CODERS' CORNER

CENTRE FOR THE STUDY OF AFRICAN ECONOMIES

AUTOGENERATE YOUR FOLDER STRUCTURE

For multi-year, multi-analysts/RA/PIs projects, it is very important to keep a clean and organized folder structure. There are a few resources out there to organize and maintain a sound folder structure that will make sense years down the line (see Gentzkow and Shapiro, 2014). However, it can be quite tedious to manually create all these folders. This post is designed as one to copy-paste whenever you start a new project. However, before I get to the code, I start with a quick description of good practices when it comes to maintaining a good structure for your data work.

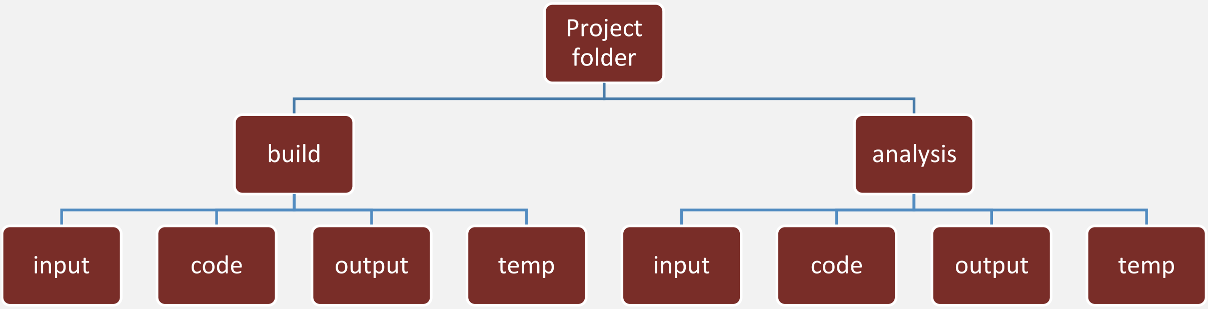
It is good practice to have a master script that call all your relevant scripts. For quality assurance purposes:

1. Keep your codes relatively short (and efficient) to avoid having to spend 10 hours checking for errors and bugs in the code. A study on the Cisco review process, for example, shows that one shouldn't review too much code at once (<500 lines of code).
2. Annotate your code (always).
3. Make data inputs and data outputs clear to identify: i.e. what data are used in your code, and what is generated.

A good folder structure will make #3 very easy, because looking at the path will make it really easy to see which stage the data are in, and #1 and #2 are easily done when you have a master script that tracks what each step is doing.

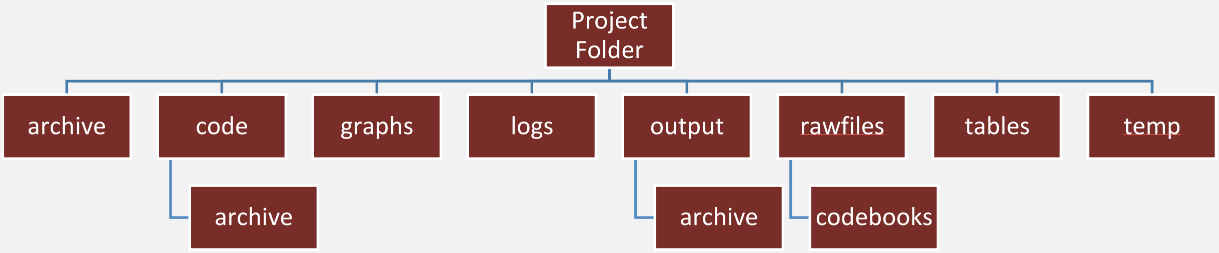
Below are two examples of helpful structures:

Structure from Gentzkow and Shapiro, 2014



This structure has the advantage of separating completely the analysis from the data cleaning.

Other alternative structure



This structure gives more flexibility but requires some discipline on the part of the coder.

> Note: The code folder contains all the scripts that are used in the project. You can then create a master file that maps out all files, and has clear information on the steps that each file is designed to take.

Once you have chosen the structure you prefer, you can then move on to editing the code below to match that preference.

<u>Stata (R version below)</u>

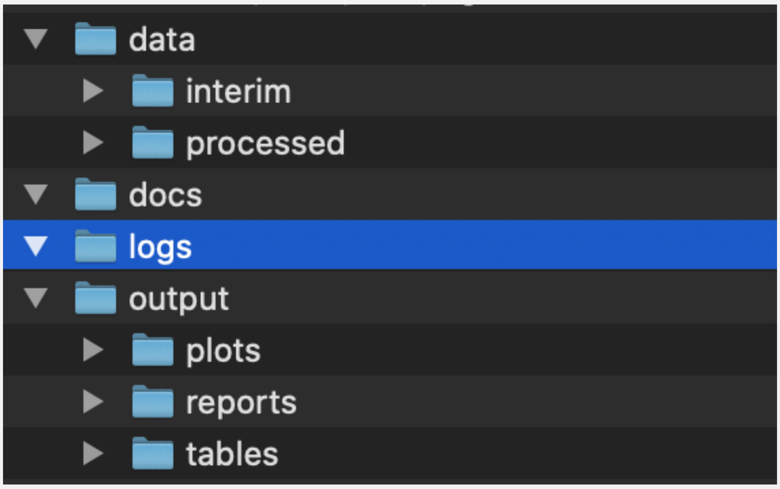
If you want to create your folders in your current working directory use the global below, otherwise, just replace `c(pwd)' with the path you would like to create it in.

Code

```
* Setting up paths
global wd   `c(pwd)'
global data $wd/data
global docs $wd/docs
global logs $wd/logs
global output $wd/output

* Creating folders
! mkdir "$data/"
! mkdir "$data/interim"
! mkdir "$data/processed"
! mkdir "$docs"
! mkdir "$logs"
! mkdir "$output"
! mkdir "$output/plots"
! mkdir "$output/reports"
! mkdir "$output/tables"
```

Data structure generated



```
▼ data
    ▶ interim
    ▶ processed
▼ docs
▼ logs
▼ output
    ▶ plots
    ▶ reports
    ▶ tables
```

The same result could be achieved by doing the following:

```
* Creating folders
foreach newfolder in "$data/" "$data/interim" "$data/processed" "$docs"
"$logs" "$output" "$output/plots" "$output/reports" "$output/tables" {
      ! mkdir `newfolder’
}
```

R

```
# Storing the working directory
location=getwd()

# Creating folders
dir.create(file.path(location,”data”))
dir.create(file.path(location,”data/interim”))
dir.create(file.path(location,”data/processed”))
dir.create(file.path(location,”docs”))
dir.create(file.path(location,”logs”))
dir.create(file.path(location,”output”))
dir.create(file.path(location,”output/plots”))
dir.create(file.path(location,”output/reports”))
dir.create(file.path(location,”output/tables”))
```

Binta Zahra Diop, DPhil Candidate in Economics, Oxford

30 October 2020

References

Gentzkow, M. and Shapiro, J.M., 2014. Code and data for the social sciences: *A practitioner’s guide. Chicago, IL: University of Chicago.*