# Basic elements of file management

## ME 447/547 Visualizing Data

Richard Layton

March 2020

Rose-Hulman Institute of Technology

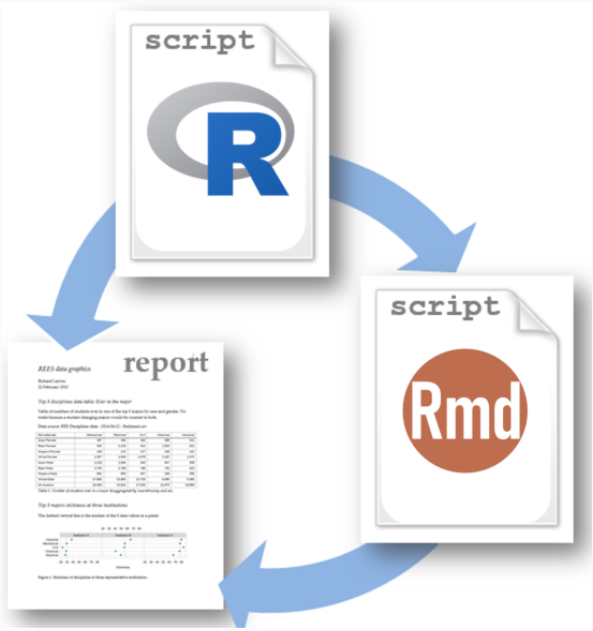
# Effective file management begins when the project begins

Plan your directory structure

- carpentry
- data
- data-raw
- design
- figures
- reports
- README.Rmd
- portfolio.Rproj

