Basic elements of file management

ME 447/547 Visualizing Data

Richard Layton

December 2018

Rose-Hulman Institute of Technology

Effective file management starts at the beginning of a project

Planning the structure

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

Naming files

```
d7_survey-data-raw.csv
d7_div-stack-bar.R
d7_div-stack-bar.png
d7_report.Rmd
```

Linking files

```
source("carpentry/d7_extract-and-tidy.R")
source("design/d7_div-stack-bar.R")
source("data/d7_survey-data.csv")
include_graphics("figures/d7_div-stack-bar.png")
```

Planning the structure

The portfolio project has a mandatory structure

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

Open portfolio.Rproj to start every work session

- carpentry
- data
- data-raw
- design
- figures
- manage
- practice
- reports
- resources
- .gitignore
- .Renviron
- README.Rmd
- portfolio.Rproj ◁ Sets the project directory as the working directory

README introduces your portfolio to the reader

- carpentry
- data
- data-raw
- design
- figures
- manage
- practice
- reports
- resources
- .gitignore
- .Renviron
- README.Rmd ◁ Creates the main page of your portfolio website
- portfolio.Rproj

Other top-level files perform administrative duties

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

Data carpentry converts raw data to tidy data

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

- R scripts that create and save tidy data
- Tidy data saved here, read by design scripts

Graph design converts to tidy data to graphs

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

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

- carpentry
- data
- data-raw
- design
- figures
- manage
- practice
- reports ◁ One Rmd report per graph
- resources
- .gitignore
- .Renviron
- README.Rmd ◁ Provides links to individual reports
- portfolio.Rproj

Resource files support the portfolio appearance and format

- carpentry
- data
- data-raw
- design
- figures
- manage
- practice
- reports ◁ Reports explicitly call on resource files
- resources ◁ Image downloads and bibliography files
- .gitignore
- .Renviron
- README.Rmd
- portfolio.Rproj

Reduce clutter by excusing some resources from version control

- carpentry
- data
- data-raw
- design
- figures
- manage ◁ Correspondence and project management
- practice ◁ Scripts for practicing and learning R
- reports
- resources
- .gitignore ◁ Directs Git to ignore specific files
- .Renviron
- README.Rmd
- portfolio.Rproj

Summary

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
design	◁ R scripts that create and save graphs
figures	◁ Graphs saved here, imported by report scripts
manage	◁ Correspondence and project management
practice	◁ Scripts for practicing and learning R
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

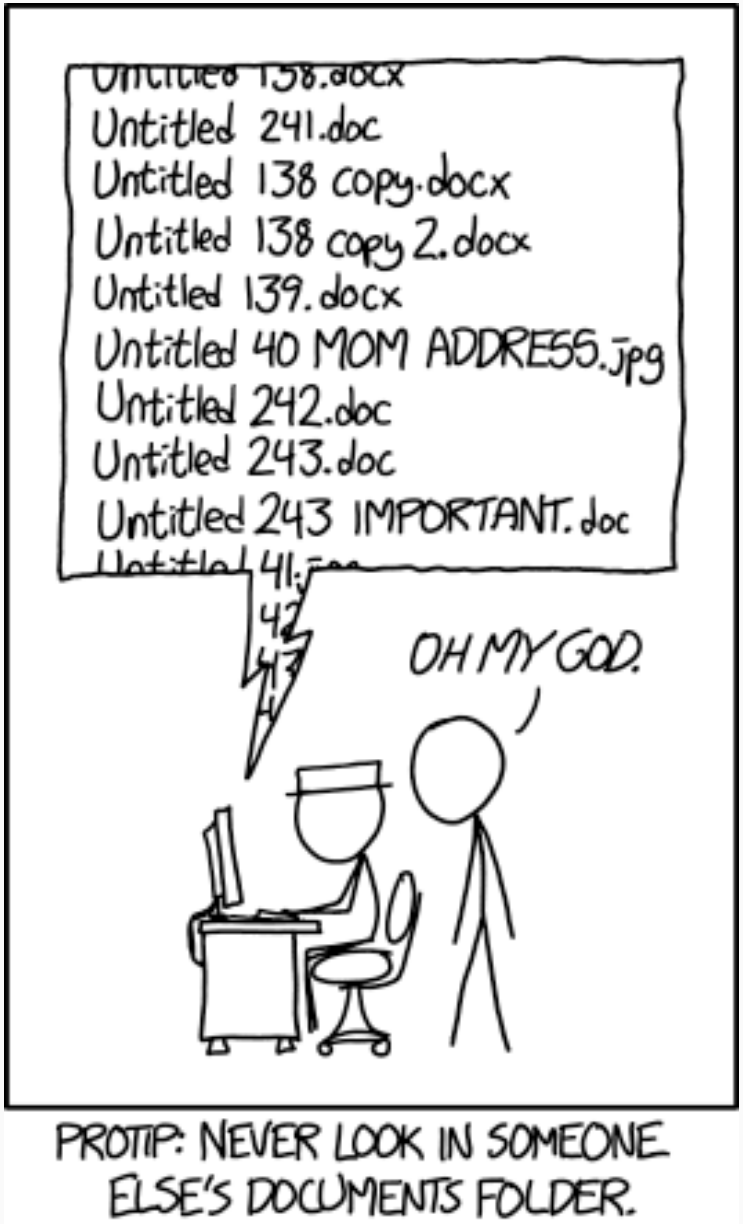
Summary

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

Use the given directory structure for the portfolio.

