Recordkeeping for Research Projects

Research projects (even for a class) will involve many files that will be created over the course of the project. It is not uncommon for a project to be left when a paper is under submission, and then must be revised months later. Must humans will have forgotten some important things. In this this document I hope to provide hints on how to apply techniques to document (and you're your work self-documenting).

First, I would suggest creating a directory where you will store all of your work for the project. Give the directory a short name that will indicate the topic of the project. (I do not recommend having spaces in the names of files or directories. Some programs don't like this, and very long names can create difficulties.) So, an example for a project that will create a paper on immigrant women might be named "immwom". Within that main directory you will store all of your files for the project. It is helpful to create a files in this directory named "anote.txt" that briefly describe the project. You can update this with notes to yourself or team members with "to do" lists, etc.

If you are doing data analysis, it is VERY IMPORTANT to use syntax (program) files to clean your data, create new variables, merge data, etc. It is also important to NOT OVERWRITE original data. Within program files, you should:

GET OLD DATA FILE
Syntax to Clean data
Syntax to Create variables
SAVE NEW DATA FILE

It is easy to make mistakes. If you overwrite your original data file you may lose the opportunity to correct a mistake. If you use syntax it will be easier to find your errors and correct them.

By using syntax to create new files you will also have a complete record of how a variable was created. Your project may last longer than your memory. And, if other people are working with you, it will be easier for all of you to figure out what was done.

Take care in deciding what you name a file. These are conventions that I recommend.

Depending on the statistical package that you use, the characters after period should be as follows.

Program.do (for Stata)
Datafile.dta (for Stata)

Program.sps (for SPSS)
Datafile.sav (for SPSS)

In each of the above, the extension for the program (or syntax file) is listed first, and the extension for the data file associated with the software package is next. It is possible to name program and data files other things, but using these conventions will help you identify what kind of file you are looking at.

If I am creating a data file in a program, I like to have both the program and the data file have the same "first" name. For example, if I am going to create "immwom.dta" in Stata, I name the program that creates it "immwom.do". This way, I can always go back to the program to figure out how I created a variable, or what else I did to create the .dta file.

When writing programs, it will be helpful to you if you take the time to document your programs and files. You can create variable and value labels, and you can create comment lines within your program.

You can also create titles that will show on your output. When doing this, it can be helpful to you to have the full path of the program that created your output in the title. For example, the full path of the document I am working on now is "H:\HonorsClass\MyWork\record.docx". If I print this and include this information, it will be easier for me to find it if I want to revise it.

Whenever you create a table or figure, it will be helpful to include information about the names of programs that created the output for the table or figure. Of course, you can't do this for your final paper, but you can for drafts.

When you are ready to turn your work in, it is a good idea to do some additional documentation. You might want to create a program for each table in your paper. The file "table1.do" would produce all of the output that went into table 1 (and so on). When it is time to revise, you will be happy that you did this.

In any session, you may create programs that test this or that. It can become very confusing to get too many programs. You might find it helpful to create a directory called "temp" where you store these files that are not important in the long run, but that you don't want to delete right now. Any program that you keep run in your main directories should run start to finish. Each program should begin by calling in the data that you want to use, and then doing the work that you want. Some people will create fragments of programs, but in six months, they may not remember what data the program file used.