Plan a file-naming scheme

Use relative file paths to link files

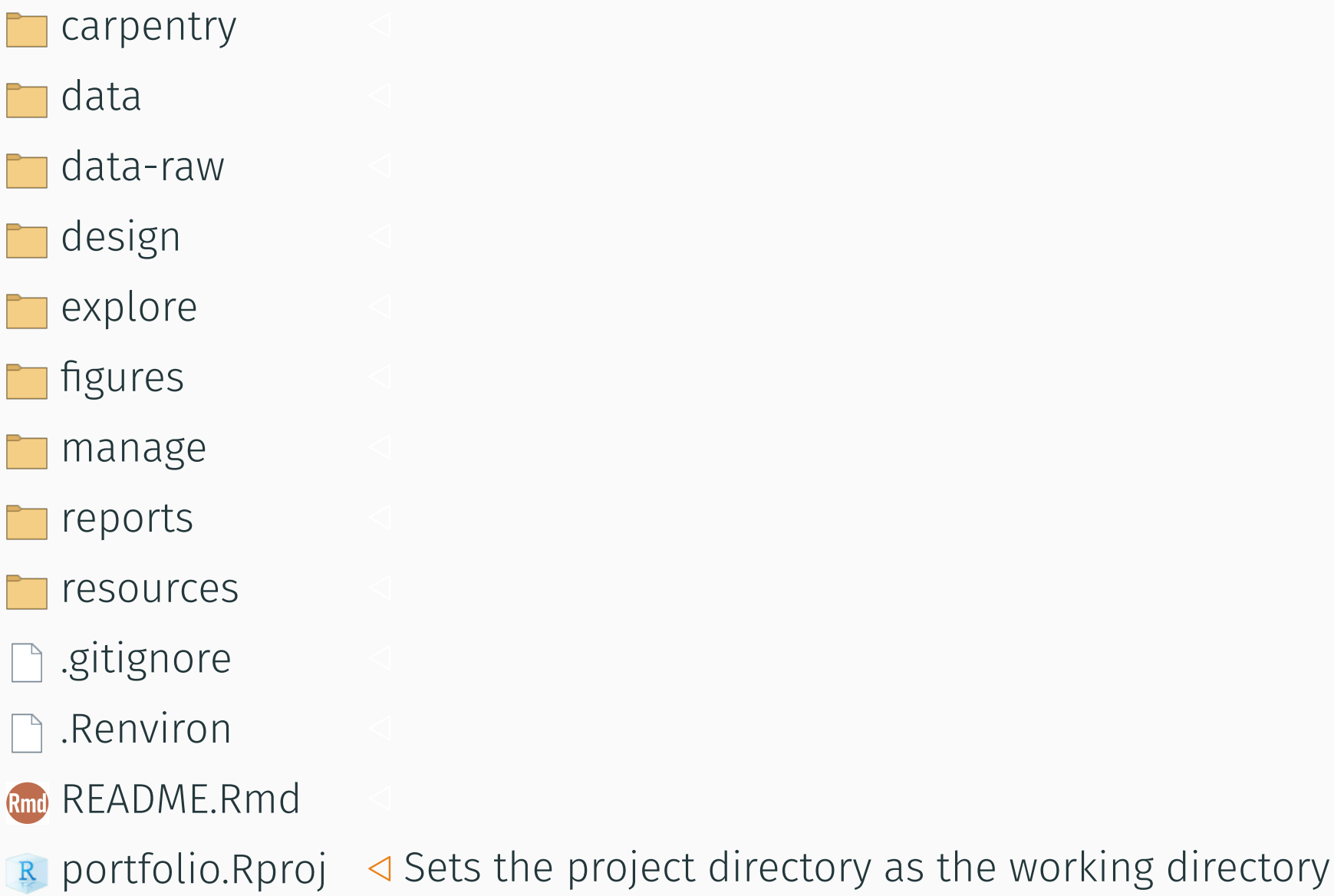


# Planning the structure

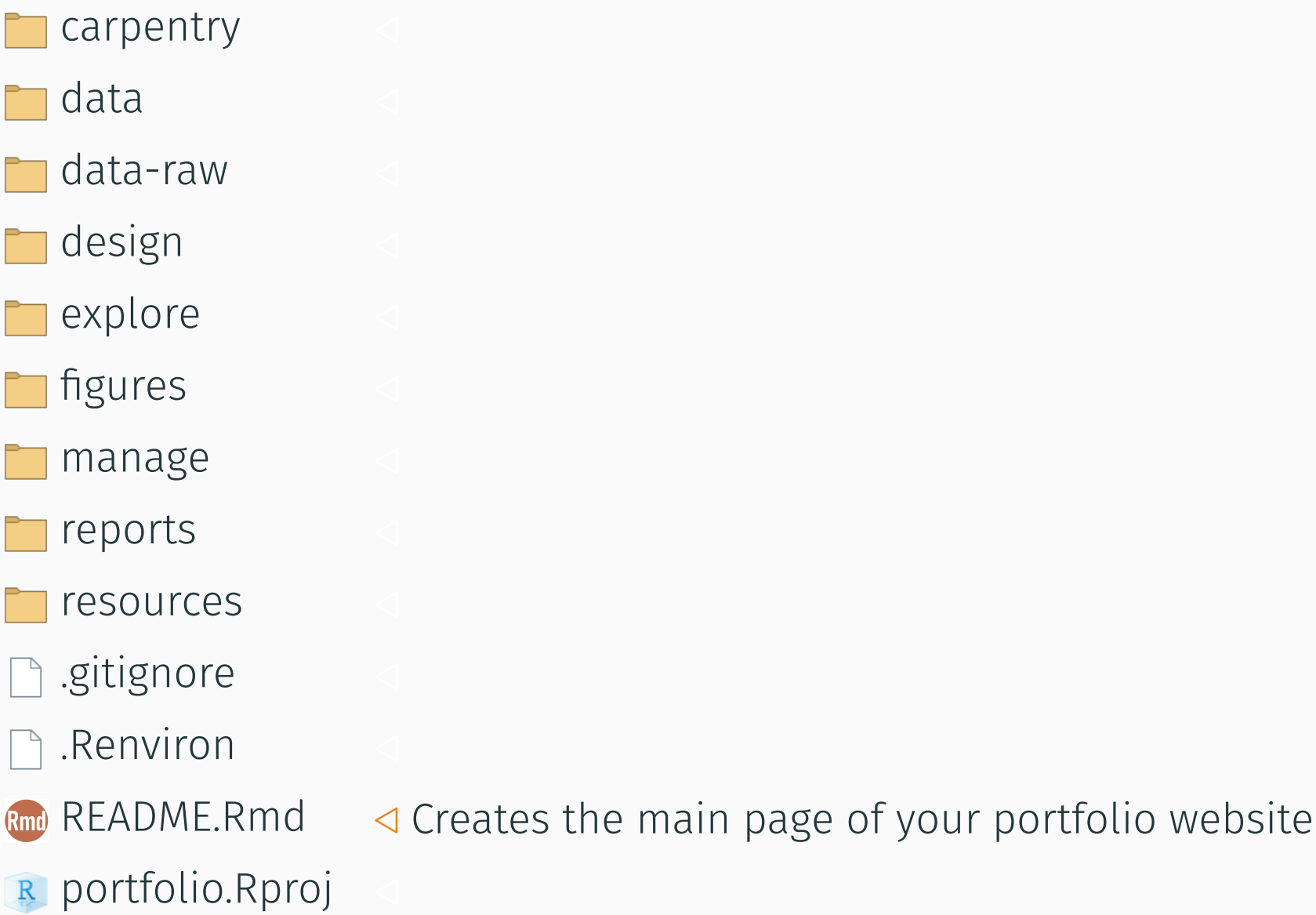
# The portfolio project has a mandatory structure

