Assignment 2

Jorge Daniel Guevara Acevedo

Wednesday, June 10^{th}, 2020

1 Code & Data

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

   Automation: there is always good to have a path that could make both; execute all the processes automatically and be reference of the order of procedures. This prevent the tedious, dangerous and inefficient task of looking backwards and try to remember which were step were first in cleaning the data, of were provided the results contained a regression table, and so on.
   Version Control: trying to make version control manually may be useful because it seems like the changes are recorded well, but could be really confusing sometimes. To avoid complications of when save the files as new versions, when not, which date or initials to assign or whatever; is better to use a software. The computer is able to show all the versions of any file, when it was changed, how and even which lines of code were edited.
   Directories: sometimes is easier to make modifications by specifying different directories for different parts of the process, all directed by a superior one. Also is easier to have one for inputs and other for outputs, but is important that their interaction is appropriate in order of avoiding problems to run the process.
   Keys: the data must be organized in way that logical structure of the data base match with it's physical structure. That is why we should be caution about the way data is presented and be sure of the real information we have. The relational bases are those that accomplish logical and physical characteristics. The keys help a lot in this matter, because it identifies the elements of a table and even may relate it to another, so it's important to using it logically for merging or cleaning data.
   Abstraction: sometimes processes doesn't execute for once and only time and for one variable, and is dangerous to copy-paste and then replace the values of the code because increases the probability of do it wrong. That is why is valuable to make tasks that could replicate again for another variable, it saves line codes, prevent errors and is more easy to understand.
   Documentation: is always good to have a reference while you are running a process, some comments along the do or script are welcome. But it represents a problem when the comments are very specific or precede a dynamic process that change values for different executions, this could lead to a mistake. That is the reason why is important to find ways of comment that preserves generality and be helpful in giving instructions at the same time.
   Management: when working in team, is very important to have all the members in the same page. That is way becomes necessary to have a software that let the team communicate and assign roles and tasks more easily.

2. **Why do Genztkow and Shapiro think these elements of modern empirical work**

are so important? What problems does each element solve?

Automation: it is important because prevents not-running the process in the way it was predefined, so also prevents ruining the data and save time in trying of remember how to run the project or how the results were achieved.
Version Control: it is important because provides traceability of all the changes made to the project. However, is better do it with help of a software designed for it, because it prevents confusion of which version I work on, when edit, so on.
Directories: it is important because sub-directories allow us to make modifications for certain phases of the process and organizes the outputs and inputs. It solves the problem of a big process to run, and make easier to edit.
Keys: it is important because preserves the logical structure of data and prevents problems of interpretation and missing values. It solves the problem of non-logical data and keep their characteristics to establish relations between data tables.
Abstraction: it is important to minimize errors that could occur because of the copy-paste problem. Abstraction consist in make general procedures for tasks that must be executed more than once.
Documentation: it is important to giving a good reference while working in coding, there is necessary to give general and opportune information about what are we doing. The problem, this method of documentation solves is the confusion that sometimes could arise from over-specific comments.
Management: it is important to have a good way to communicate and assign roles to the team you are working with. This solves problems of communication and misunderstanding on what tasks is assigned to who.

3. **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.**

   The sort of problems that would be facing if dont follow this principles could be any of the ones that provides the paper; could be impossible to replicate results found time ago, confusion in editing the versions, don't knowing where a directory is, loosing data or results, lack of communication; but the most important one in my opinion is **time.**

   This problems should be measured in time wasted because of the lack of interest in save time in the future.

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

   I think that I have to begin with directories, abstraction, keys and automation, because is the most simple and elemental things I can do in my usual coding and research. Also, I would include the organization by the folders that we learned in class.

   By incorporating this tools, I expect that my time will be save and many phases of the process will improve. For incorporate the other ones like version control and management I think there is some research ahead, because I have to be sure about the tools and software that better fit to my needs.

2 Git

Create a new section in the document you used to answer questions 1-4.

1. **Briefly explain what git and github are used for, how they are similar and how they are different.**

   The main difference between them is that Git is a software installed in a local computer in order of control versions of proyects by this system. In the other hand, GitHub allow to create repositories in the cloud, and let others edit and collaborate.

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?**

   I would say that the main advantage is to have control of all the material and the progress of the project. If I don't used, I would experiment the problems in chapter **Version Control** of Gentzkow and Shapiro's paper, that means ignore when the changes were made, when to edit or create a new file and where the changes between versions are.

3. **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?**

   Is very challenging now because I ignore almost all tools Git and Github offers to me. If I start using it for the class or another projects, I will study and use all the info available on the internet for take a very good advantage of it.

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

   - **Stage:** allows to edit the repository history.
   - **Commit:** save the edition made and add it to history.
   - **Pull:** update the upgrades made in a repository.
   - **Push:** send local changes to GitHub's repository.

   The differences between Stage and Commit is that one starts the edit order and the other one save it. Between Pull and Push, is about the direction they make the updates.