File management and projects in R

or, How to keep your computer safe from fire

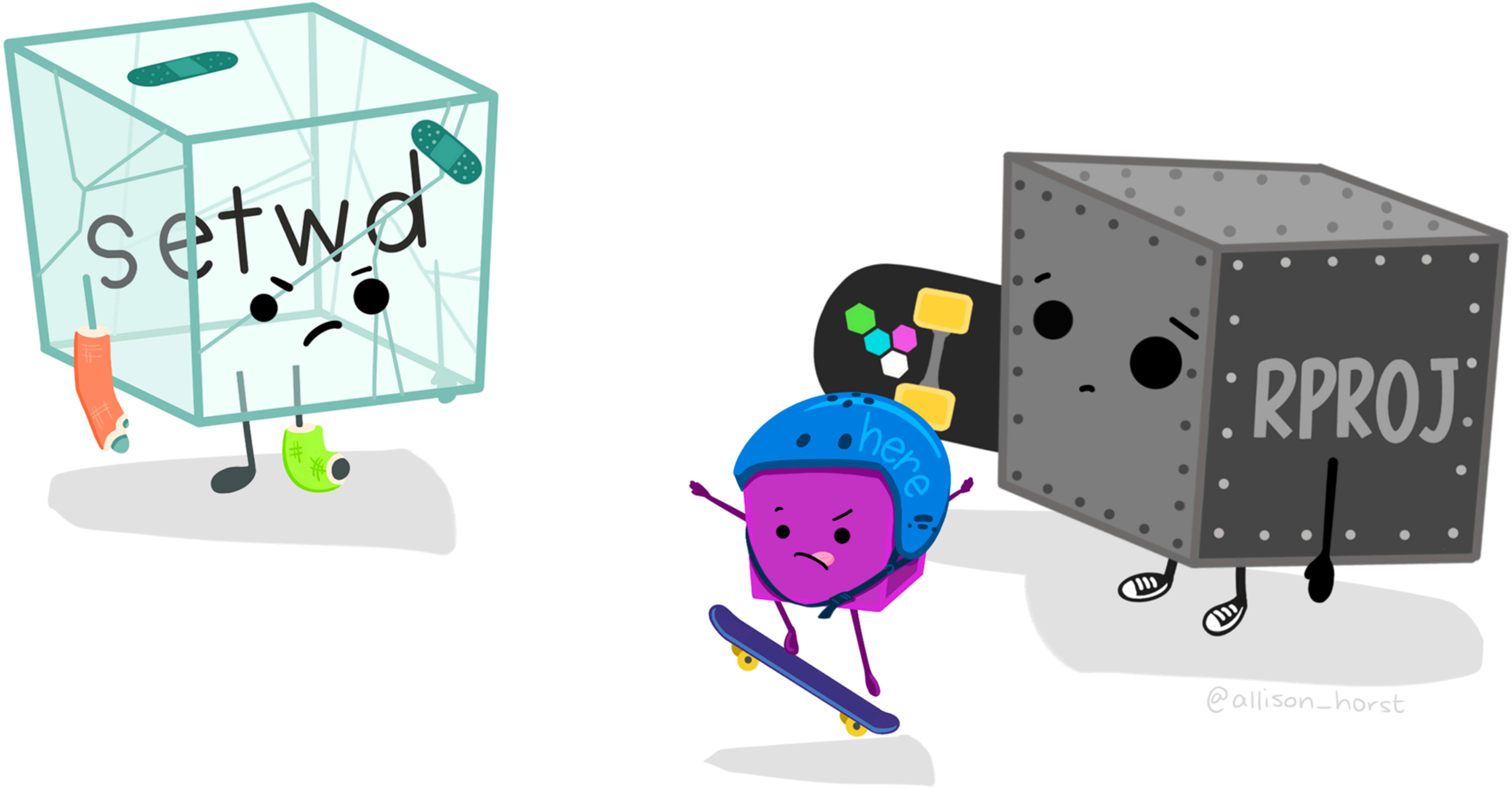
There's a famous blog post about workflows in R[1] about a talk Jenny Bryan gave that included this slide:

> If the first line of your R script is
>
> ```
> setwd("C:\Users\jenny\path\that\only\I\have")
> ```
>
> I will come into your office and SET YOUR COMPUTER ON FIRE 🔥.
>
> If the first line of your R script is
>
> ```
> rm(list = ls())
> ```
>
> I will come into your office and SET YOUR COMPUTER ON FIRE 🔥.

Instead: project-oriented workflow

- R projects provide a structured and organized way to work on projects in R
- R projects encapsulate all project-related files and settings into a single directory
- RStudio makes it easy to work with R projects

R Projects (and related tools) can prevent a lot of accidents!


