Project-oriented workflow

Tidyverse Packages Blog

> *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 🔥.*

–Jenny Bryan

Workflow versus product

Definitions

- Workflow: personal taste and habits.
- Product: essence of your project.

Don’t hardwire your workflow into your product.

Which is workflow or product?

1. The editor you use to write your R code.
2. The raw data.
3. The name of your home directory.
4. The R code someone needs to run on your raw data to get your results, including the explicit library() calls to load necessary packages.

Example: Remove workflow

The name of the home directory is workflow, not product.

```
home <- "C:/Users/Mauro/Documents/"  # Workflow
proj_path <- "path/to/project"
paste0(home, proj_path)
#> [1] "C:/Users/Mauro/Documents/path/to/project"
```

Better

```
proj_path <- "path/to/project"
fs::path_home_r(proj_path)
#> C:/Users/Mauro/Documents/path/to/project
```

Best

```
fs::path_home_r("path", "to", "project")
#> C:/Users/Mauro/Documents/path/to/project
```

Self-contained projects

Self-contained projects can be moved around on your computer or onto other computers and will still “just work”.

It’s like agreeing that we will all drive on the left or the right. A hallmark of civilization is following conventions that constrain your behavior a little, in the name of public safety.

–Jenny Bryan

What do they look like?

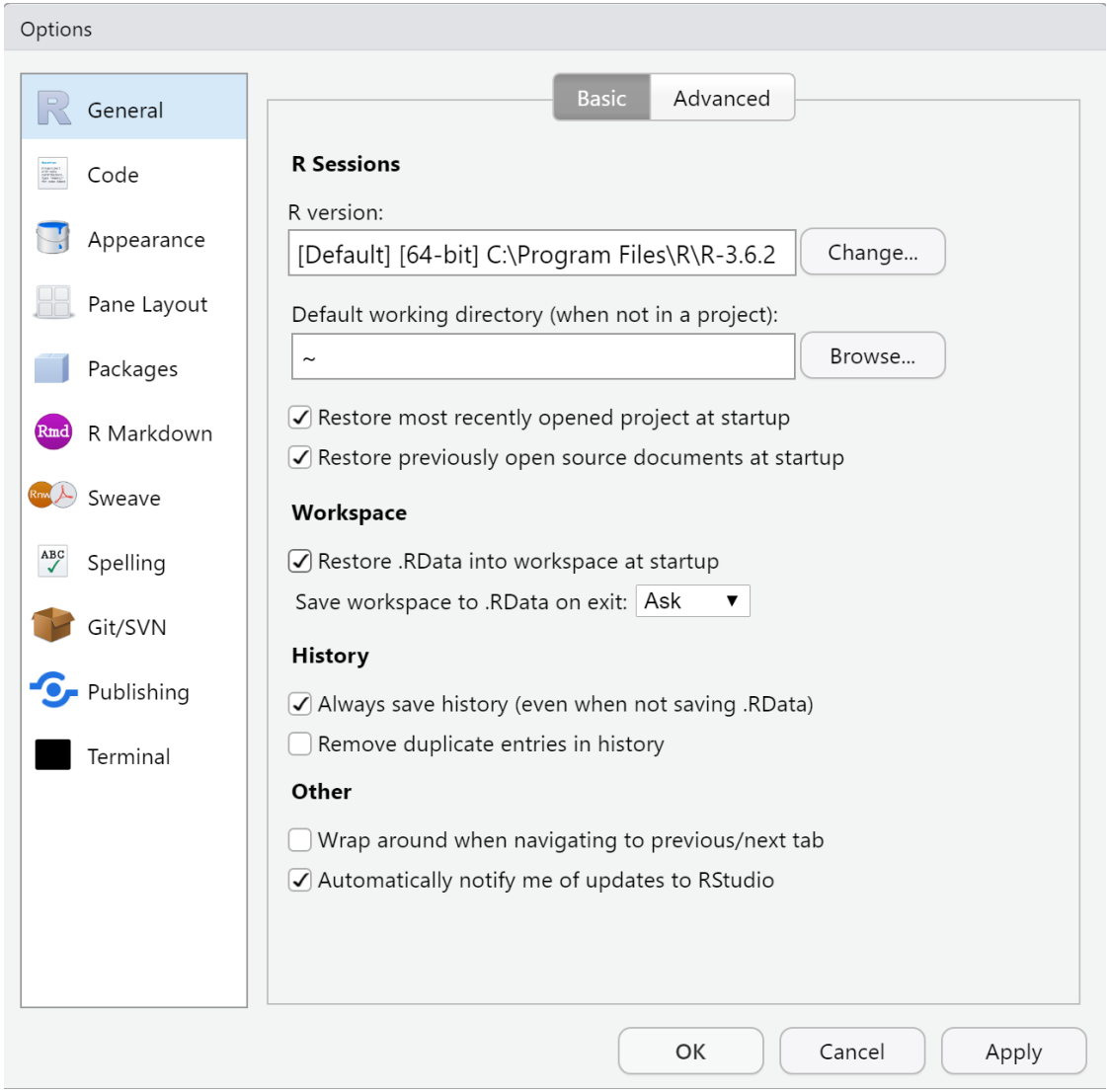
1. The Project folder contains all relevant files.
2. Any .R can run from a fresh R process with wd set to root.
3. Any .R creates all it needs, in its own workspace or folder
4. Any .R touches nothing it didn’t create (e.g. doesn’t install).

Violations …

What should you do instead?

```
path_to_data <- "../datasets/my-data.csv"
```

What should you do instead?



What should you do instead?

```
pacman::p_load(random)
```

setwd( )

What's wrong?

```
library(ggplot2)
setwd("/Users/jenny/cuddly_broccoli/verbose_funicular/foofy/data")
df <- read.delim("raw_foofy_data.csv")
p <- ggplot(df, aes(x, y)) + geom_point()
ggsave("../figs/foofy_scatterplot.png")
```

What's wrong?

- Paths work for nobody besides the author.
- Project not self-contained and portable.
- To run, it first needs to be hand edited.
- Suggests that the useR does all of their work in one R process:
  - Unpleasant to work on more than one project at a time
  - Easy for work done on one project to accidentally leak into another (e.g., objects, loaded packages, session options).

What should you do instead?

- Use RStudio projects, and/or
- Use the here package (works well with .Rmd files)

```
library(ggplot2)
library(here)

df <- read.delim(here("data", "raw_foofy_data.csv"))
p <- ggplot(df, aes(x, y)) + geom_point()
ggsave(here("figs", "foofy_scatterplot.png"))
```

```
rm(list = ls( ))
```

What's wrong?

- Suggests the useR works in one long-running (not fresh) R process.
- Does NOT, in fact, create a fresh R process – it only deletes objects from the global workspace but leaves stuff that make your script vulnerable to hidden dependencies (e.g. packages, options, working directory).
- Is hostile to anyone that you ask to help you with your R problems.

What's better?

- Start from blank slate.
- Restart R very often.
- Re-run your under-development script from the top. For long running processes:
  - Isolate slow bit in its own script; write it with `saveRDS()` and read it with `readRDS()`, or
  - Use drake.

Discuss: Must have or nice to have?

The importance of these practices has a lot to do with whether your code will be run by other people, on other machines, and in the future. If your current practices serve your purposes, then go forth and be happy

– Jenny Bryan

Learn more

- What They Forgot to Teach You About R (Jenny Bryan & Jim Hester).