

Exercise Instructions

In this exercise, you will use data from a survey conducted in many countries to investigate whether there is an association between people's overall satisfaction with their lives and two macroeconomic variables: per capita income and the unemployment rate in their countries.

Overview

The data Pew Research Center survey of individuals in xxx countries, conducted in 2002—including a question asking respondents to rate their overall satisfaction with their lives on a scale of 0-10—variable===swb. WDI data on income per capita of different countries in 2002.

What you are going to do in this exercise

STEP BY STEP INSTRUCTIONS

I. Build a set of folders to keep your work in.

→ Make a new folder on your computer, and give it the name

YourNameSWBExercise/, but replace "YourName" with your first and last name.

Tip: You may keep this folder anywhere on your computer you like, but be sure to remember where you put it so that you can find it later.

Tip: Do not include any spaces in the name you give this folder. In fact, you should you should not include spaces in the names of any folders you create for this exercise.

→ In your **YourNameSWBExercise/**, make another new folder called **ProjectFolder/**, and in the **ProjectFolder/** build a hierarchy of subfolders as illustrated below:

```

ProjectFolder/
|
|---Data/
|   |
|   |---AnalysisData/
|   |
|   |---InputData/
|   |
|   |---IntermediateData/
|
|---Output/
|   |
|   |---Figures/
|   |
|   |---Tables/
|
|---Scripts/
|   |
|   |---AnalysisScripts/
|   |
|   |---ProcessingScripts/

```

As you assemble and create the files you use for this exercise, you will store them in the appropriate folders in this hierarchy.

II. Put the input data files in the InputData/ folder.

→ Put copies of the data files, *pew_input.csv* and *wdi_input.csv*, in your **Data/InputData/** folder.

XX. Launch Stata and set the working directory.

→ Launch Stata.

→ Manually check which folder is currently designated as Stata's working directory.

→ If the **ProjectFolder/** is designated as the working directory, do not change it.

→ If any other folder is designated as the working directory, manually change the working directory to the **ProjectFolder/**.

Tip: After you have determined that the **ProjectFolder/** is designated as Stata's working directory, do not change it. For this entire exercise, whenever you are running Stata, the working directory should be set to the **ProjectFolder/**.

XX. Create and save a do-file called *import_pew.do*.

→ Open a new do-file.

→ Give this new do-file the name *import_pew.do*, and save it in your **Scripts/ProcessingScripts/** folder.

XX. Write commands and comments in *import_pew.do*.

→ At the top of the do-file, write a comment indicating that when it is being edited or executed, Stata's working directory should be set to the **ProjectFolder/**.

→ After the comment about setting the working directory, write a command that clears Stata's memory.

→ For this exercise, just the command `clear`, without any arguments or options, is sufficient.

→ Before you type this command, you should type a comment indicating the purpose of the command or what it accomplishes.

Tip: There will be no further reminders about this in these instructions, but you need to remember to write explanatory comments before all the commands you write in all the do-files for this exercise.

→ Write a command that reads the Pew input data from *pew_input.csv*.

→ Use a relative directory path to tell Stata that the file is stored in the **Data/InputData/** folder.

```
import delimited using Data/InputData/pew_input.csv, ///  
varnames(1)
```

→ Write a command that collapses the individual-level data from the Pew input data file into country-level data in which the values of *swb* are equal to the means of the values of *swb* for each country.

```
collapse swb, by(countryname)
```

→ Since the variable *swb* in the collapsed data gives country means of the individual responses in the original data, write a command that changes the name of *swb* to *meanswb*.

```
rename swb meanswb
```

→ Write a command that assigns the label "Mean SWB" to the *meanswb* variable.

```
label variable meanswb "Mean SWB"
```

→ The Pew data is now ready to be merged with the WDI data, so write a command that saves the data in a file named *pew_intermediate.dta*.

→ Use a relative directory path to tell Stata to save this file in the **Data/IntermediateData/** folder.

```
save Data/IntermediateData/pew_intermediate.dta, replace
```

XX. Execute and save *import_pew.do*

→ After you have finished writing the commands described above, execute *pew_import.do* from start to finish. Be sure that it runs without generating an error, and that it generates the *pew_intermediate.dta* data file and saves it in the **Data/IntermediateData/** folder.

→ Save the complete *pew_import.do* script in the **Scripts/ProcessingScripts/** folder (which is where you saved it when you first created it).

Tip: Remember that you save do-files interactively; you do not write a command in the do-file that saves the do-file.

XX. Create and save a do-file called *import_wdi.do*.

→ Open a new do-file.

→ Give this new do-file the name *import_wdi.do*, and save it in your **Scripts/ProcessingScripts/** folder.

XX. Write commands and comments in *import_wdi.do*.

→ At the top of the do-file, write a comment indicating that when it is being edited or executed, Stata's working directory should be set to the **ProjectFolder/**.