- carpentry
- data
- data-raw
- design
- explore
- figures
- manage
- reports
- resources
- .gitignore
- .Renviron
- README.Rmd
- portfolio.Rproj

# Open portfolio.Rproj to start every work session



# README introduces your portfolio to the reader



# Other top-level files perform administrative duties

- carpentry
- data
- data-raw
- design
- explore
- figures
- manage
- reports
- resources
- .gitignore ◁ Directs Git to ignore specific files
- .Renviron ◁ Stores packages in a library separate from base R
- README.Rmd
- portfolio.Rproj

# Raw data are never edited manually

- carpentry
- data
- data-raw ◁ Data in its original form
- design
- explore
- figures
- manage
- reports
- resources
- .gitignore
- .Renviron
- README.Rmd
- portfolio.Rproj

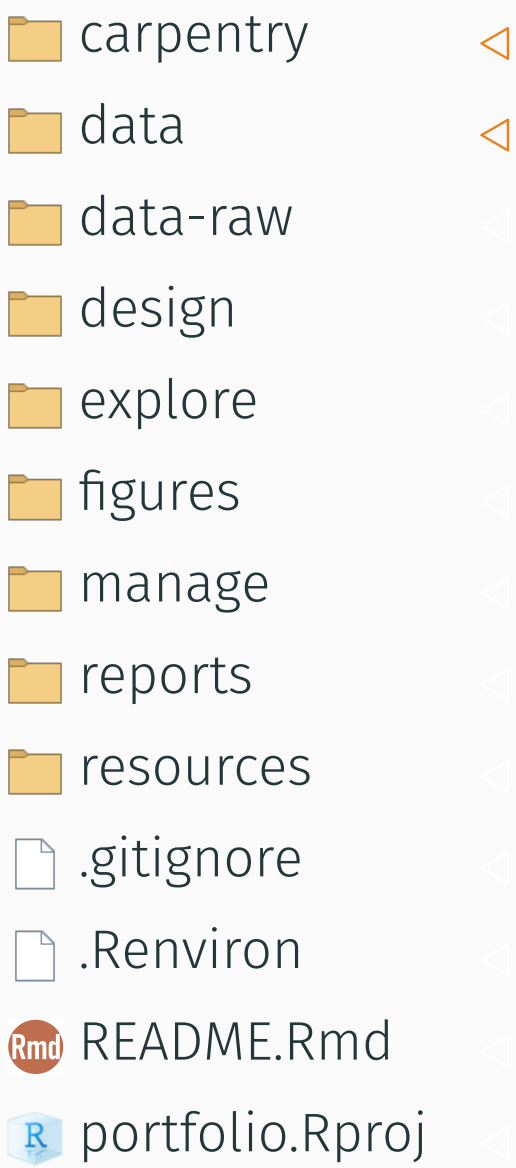
# Explore data structure before tidying

- carpentry
- data
- data-raw ◁ Data in its original form
- design
- explore ◁ R scripts for exploring data and potential graphs
- figures
- manage
- reports
- resources
- .gitignore
- .Renviron
- README.Rmd
- portfolio.Rproj