On future projects, your mileage may vary. You will probably adapt this structure to meet the needs of the new project.

Naming files



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

Fail to plan

A file-naming scheme

And after a time

You are the meme.

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

```
carpentry/ d7_extract-and-tidy.R
data/      d7_survey-data.csv
data-raw/  d7_survey-data-raw.csv
design/    d7_div-stack-bar.R
figures/   d7_div-stack-bar.png
reports/   d7_report.Rmd
```

All lowercase, no special symbols, no spaces

Underscores support machine readability

Add logical ordering when a process requires several files

Add a number `01`, `02`, etc., when related files are run in order

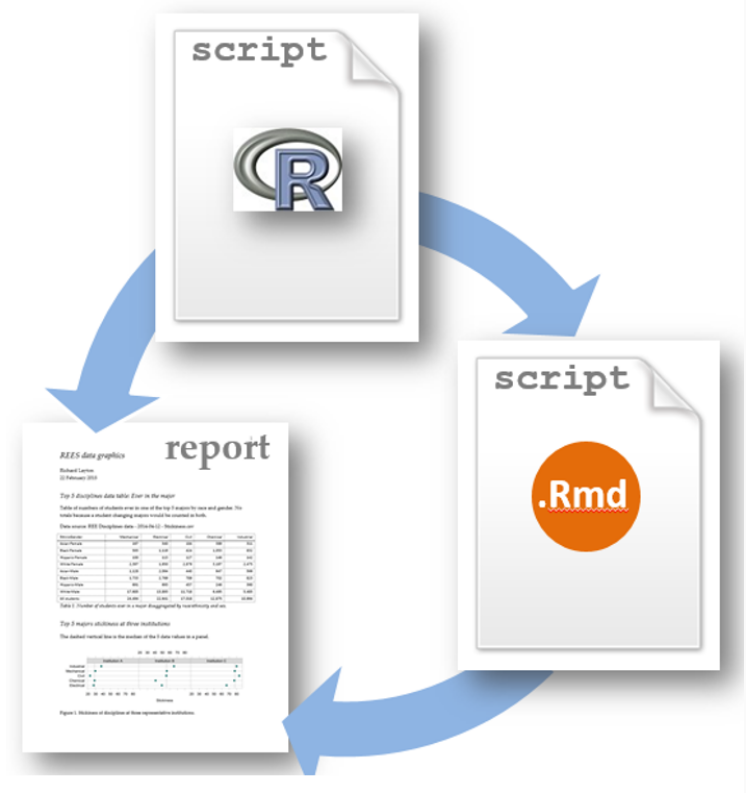
```
carpentry/ d7_01-extract-financials.R
           d7_02-extract-mortality.R
           d7_03-tidy-inequality-data.R
```

Add a number `01`, `02`, etc., when content is saved in different forms

```
data/      d7_01-survey-data.csv
           d7_02-survey-data-wide.csv
```

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/d7_extract-and-tidy.R`

In this R script, read the raw data

```
read_csv("data-raw/d7_survey-data-raw.csv")
```

prepare it for graphing and write the dataframe

```
write_csv(dataframe, "data/d7_survey-data.csv")
```

to the data directory

`data/d7_survey-data.csv`

Use your file-naming scheme consistently

Write an R script for data tidying

`carpentry/d7_extract-and-tidy.R`

In this R script, read the raw data

```
read_csv("data-raw/d7_survey-data-raw.csv")
```

prepare it for graphing and write the dataframe

```
write_csv(dataframe, "data/d7_survey-data.csv")
```

to the data directory

`data/d7_survey-data.csv`

Relative file paths document the graph design workflow

Write an R script for graph design

`design/d7_div-stack-bar.R`

In this R script, read the tidy data

```
read_csv("data/d7_survey-data.csv")
```

create the graph and write the image

```
ggsave("figures/d7_div-stack-bar.png")
```

to the figures directory

`figures/d7_div-stack-bar.png`

Again, note the file-naming scheme

Write an R script for graph design

`design/d7_div-stack-bar.R`

In this R script, read the tidy data

```
read_csv("data/d7_survey-data.csv")
```

create the graph and write the image

```
ggsave("figures/d7_div-stack-bar.png")
```

to the figures directory

`figures/d7_div-stack-bar.png`

The Rmd report runs all the required files in order

Write an Rmd report

`reports/d7_report.Rmd`

containing the report text interleaved with code chunks that

run every R script for this display

```
source("carpentry/d7_extract-and-tidy.R")
source("design/d7_div-stack-bar.R")
```

import data to print a data table

```
read_csv("data/d7_survey-data.csv")
```

and import the figures

```
include_graphics("figures/d7_div-stack-bar.png")
```

Again, note the file-naming scheme

Write an Rmd report

`reports/d7_report.Rmd`

containing the report text interleaved with code chunks that

run every R script for this display

```
source("carpentry/d7_extract-and-tidy.R")
source("design/d7_div-stack-bar.R")
```

import data to print a data table

```
read_csv("data/d7_survey-data.csv")
```

and import the figures

```
include_graphics("figures/d7_div-stack-bar.png")
```

Effective file management summary

Plan your file and directory structure at the start of a project

Adopt a scheme to consistently name your files

Explicitly link files using relative file paths

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) `https://doi.org/10.1371/journal.pcbi.1005510`