Basic elements of file management

ME 447/547 Visualizing Data

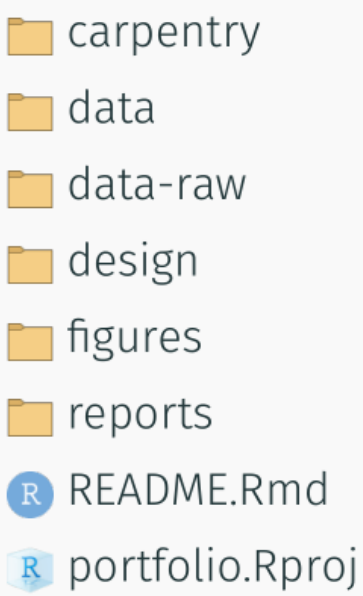
Richard Layton

December 2018

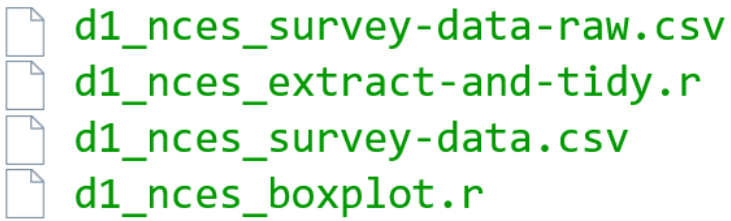
Rose-Hulman Institute of Technology

Effective file management starts at the beginning of a project

Planning the structure



Naming files



Linking files explicitly

```
source("carpentry/d1_nces_extract-and-tidy.r")
source("design/d1_nces_boxplot.r")
source("data/d1_nces_survey-data.csv")
include_graphics("figures/d1_nces_boxplot.png")
```

Given the project directory 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

Project directory 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	◁ Reports explicitly call on resource files
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

Three basic principles should guide your choice of filenames

Filenames should be machine readable

- avoid spaces, use delimiters “_” and “-” deliberately
- avoid punctuation, symbols, 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

  e.g., d1, d2, ... or yyyy-mm-dd
- use leading zeros

  e.g., 01, 02, ..., 99 or 001, 002, ..., 999

A sample set of portfolio file names illustrates the principles

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

Hypenated 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

For example, suppose the data tidying requires 3 files, run in order,

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

Or if the same content is rearranged

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

Creating explicit links

Workflow begins by acquiring and saving the raw data

Raw data are never edited manually

`data-raw/d7_data-raw.csv`

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_data-raw.csv)`

prepare it for graphing and write the dataframe

`write_csv(data/d7_data.csv)`

to the data directory

`data/d7_data.csv`

Plan a file-naming scheme

Write an R script for data tidying

`carpentry/d1_01_data-carpentry.R`

In this R script, read the raw data

`read_csv(data-raw/d1_01_raw-data.csv)`

prepare it for graphing and write the dataframe

`write_csv(data/d1_01_tidy-data.csv)`

to the data directory

`data/d1_01_tidy-data.csv`

Relative file paths document the graph design workflow

Write an R script for graph design

`design/d1_01_graph.R`

In this R script, read the tidy data

`read_csv(data/d1_01_tidy-data.csv)`

create the graph and write the image

`ggsave(figures/d1_01_graph.png)`

to the figures directory

`figures/d1_01_graph.png`

Use the file-naming scheme consistently

Write an R script for graph design

`design/d1_01_graph.R`

In this R script, read the tidy data

`read_csv(data/d1_01_tidy-data.csv)`

create the graph and write the image

`ggsave(figures/d1_01_graph.png)`

to the figures directory

`figures/d1_01_graph.png`

The Rmd report runs all the required files in order

Write an Rmd report

`reports/d1.Rmd`

containing the report text interleaved with code chunks that

run every R script for this display

```
source(carpentry/d1_01_data-carpentry.R)
source(design/d1_01_graph.R)
```

import data to print a data table

```
read_csv(data/d1_01_tidy-data.csv)
```

and import the figures

```
include_graphics(figures/d1_01_graph.png)
```

Again, use the file-naming scheme consistently

Write an Rmd report

- `reports/d1.Rmd`

containing the report text interleaved with code chunks that

run every R script for this display

- `source(carpentry/d1_01_data-carpentry.R)`
- `source(design/d1_01_graph.R)`

import data to print a data table

- `read_csv(data/d1_01_tidy-data.csv)`

and import the figures

- `include_graphics(figures/d1_01_graph.png)`

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`