# Data carpentry converts raw data to tidy data

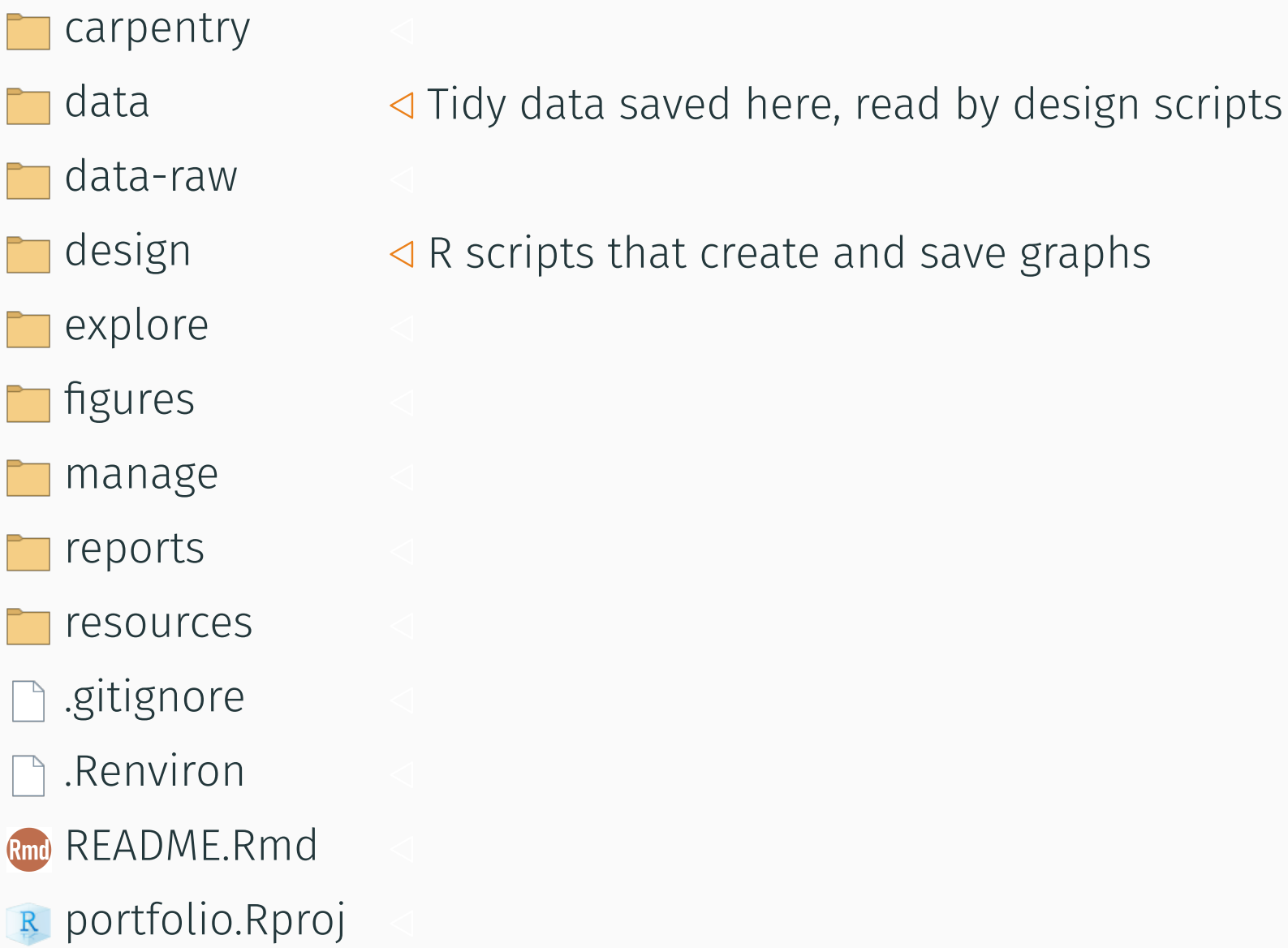
- carpentry ◁ R scripts that turn raw data into tidy data
- data ◁
- data-raw ◁ Data in its original form
- design ◁
- explore ◁
- figures ◁
- manage ◁
- reports ◁
- resources ◁
- .gitignore ◁
- .Renviron ◁
- README.Rmd ◁
- portfolio.Rproj ◁

# Data carpentry converts raw data to tidy data




- R scripts that turn raw data into tidy data
- Tidy data saved here, read by design scripts

# Graph design converts to tidy data to graphs



# Graph design converts to tidy data to graphs

- carpentry
- data
- data-raw
- design ◁ R scripts that create and save graphs
- explore
- figures ◁ Graphs saved here, imported by report scripts
- manage
- reports
- resources
- .gitignore
- .Renviron
- README.Rmd
- portfolio.Rproj

# Reports commingle data, scripts, graphs, prose, and references

- carpentry
- data
- data-raw
- design
- explore
- figures ◁ Graphs saved here, imported by report scripts
- manage
- reports ◁ Reports draw from data, graphs and resources
- resources ◁ Image downloads and bibliography files
- .gitignore
- .Renviron
- README.Rmd
- portfolio.Rproj

# README creates the main page of your portfolio website

- carpentry
- data
- data-raw
- design
- explore
- figures
- manage
- reports ◁ Reports draw from data, graphs and resources
- resources
- .gitignore
- .Renviron
- README.Rmd ◁ Provides explicit links to every report
- portfolio.Rproj

