Quality Control

Data Science Tools Workshop

Fabien Forge

22/10/2021

Quality Control

- There are two sorts of issues that can make reproduction fail
- 1. The structure of your work environment makes it very hard to take over
- 2. There are errors in your code

Quality Control, continued

- The data science community has developed some standards and technique to deal with these issues
- 1. Standards on how to build a workspace
- 2. Unit testing to put checks on your code

Workspace

- You probably already use a structure for your projects
  - i.e. hopefully your data is not in download and your scripts on your desktop
- For replication, the organization of these folders matters
- Creating folders on the fly may also not be optimal
- Let's learn how to create a proper workspace suited for data work

Path

- Your computer organizes folders and files in a tree-like fashion
- Think of an upside down tree. The top of the tree is the `root`
- Your file leaves a certain number of branches away from the root
- This is why we talk about a `path`

Path

- Paths are separated by slashes
  - backward slashes in Windows \
  - forward slashes in Unix, Linux /
- After each slash you either find a `directory` name or a file name
  - A directory is the same thing as a folder

Path

Windows:

```
E:\Data\MyStuff #(path terminating in a directory name)
E:\Data\MyStuff\roads.shp  #(path terminating in a file nan
```

Unix - Lixux:

```
~/Dropbox/McGill/Data Science Workshop/Data-Science-Tools-W
/Users/forgef/Dropbox/McGill/Data Science Workshop/Data-Sc
```

Absolute or Full Path

- When you write down a path from the root to the end file we talk about a **full path**
- A full path is the least sharable path that you can use
- You can't pass it to another person easily
    - You can't pass it to your next laptop easily!

Relative path

- A **relative path** refers to a location that is relative to a **current directory**
- Relative paths make use of two special symbols:
  - dot (`.`) - `current directory`
  - double-dot (`..`), `parent directory`
- **Double dots** are used for **moving up** in the hierarchy.
- A **single dot** represents the **current** directory itself.

Current directory

- Everytime you open a script a `default`current directory‘ is associated to it
- By default, the curren directory usually corresponds to where your script is open
- You can ask your statistical software of choice or the shell (command line) to display the current directory

What is my current directoy?

Windows, `cd` for current directory

```
cd
```

Mac, **Stata**, `pwd` for print work directory

```
pwd
```

R, `getwd()` for get work directory

```
getwd()
```

Python `os.getcwd()` or start a shell command using `!`

```
import os
os.getcwd()

!pwd
!cd
```

Setting the work directory

Windows, `cd` for current directory + `<PATH>`

```
cd D:\Root\ParentFolder\FinalFolder
```

Mac, **Stata**, `cd` for change directory

```
cd ~/ParentFolder/FinalFolder
```

R, `setwd()` for set work directory

```
setwd("c:/Documents/my/working/directory") # windows
setwd("/path/to/my/directory") # unix
```

Python `os.getcwd()` or start a shell command using `!`

```
import os
os.chdir()

%cd
```

Navigating though directories - Windows

`Say that you are currently in D:\Data\Shapefiles\Soils`

Create a new folder

mkdir

Cookiecutter

- Cookicutter