1. **Understand folder structure. Perform operations on files in different folders.**

After opening the Visual Studio (an interface to Gitbash), we navigate the existing path via “pwd” and change the directory into a desired location via cd “…”. To see the files within the selected folder, we can either use “ls” or “ls -l”. The former just shows the folder items in sequential order, while the latter shows them in more detail, e.g., when the file was last updated, who the owner of the file is (as shown below).

A picture containing logo

Description automatically generated

Text

Description automatically generated with medium confidence

To open a subfolder within the selected, we simply use “cd” followed by the subfolder name. In our case, we will open “data subfolder” as shown below.

Graphical user interface

Description automatically generated with medium confidence

1. **Automate repeating tasks using Python “for” loops.**

We can automate repeated tasks, such as listing number in the range between 0 and 100. The python will list the numbers as follows. Here, one important part of the coding is, we need to make sure to indent the command “print”.

Text

Description automatically generated with medium confidence

1. **Break up work into smaller components using Python functions.**

Breaking up the work into smaller components (e.g., breaking down the codes into functions) is helpful to keep our codes in order and eases the navigation process for the user. As simple example of “date” function is shown as follows:

Text

Description automatically generated

1. **Use Python “lists” and “dictionaries” appropriately. Demonstrate one of the two.**

One of the advantages of using list is to store as many values as needed in the list, so it saves our time when doing various calculations using these values. An example is shown below:

Text

Description automatically generated

1. **Automate repeating tasks using Stata “for” loops.**

The logic is similar to what we did in Python: we are automating repeated tasks, especially when there are hundreds of entries in the given data. The example below illustrates how we can generate “race” dummy variables and generating mean values of job interview phone calls that they have received based on their race.

Text, letter

Description automatically generated

1. **Break up work into smaller components using Stata .do files.**

The logic is also similar to the case in python. A simple example below is showing the processes of creating “mean” of years of experience and generating a new variable EXP by subtracting mean from the actual years of experience and creating a new variable by squaring the EXP variable.

Text

Description automatically generated

1. **Read .csv data in in Stata.**

To read csv file in Stata, we can use the following code.

Text

Description automatically generated with medium confidence

1. **Fix common data quality errors in Stata (for example, string vs number, missing value).**

To fix a missing value issue, we will replace the empty cells of the variable with zeros. Otherwise, when we e.g., create a dummy variable from a variable with missing values, STATA will generate “1” as opposed to zero, assuming the missing values have infinite values. The example below is showing this process, we are replacing missing values for the variable “fed” (the employer is federal contractor) with zeros.



1. **Aggregate, reshape, and combine data for analysis in Python or Stata. Demonstrate at least one of these data manipulations.**

In STATA, we can show aggregate statistics such as sum and mean using “collapse” command. For example, if we want to see the mean of high- and low-quality resumes by job applicants’ race, we can run the following code and get the results as follows:



Graphical user interface

Description automatically generated with medium confidence

1. **Prepare a sample for analysis by filtering observations and variables and creating transformations of variables. Demonstrate all three.**

To prepare a sample for analysis using the nls data (national longitudinal survey data), we should start preparing the seed for random numbers e.g., 123. This will allow us to reproduce a particular sequence of 'random' numbers. If we have a big data set with many variables, we can filter by dropping the variables that are not going to be used for the analysis and just keep the ones that will be needed. These steps are shown below.

Graphical user interface, text, application

Description automatically generated

From the chosen variables above, if we want to for example transform momed (mom’s years of education) variable by taking the log of it.

A screenshot of a computer

Description automatically generated with low confidence

1. **Save data in Stata.**

To save the filtered and transformed data that we generated in question 10 by replacing the old file, we will use the “save” command as follows.

A screenshot of a computer

Description automatically generated with low confidence

1. **Run ordinary least squares regression in Stata.**

If we want to know the effect of age, race, parents’ education, and residential location on salary, we can run the following OLS regression using the relevant variables. The result is shown as follows:

Table

Description automatically generated

1. **Create a graph (of any type) in Stata.**

We can generate a scatter plot using “twoway” scatter plot command. This illustrates the effect of e.g., education on wage. The result is shown as follows:

A picture containing text

Description automatically generated



1. **Save regression tables and graphs as files. Demonstrate both.**

The following outreg2 command saves the regression table into an excel file.

**Logo

Description automatically generated with medium confidence**

To export a graph from STATA and save it as PDF file, we will use the “graph export” command and get the following result:

**A screenshot of a computer

Description automatically generated with low confidence**

1. **Install a Stata package. (Can be the same as we already did in class.)**

To install a package that allows the user to output the analysis result in different file formats (e.g., MS-Word or MS-Excel), we can use the “ssc install” command as follows:

Text

Description automatically generated