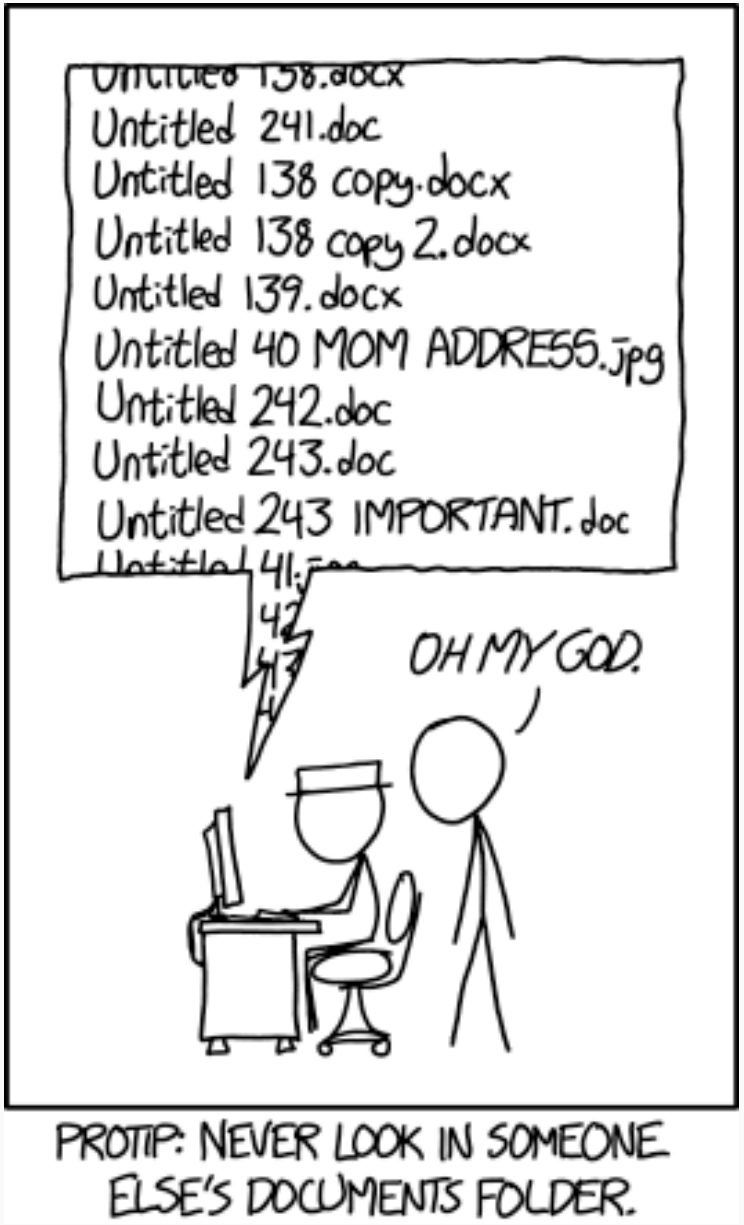
# Reduce clutter by excusing some resources from version control

- carpentry
- data
- data-raw ◁ Version control optional
- design
- explore ◁ Scripts for learning new packages, functions, etc.
- figures
- manage ◁ Correspondence and project management
- reports
- resources
- .gitignore ◁ Directs Git to ignore specific files
- .Renviron
- README.Rmd
- portfolio.Rproj

# Summary: Use the given directory structure for the portfolio

| | |
|---|---|
| carpentry | ◁ R scripts that create and save tidy data |
| data | ◁ Tidy data saved here, read by design scripts |
| data-raw | ◁ Data in its original form (version control optional) |
| design | ◁ R scripts that create and save graphs |
| explore | ◁ R scripts for exploration and learning new skills |
| figures | ◁ Graphs saved here, imported by report scripts |
| manage | ◁ Correspondence and project management |
| reports | ◁ One report per display type |
| resources | ◁ Image downloads and bibliography files |
| .gitignore | ◁ Directs Git to ignore specific files |
| .Renviron | ◁ Stores packages in a library separate from base R |
| README.Rmd | ◁ Creates the main page of your portfolio website |
| portfolio.Rproj | ◁ Sets the project directory as the working directory |

# Naming files




Source: https://xkcd.com/1459/

# Three basic principles should guide your choice of filenames

Filenames should be machine readable

- use delimiters “_” and “-” instead of spaces
- avoid symbols, punctuation marks, and case-sensitivity

Filenames should be human readable

- include information about the file content

Filenames should be friendly to default ordering

- start filenames with a numeric ID
- use leading zeros, e.g., 001, 002, ..., 999

