

Introduction to EViews

1. Getting started

EViews is a program that allows you to easily perform statistical analysis such as Ordinary Least Squares regression. To get started, you will need to open a data set in EViews. There are three primary ways of doing this.

- 1) **Open an existing workfile.** EViews data files are saved in `.wf1` files known as workfiles. Open an existing EViews workfile by opening the EViews application, navigating to “Open” and choosing the existing workfile from the file directory, or just double click on the existing `.wf1` file to open it in EViews.
- 2) **Import EViews data from a `.csv` file.** Data is often stored in `.csv` (comma-separated values) files because these files are generic and readable by nearly any program that uses data tables. To open a `.csv` file in EViews, choose “Import” from the File menu. Navigate to the folder containing your `.csv` file, choose filetype `.csv` from the filetype dropdown menu, and open the `.csv` file.
- 3) **Import data by hand.** You can create a new workfile in EViews and manually input data by following these steps.
 1. Open the EViews application.
 2. For a new cross-sectional data set, type the following into the command window:

`wf u n`

The `wf` stands for workfile, `u` stands for unstructured (which indicates you want to create an unstructured cross-sectional data set, as opposed to a time series data set), and `n` is the number of observations in your data set. For example, if you want a cross-sectional data set with 10 observations, type `wf u 10`.

3. Create a new variable within your data set by typing:

`series name`

For example, to create a new variable called height, type `series height`.

4. Double-click on the series name, click the “Edit +/-” button, and manually enter values for the new variable for each observation.

2. Summarize and visualize your data

You may often want to examine summary statistics or see what your data look like before conducting any further analysis. Three ways of doing this are:

- 1) **Summary statistics** To get summary statistics for a series, type:

`stats name`

- 2) **Histogram** To draw a histogram for a series, type:

`hist name`

3) **Scatterplot** To draw the scatterplot for two variables, type:

```
scat xname yname
```

where **xname** is the name of the variable that will be plotted along the x-axis and **yname** is the name of the variable that will be plotted along the y-axis.

3. Creating new variables from existing variables

You may want to create a new variable from an existing variable. For example, you may want to convert it to different units. To do this you can create a new variable equal to some mathematical expression of an existing variable. For example, if you have a variable for height measured in inches but want to convert it to feet, you can type:

```
series heightft = height*12
```

You can also type such things as `series heightsq = height^2` or `series female = 1 - male`.

When using time series data, you may want to create a new series equal to some variable's value from the previous time period. To do this, type:

```
series newvar = name(-1)
```

4. Limiting the sample

You may want to limit your sample so that you perform operations on only a subset of your observations. In order to limit your sample from observation k to observation m , type:

```
smpl k m
```

Once you do this, any operation you perform will only be performed on the subsample of observations k through m . If you type, `stats height`, you will only be given the summary stats for the subsample. To revert back to the full sample, type:

```
smpl @all
```

You may also subset the sample based on a condition. For example, if you only want to look at the sample of individuals taller than 72 inches, you can type:

```
smpl if height > 72
```

5. Running a regression

Econ 140A is all about estimating OLS regressions, so this section is important! In order to estimate a least squares regression, type the following (where c is the constant in your regression):

```
ls depvar c xvar
```

For multiple regressions, type:

```
ls depvar c xvar1 xvar2
```

You can include as many independent variables as is necessary.