Eltecon Data Science Course by Emarsys

Introduction

Gábor Kocsis

September 9, 2020

R and RStudio

R and RStudio

- R is a programming language
- RStudio is an IDE for using R
    - IDE: Integrated Development Environment
    - Code editor
    - Built-in automation tools
- R may be used without RStudio, but RStudio cannot be used without R

RStudio projects

- RStudio projects help to organize your work
- Each project has its own working directory and workspace
- You can work in multiple projects at the same time

Relative paths

- Check your working directory with `getwd()`, it might return `"/Users/your_name"`
- If you need, set your working directory with `setwd()`
  - e.g. `setwd("~/eltecon-ds/first_class")`
- When you refer to a file in the working directory, don't need to specify the path
  - `foo <- fread("sales.csv")`
- but, when you refer to a file in another directory, you need to define the path relative to the working directory
  - `foo <- fread("data/sales.csv")`
  - `foo <- fread("../sales.csv")`

Annoying situations

Code calls files that are not available anymore

```
Error in fread("data/raw_data.csv"): File
'data/raw_data.csv' does not exist or is non-readable.
```

Don’t understand what is happening in the code you wrote last year

```
dt <- fread("data/data.csv")
boo <- foo(dt)
out1 <- doCalculations(boo)
out2 <- doMoreCalculations(boo)
plotFigures(boo)
plotMoreDetailedFigures(boo)
```

Have to change one of your assuptions in your model that was copy pasted throughout the whole project

Code and data - Guidelines

A good directory structure

- project_directory
    - analysis.R
    - functions.R
    - data
    - figures
    - README.md
    - .gitignore

analysis.R

- Save your analysis script here
- Write comments to separate sections, but do not overuse them
- Use intention revealing names
  - e.g. instead of `ps` use `product_supply`
- Make meaningful distinctions
  - e.g. don't use `cust` and `customers` close to each other
- Use pronounceable names
  - e.g. instead of `purchymd` use `purchase_date`
- Use uppercase letters for constants
  - e.g. `PROPORTION_OF_INCOME_SAVED`, `LEVEL_OF_UTILITY`

functions.R

- Save your functions here
- Functions in long scopes should have short evocative names
  - e.g. `replaceNAWithZero()`
- Functions in small scopes should have long and precise names
  - e.g. `plotPriceElasticityOfGasolineDemand()`

Clean coding

"You shouldn't have to read the body of a function to know what it does. It's name should tell you." - Robert Cecil Martin aka Uncle Bob

data folder

- File names should declare their function
- Always keep a version of the original raw data
- Optionally, use sub-folders in the data folder like
  - raw_data
  - processed_data

Summary

Summary

- R and RStudio
  - RStudio projects
  - Relative paths
- Annoying situations
- Code and data - Guidelines
  - A good directory structure
  - analysis.R
  - functions.R

Sources

Sources

- https://www.r-bloggers.com/structuring-r-projects/
- https://support.rstudio.com/hc/en-us/articles/200526207-Using-Projects
- https://web.stanford.edu/~gentzkow/research/CodeAndData.pdf
- http://www.informit.com/articles/article.aspx?p=1323426
- https://dzone.com/articles/naming-conventions-from-uncle-bobs-clean-code-phil