# A sample set of portfolio file names illustrates the principles

Numeric display ID starts every file name: d1, d2, ..., d7

Hyphenated content-information supports human readability

All lowercase, no special symbols, no spaces

```
explore/      d1-strip-plot-speedski-explore.R
carpentry/    d1-strip-plot-speedski-data.R
data/         d1-strip-plot-speedski-data.rds
design/       d1-strip-plot-speedski.R
figures/      d1-strip-plot-speedski.png
reports/      d1-strip-plot-speedski.Rmd
```

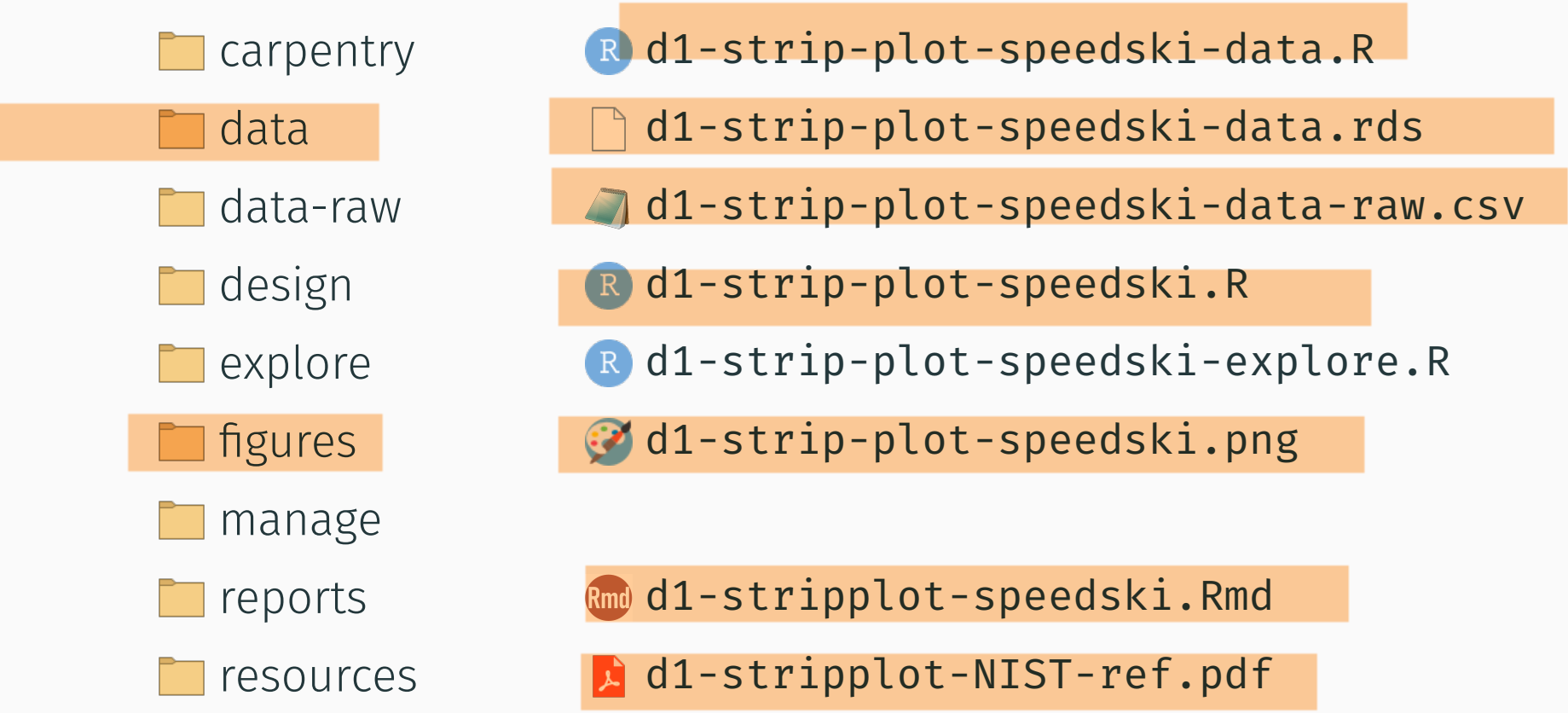
# Add a sequence number `01`, `02`, etc., for related files

Sequential numbers indicate the order in which related files are run.

```
carpentry/ d1-strip-plot-speedski-data-01.R
           d1-strip-plot-speedski-data-02.R
           d1-strip-plot-speedski-data-03.R
```