→ After the comment about setting the working directory, write a command that clears Stata's memory.

→ Write a command that reads the WDI input data from *wdi_input.csv*.

→ Use a relative directory path to tell Stata that the file is stored in the **Data/InputData/** folder.

```
import delimited using Data/InputData/wdi_input.csv, ///  
varnames(1)
```

→ Write a command that assigns the label "GDP pc, PPP international \$" to the variable *gdppc*.

```
label variable gdppc "GDP pc, PPP international $"
```

→ Write a command that generates a new variable called *lngdppc*, equal to the natural log of *gdppc*.

```
gen lngdppc=ln(gdppc)
```

→ Write a command that assigns the label "Natural log of GDP pc" to the variable *lngdppc*.

```
label variable lngdppc "Natural log of GDP pc"
```

→ The WDI data is now ready to be merged with the Pew data, so write a command that saves the data in a file named *wdi_intermediate.dta*.

→ Use a relative directory path to tell Stata to save this file in the **Data/IntermediateData/** folder.

```
save Data/IntermediateData/wdi_intermediate.dta, replace
```

XX. Execute and save *import_wdi.do*

→ After you have finished writing the commands described above, execute *wdi_import.do* from start to finish. Be sure that it runs without generating an error, and that it generates the *wdi_intermediate.dta* data file and saves it in the **Data/IntermediateData/** folder.

→ Save the complete *wdi_import.do* script in the **Scripts/ProcessingScripts/** folder (which is where you saved it when you first created it).

Tip: Remember that you save do-files interactively; you do not write a command in the do-file that saves the do-file.

XX. Create and save a do-file called *merge.do*.

→ Open a new do-file.

→ Give this new do-file the name *merge.do*, and save it in your **Scripts/ProcessingScripts/** folder.

XX. Write commands and comments in *merge.do*.

→ At the top of the do-file, write a comment indicating that when it is being edited or executed, Stata's working directory should be set to the **ProjectFolder/**.

→ After the comment about setting the working directory, write a command that clears Stata's memory.

→ Write a command that opens the intermediate Pew data file, *pew_intermediate.dta*.

→ Use a relative directory path to tell Stata that *pew_intermediate.dta* is stored in the **Data/IntermediateData/** folder.

```
use Data/IntermediateData/pew_intermediate.dta
```

→ Write a command that merges the intermediate Pew data (which was just loaded) with the WDI intermediate data in *wdi_intermediate.dta*.

→ This should be a 1:1 merge, matching on the variable *countryname*.

→ Use a relative directory path to tell Stata that *wdi_intermediate.dta* is stored in the **Data/IntermediateData/** folder.

```
merge 1:1 country using ///  
    Data/IntermediateData/wdi_intermediate.dta
```

→ Write a command that keeps only the observations for countries that appear in both the Pew and WDI data (i.e., countries that matched when the merge was performed).

```
keep if _merge==3
```

→ Write a command that drops the *_merge* variable.

```
drop _merge
```

→ The data is now cleaned and organized in such a way that it is ready to be used for the analysis you will conduct for this exercise. So write a command that saves the data in a file named *analysis.dta*.

→ Use a relative directory path to tell Stata to save this file in the **Data/AnalysisData/** folder.

```
save Data/AnalysisData/analysis.dta, replace
```

XX. Execute and save *merge.do*

→ After you have finished writing the commands described above, execute *merge.do* from start to finish. Be sure that it runs without generating an error, and that it generates the *analysis.dta* data file and saves it in the **Data/AnalysisData/** folder.

→ Save the complete *merge.do* script in the **Scripts/ProcessingScripts/** folder (which is where you saved it when you first created it).

XX. Create and save a do-file called *analysis.do*.

→ Open a new do-file.

→ Give this new do-file the name *analysis.do*, and save it in your **Scripts/AnalysisScripts/** folder.

XX. Write commands and comments in *analysis.do*.

→ At the top of the do-file, write a comment indicating that when it is being edited or executed, Stata's working directory should be set to the **ProjectFolder/**.

→ After the comment about setting the working directory, write a command that clears Stata's memory.

→ Write a command that opens the analysis data file, *analysis.dta*.

→ Use a relative directory path to tell Stata that *analysis.dta* is stored in the **Data/AnalysisData/** folder.

```
use Data/AnalysisData/analysis.dta
```

→ Write a command that generates a histogram of *meanswb*, where the height of each bin represents the number of observations in the bin.

→ Specify options in the `histogram` command that:

i) Generate titles at the top of the figure that look like this:

FIGURE 1
Histogram of Mean SWB

ii) Save the histogram in a file called *Figure1.png*.

→ Use a relative directory path to tell Stata that *Figure1.png* should be stored in the **Data/AnalysisData/** folder.

