Assignment 2

1. Summary of Gentzkow and Shapiro:

This guide helps researchers to be more efficient in the process of research creation, especially when working collaboratively with other people. The authors propose 8 steps for doing this:

1. Automate as much of the process of research creation as you can as this will be helpful when trying to replicate your own work and it makes all the steps in the process more efficient as it avoids you having to run a thousand of independent files and avoids confusion. Add shell scripts to your working directory that are like roadmaps for the operating system, thus when you execute such script, all the process from importing raw data to producing graphs and the final version of the paper can be done automatically.
2. Use version control software as this will automatically keep track of all the changes made to any file and about who made the changes. In addition to that, version control allows you to go back to a previous version of the paper or to a previous version of the code in case you want.
3. Construct independent directories by functionality to organize your code, so you can make changes in the data analysis process without having to re-run the code that builds the data. Separating scripts into functional groups makes debugging easier and faster when something goes wrong.
4. Make sure the physical structure of your database communicates its logical structure and includes unique keys that are self-documenting, so when you look at the database you understand which variables are included and what they are.
5. Use abstraction wisely as it is important for writing efficient code because it eliminates redundancy, the possibility of mistakes and makes the code more readable, but do not use abstraction without purpose as it can be a waste time trying to make simple processes that you won’t use again more complex. In this regard, when writing functions it is recommended to ”unit-test” on some fake data before using the function in your real data in case you need to fix mistakes and this also allows you to document how your function works.
6. Only document things that you will keep up to date when changing the code and, if possible, write code that is self-documenting. This will avoid unnecessary confusion when using codes that have comments and notes that are not up to date.
7. Use task-management systems when working collaboratively as this solves communication issues and makes it easier to see who is in charge of what and if it was done or not.
8. These tips help us to be more efficient in the process of research creation, especially when working collaboratively with other people as it allows us to save time when cleaning data or when we need to fix mistakes and make changes in regression specifications, replicate results, and so on. It makes the process of research creation more smooth and easy to track.
9. A coauthor makes a change in a do file that leads to different results, we use those results in the paper we submit to a journal, then 5 months later some referee asks you to run the same specification adding an extra variable but you cannot find the original do file you used and you don’t know exactly which controls you included in said regression or which version of the 10 data files you have you used.
10. I currently use some of these tools as try to keep all my do-files organized, step by step, and use LaTeX for writing academic papers, but I will start creating (learning how to!) a shell script so the whole process can be automatic, use abstraction more efficient as I tend to repeat many times the same lines of code and I recently started to migrate to Slack instead of email+whatsapp to keep track of all the comments we make with my coauthors when discussing about our work. I will also incorportate a task-management system, as I use a very updated version of it: hand notes that I then translate to an email.
11. Git is a version control software that allows you to keep track of changes made to an array of files such as text files, code, etc. Github is an online hosting platform that makes it easier to use the Git system on top of offering other functionalities. You don’t need GitHub in order to use the Git system.
12. The Git system helps you for version control. Thus, when you or a coauthor make changes to some of the files used in the process of producing a paper, then it is easy to see which changes were made, who made them and when, and in case you want to go back to an older version of the files, you can do it. It is like the undo button in Office, but for entire files.
13. I have never used Git before. I use Dropbox and I’ve experienced the hassle of having to constantly communicate with coauthors to see what changes they did, and which one is the latest version of the do-file we’re using, so Git sounds like a great solution for that. Based on what I’ve seen so far in this class, GitHub seems a bit confusing at first and I don’t feel comfortable yet, but with a little practice I can become used to it. Actually, pushing myself to use new tools that increase my productivity and becoming more efficient at coding are two of my goals for this year and that is why I enrolled in this class.
14. Stage: Tells Git that you want to add changes to the repository history.

Commit: Tells Git that you are sure about these changes. It is like double-checking what you did in the Stage step.

Pull: Get any new changes made by you or coauthors on the GitHub repository.

Push: Push any local changes we did to the GitHub repo. This allows other people to know that they have to pull our changes.

My GitHub repository: <https://github.com/aoteroco/titanic>