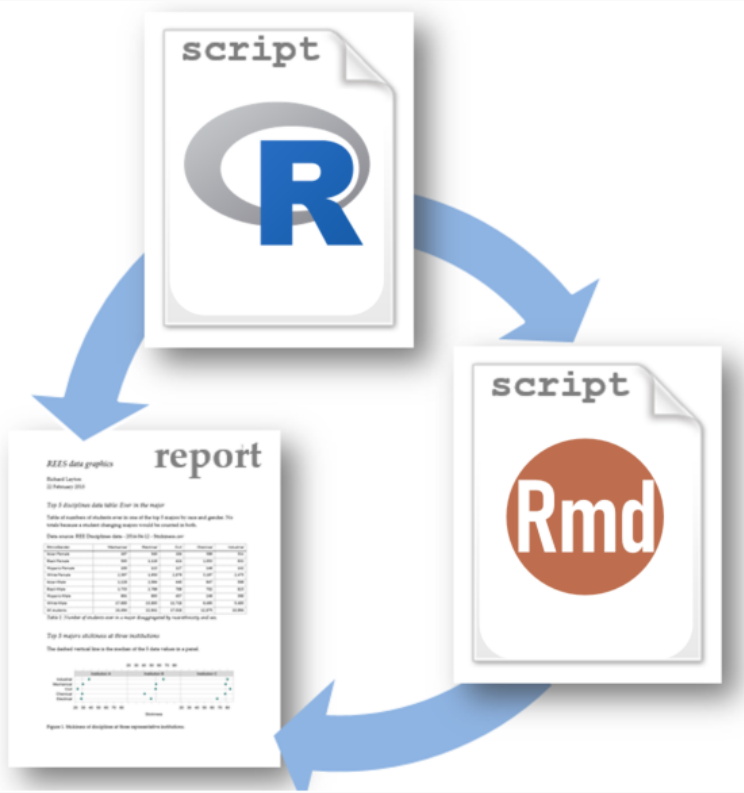
For version control, use git, not sequential file numbers.

# With a plan for managing files, we can start writing them



# Linking files

# Explicitly linking files supports reproducibility




Remove all ambiguity about what files are used to create a report

portfolio.Rproj sets the working directory and supports relative file paths

# Relative file paths document the data tidying workflow

Write an R script for data tidying

carpentry/d1-strip-plot-speedski-data.R

that reads the raw data, prepares it for graphing,

```
read_csv("data-raw/d1-strip-plot-speedski-data-raw.csv")
```

and writes the tidy data frame to the data directory.

```
saveRDS(mydata, "data/d1-strip-plot-speedski-data.rds")
```

# Relative file paths document the graph design workflow

Write an R script for graph design

design/d1-strip-plot-speedski.R

that reads the tidy data, creates the graph,

```
readRDS("d1-strip-plot-speedski-data.rds")
```

and writes the image to the figures directory.

```
ggsave("figures/d1-strip-plot-speedski.png")
```

# The report script runs all the files in order

Write an Rmd report script containing the report text

- reports/d1-stripplot-speedski.Rmd

interleaved with Rmd code chunks that run every R script,

- `source("carpentry/d1-strip-plot-speedski-data.R")`
- `source("design/d1-strip-plot-speedski.R")`

import data to print a data table,

- `readRDS("data/d1-strip-plot-speedski-data.rds")`

and import figures.