→ Write a sequence of commands that generate a table of summary statistics for the *meanswb* variable, and export the table to a text file.

→ The table should include the following summary statistics for *meanswb*:

i) The number of observations.

ii) The sample mean and standard deviation.

iii) The minimum and maximum values.

iv) The 25th, 50th, and 75th percentiles.

→ The table should be exported to a file named *Table1.txt*, and should be saved in the **Output/Tables/** folder.

```
collect table, statistic(count meanswb) ///
    statistic(mean meanswb) ///
    statistic(mean meanswb) ///
    statistic(sd meanswb) ///
    statistic(min meanswb) ///
    statistic(max meanswb) ///
    statistic(p25 meanswb) ///
    statistic(p50 meanswb) ///
    statistic(p75 meanswb)

collect export Output/Tables/Table1.txt, replace
```


→ Write a command that generates a histogram of *gdppc*, where the height of each bin represents the number of observations in the bin.

→ Specify options in the `histogram` command that:

i) Generate titles at the top of the figure that look like this:

FIGURE 2
Histogram of GDP pc

ii) Save the histogram in a file called *Figure2.png*.

→ Use a relative directory path to tell Stata that *Figure2.png* should be stored in the **Data/AnalysisData/** folder.

→ Write a sequence of commands that generate a table of summary statistics for the *gdppc* variable, and export the table to a text file.

→ The table should include the following summary statistics for *gdppc*:

i) The number of observations.

ii) The sample mean and standard deviation.

iii) The minimum and maximum values.

iv) The 25th, 50th, and 75th percentiles.

→ The table should be exported to a file named *Table2.txt*, and should be saved in the **Output/Tables/** folder.

```
collect table, statistic(count gdppc) ///
    statistic(mean gdppc) ///
    statistic(mean gdppc) ///
    statistic(sd gdppc) ///
    statistic(min gdppc) ///
    statistic(max gdppc) ///
    statistic(p25 gdppc) ///
    statistic(p50 gdppc) ///
    statistic(p75 gdppc)

collect export Output/Tables/Table2.txt, replace
```

Specify options in the `histogram` command that:

- i) Generate titles at the top of the figure that look like this:

FIGURE 1

Histogram of Mean SWB

merges the intermediate Pew data (which was just loaded) with the WDI intermediate data in *wdi_intermediate.dta*.

→ This should be a 1:1 merge, matching on the variable *countryname*.

→ Use a relative directory path to tell Stata that *wdi_intermediate.dta* is stored in the **Data/IntermediateData/** folder.

```
merge 1:1 country using ///  
    Data/IntermediateData/wdi_intermediate.dta
```

→ Write a command that keeps only the observations for countries that appear in both the Pew and WDI data (i.e., countries that matched when the merge was performed).

```
keep if _merge==3
```

→ Write a command that drops the *_merge* variable.

```
drop _merge
```

→ The data is now cleaned and organized in such a way that it is ready to be used for the analysis you will conduct for this exercise. So write a command that saves the data in a file named *analysis.dta*.

→ Use a relative directory path to tell Stata to save this file in the **Data/AnalysisData/** folder.

```
save Data/AnalysisData/analysis.dta, replace
```

XX. Execute and save *merge.do*

→After you have finished writing the commands described above, execute *merge.do* from start to finish. Be sure that it runs without generating an error, and that it generates the *analysis.dta* data file and saves it in the **Data/AnalysisData/** folder.

→Save the complete *merge.do* script in the **Scripts/ProcessingScripts/** folder (which is where you saved it when you first created it).

This do-file should:

--Read read the data from *pew_input.csv* (using a relative directory path to tell Stata that the file is stored in the **InputData/** folder)

--Do some steps of processing to get the Pew data ready to merge with the WDI data. (Notable among these steps is collapsing the individual-level dataset with observations for thousands of respondents in 44 countries to a country-level dataset with one observation per country, where the values of quantitative variables are the means for all respondents from each of the 44 countries. This makes it possible to

merge the Pew data with the WDI data, because the WDI data consists of country-level observations)

--Save the modified data...

Further instructions will tell students to write do-files that:

--read and modify the WDI data

--merge the modified WDI data and the modified Pew data

--save the fully processed data in a file called *analysis.dta*

--write a do-file, called *analysis.do*, that generates descriptive statistics for the key variables in *analysis.dta*, and makes scatterplots of the SWB (subjective well-being) variable against GDP per capita and unemployment, and saves the graphs and summary statistics produced in the appropriate subfolders of the **Output/** folder

--The instructions will indicate where in the directory structure students should save the do-files they write, and where the files that are produced when the do-files are executed (including data files, figures and tables) should be stored—so that they populate the empty folders they created at the beginning as their work on the exercise progresses.

Students will also write a report, in which they present the figures and tables they produced and answer some questions about them. The project instructions will give the specific questions students should answer and details about constructing the report.