**Assignment 2 – Empirical workflow**

**Due date: Wednesday, June 10th, 2020 by 5:00pm**

**Student: Danilo Esteban Aristizabal Giraldo**

**Gentzkow and Shapiro**

Read Gentzkow and Shapiro “Code and Data for the Social Sciences” in the “Helpful stuff” Github directory and answer the following

1. Summarize briefly the point of chapters 2- 8 in less than one page.

**Automation**: Automate everything that can be automated. Pushing the boundaries of automation pays big dividends. Using loops and functions save a lot of time in the future.

**Version control**: There are good reasons to store multiple versions of the same file: provide a quick way to roll back changes you want to discard. However, in the future it is a pain and it can be confusing. To avoid this, we can use version control software. It is like an undo command for all steps you are working on.

Their first rule is to keep everything under version control. The second one is to run the entire directory before you check in your changes.

Version control provides a comprehensive solution that guarantees both replicability and undo-ability with minimal effort.

**Directories**: Separate directories by function.

**Keys:** Store cleaned data in tables with unique, non-missing keys. It is important to store row data in normalized files that preserve the information in the original data source. Also, it is important to construct a second set of normalized files that includes the transformations of the original variables that you will need for your analysis. Finally, merge together the tables in the database to form the rectangular array on which you will estimate your model.

**Abstraction:** Abstract only to eliminate redundancy and to improve clarity.

**Documentation:** Avoid comments that potentially in the future may contradict the code. Keep your comments up to date, meaning just as up to date as your code.

**Management:** Manage task with a task management system. There are several options: [www.asana.com](http://www.asana.com), [www.wrike.com](http://www.wrike.com) , [www.getflow.com](http://www.getflow.com)

1. Why do Genztkow and Shapiro think these elements of modern empirical work are so important? What problems does each element solve?

These elements are so important basically because of two reasons: nowadays, making empirical work involves working with teams, and second because to make modern empirical work we need to deal with big data. Having this organization reduces the probability of making mistakes in the research process.

Automation solves the problem of spending too much time making repeated processes when writing our codes.

Version control avoids the problem of having several do files or R scripts with different versions. We can just keep track of the changes that each person, working on a project, has done. So, it provides replicability and undo-ability with minimal effort.

Directories solve the problem of not understanding the files that are in a folder. It is a simple way to know what elements we have in each directory.

Identifying keys help a lot in not having to spend to much time understanding why we have the same observation in different counties, for example. Identifying the keys is crucial to make appropriate merges of datasets that are going to be the input of our research.

Abstraction solves the problem of redundancy and simple typos mistakes.

Documentation help us to have our comments updated.

And finally, management solves the problem of not knowing who is charge of a specific task.

1. Give an example of the sort of problem that could arise in the course of an empirical project if someone were to fail to adopt these principles.

A typical problem that happens when working in an empirical project that doesn’t follow these principles is that when directories are not appropriately defined, we need to spend a lot of time understanding the elements that are in the folder: databases (whether they are original or processed), do files, R scripts. Furthermore, we can make irreparable mistakes if for example we manipulate a database and save it in the same folder with the same name, overwriting the original one.

1. How do you plan to incorporate these solutions into your own work?

To use the version control I plan to create a GitHub account to work on projects with my colleagues to keep track of all changes done during my empirical research. I will also incorporate all useful tips of automation and working with directories well specified in my research projects.

1. **(Section GitHub)** Git is the most popular version control program and it integrates perfectly with online collaboration tools like GitHub and GitLab. GitHub is an online hosting platform that provides different services built on top of the Git system. It is important to know that Git and GitHub are distinct things. We do not need GitHub to use Git, but it will make our lives so much easier.

**Git**

These next questions concern the software “git” and “github”.

1. Create a new section in the document you used to answer questions 1-4. Briefly explain what git and github are used for, how they are similar and how they are different. (Done above!)
2. Name a benefit of using git to organize your empirical research. What types of common problems can occur if you don’t use git?

A benefit of using git is to have control of different versions of our codes when working in empirical research. Typical problems when we don’t use git can be related with losing several weeks of work after we realize that we made a mistake on the code we already save it, or conflicts with the files where your coauthor may be working at the same time.

1. What about using git is challenging for you for right now? What steps can you take to minimize those challenges such that you can adopt git for this class?

I am not very used to work on version control systems like Git, so at the beginning it can be a challenge to me, but I think as Gentzkow and Shapiro say, it pays big dividends and avoid future headaches. I have created my GitHub account, I really enjoyed it! To make it easier the transition on github I plan to work together with another colleagues of the class and other friends that are not taking this course.

1. Name the four main Git operations. What does each operation do and how are is each operation different from one another?

**Commit**: Make comments and tell Git that you are sure some changes should be part of the repository history.

**Stage**: Tell Git that you want to add changes to the repository history.

**Pull**: Get any changes made on the GitHub repository, either by your colleagues or you on another machine.

**Push**: Push any (committed) local changes to the GitHub repository.

1. The first step in your new empirical workflow is the creation of a Github repository (“repo”). You can either do this independently or do this through R functionality. You need to create a github account, then create your first repository called “Titanic”. Initialize with a Readme and create the separate folders that we discussed in class on Monday.
2. Post a link to your repository
3. Please clone our course github repository on your desktop