- `include_graphics("figures/d1-strip-plot-speedski.png")`

# The README script includes links to each report

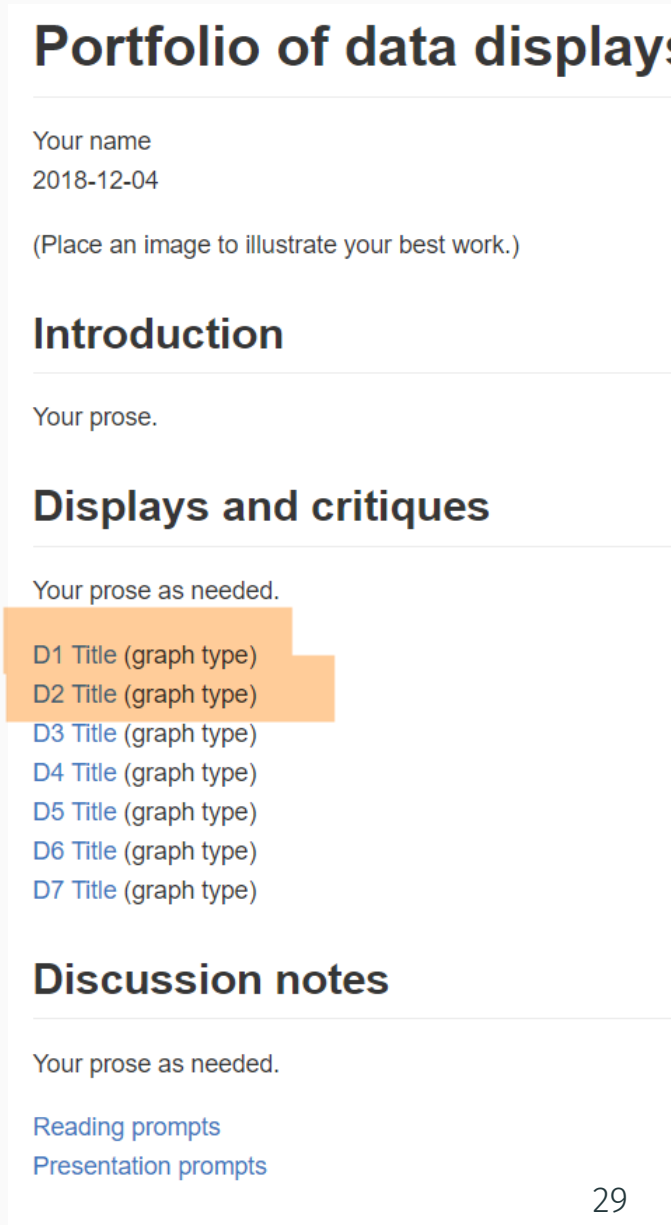
README.Rmd

creates the portfolio main webpage

```
## Displays and critiques

Your prose as needed.

[D1 Title](reports/d1-strip-plot-speedski.md)
[D2 Title](reports/d2_report.md)
[D3 Title](reports/d3_report.md)
[D4 Title](reports/d4_report.md)
[D5 Title](reports/d5_report.md)
[D6 Title](reports/d6_report.md)
[D7 Title](reports/d7_report.md)
```



Portfolio of data displays

Your name
2018-12-04

(Place an image to illustrate your best work.)

## Introduction

Your prose.

## Displays and critiques

Your prose as needed.

D1 Title (graph type)
D2 Title (graph type)
D3 Title (graph type)
D4 Title (graph type)
D5 Title (graph type)
D6 Title (graph type)
D7 Title (graph type)

## Discussion notes

Your prose as needed.

Reading prompts
Presentation prompts

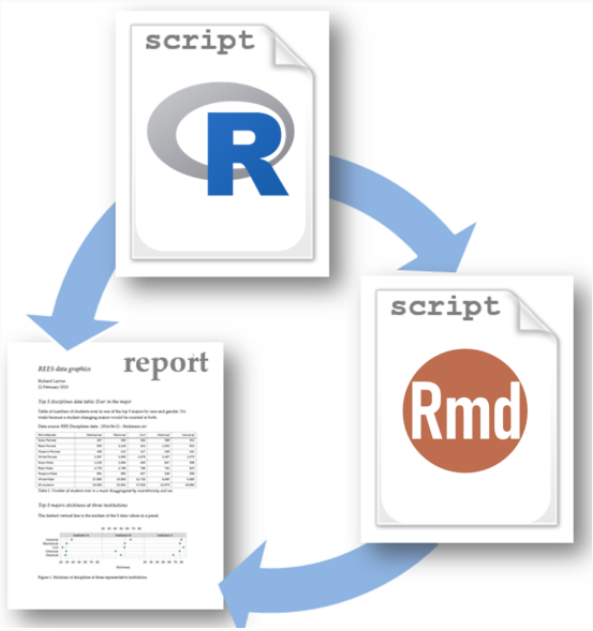
# Effective file management begins when the project begins

Plan your directory structure

- carpentry
- data
- data-raw
- design
- figures
- reports
- README.Rmd
- portfolio.Rproj

Plan a file-naming scheme

Use relative file paths to link files



# References


Bryan J (2015) Naming things. https://speakerdeck.com/jennybc/how-to-name-files

Bryan J (2018) Excuse me, do you have a moment to talk about version control? *The American Statistician* **72**(1), 20–27 (doi:10.1080/00031305.2017.1399928)

Wilson G, Bryan J, Cranston K, Kitzes J, Nederbragt L and Teal TK (2017) Good enough practices in scientific computing. *PLoS Computational Biology* **13**(6) (doi:10.1371/journal.pcbi.1005510)