planning, and co-teaching this workshop

Daniel Falster
Rich FitzJohn

for creating the nice R code blog

UNIVERSITY OF EXETER