setwd
here
RPROJ
@allison_horst

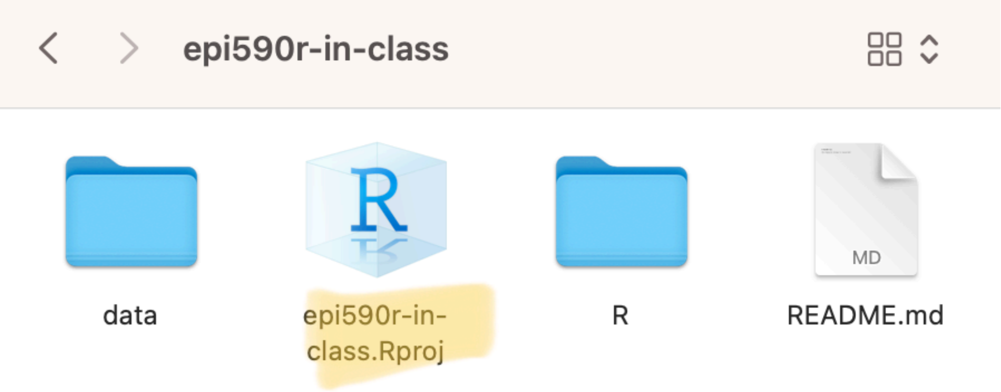
R Projects

Benefits of R Projects

1. **Isolation**: Each project has its own workspace, separate from other projects
2. **Reproducibility**: Projects ensure that code and data are self-contained and portable
3. **Collaboration**: Projects facilitate collaboration by sharing the entire project directory

Always open a project by opening the `.Rproj` file

Mac | Windows



You can have multiple projects open at once in different RStudio sessions!

You can also switch between R projects from RStudio

Creating an R Project

1. Open RStudio and go to **File > New Project**, or click on the projects button in the upper-right corner of RStudio.
2. Choose a project location (New Directory, Version Control, Existing Directory).
3. Specify the project directory (where on your computer you are storing the folder with the project) and create the project.
4. Choose the project type (e.g., regular project, R package, Shiny app, Quarto website, Bookdown book)

You already have an R project!

In the exercises, we are going to make some more changes to the repo you *forked* and *cloned*

1. Download an `.R` script and a `.csv` file from the website

- We'll be using some data from the 1979 National Longitudinal Survey of Youth

2. Find your `epi590r-in-class` repo in your file browser

- Create an `R` folder and a `data` folder
- Within the `data` folder add a `raw` and a `clean` folder.
- Put the `.csv` file in the `data/raw` folder and the script in `R` folder.

File structure goal

```
epi590r-in-class/
├─ epi590r-in-class.Rproj
├─ README.md
├─ R/
│   └─ clean-data-bad.R
├─ data/
│   ├─ raw/
│   │   └─ nlsy.csv
│   └─ clean/
```

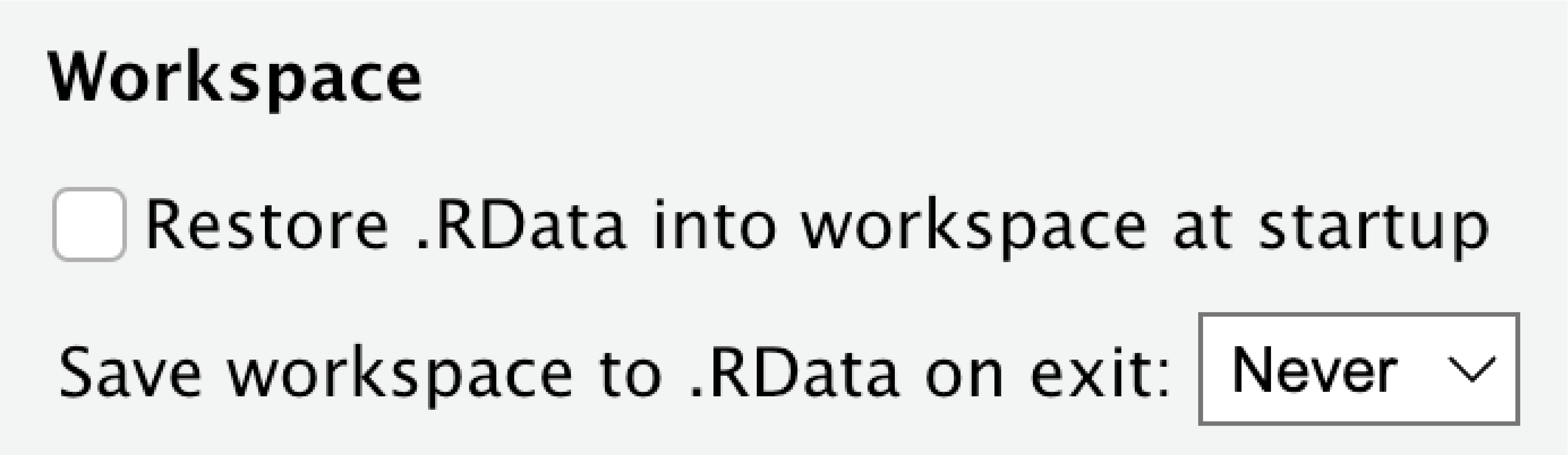
Exercises, cont.

3. Return to RStudio. If you closed RStudio, make sure you re-open this project. Look to the filepane to confirm the files are there.
4. Stage, commit, and push the changes you've made.
5. Try to run the code, line-by-line, in `clean-data-bad.R`.

- As you're running it, try to think of changes you might make

Stop for a settings change!

6. Tell RStudio to start fresh whenever you start a new session



Workspace

□ Restore .RData into workspace at startup

Save workspace to .RData on exit: Never

7. Close RStudio, then open it up again by opening the `epi590r-in-class.Rproj` file in your file browser

Exercise: work with files in your R project

15:00