

Econ 452 - Graphing in STATA

Connor Cole

November 26, 2015

Graphing in Stata

We have previously talked a lot about using STATA for statistical analysis. Often a good complement to this work is using STATA to graphically present data, especially as plotting the data sometimes reveals errors in coding or distinctive patterns in the data that might be obscured by looking at regression output and t-tests alone.

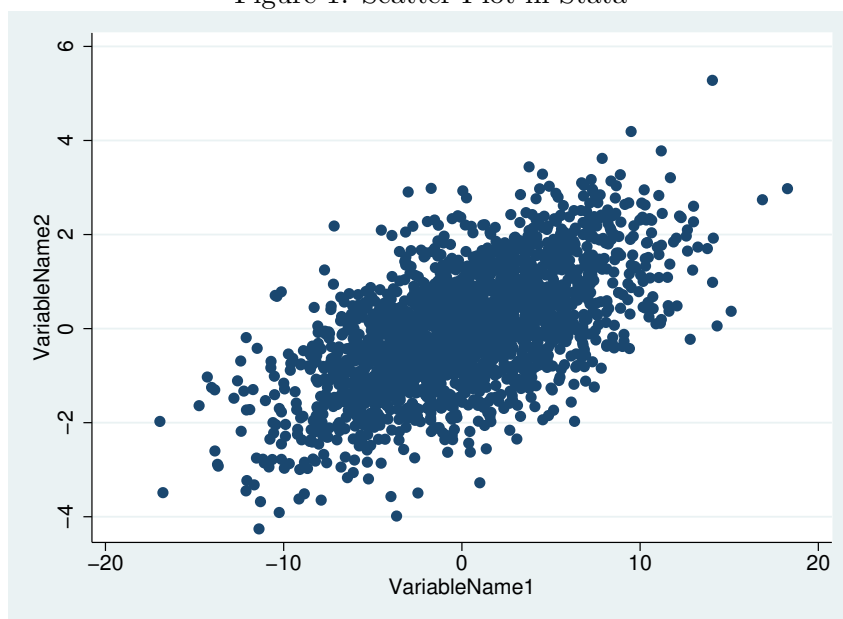
STATA has some powerful graphing capabilities that, in my opinion, are far superior to Excel and more straightforward to use than R, but learning how to use STATA's graphing capabilities fluently takes a little bit of effort. However, using STATA well for this purpose is a skill that pays dividends in any work you might do, as you will be able to make far more attractive graphs with much greater flexibility in presenting the data than people who work with Excel alone.

Scatter Plots

The simplest, and sometimes most effective way to present your data is to do a simple scatter plot of two variables. Let's say that we want to graph *VariableName2* on the *Y* axis and *VariableName1* on the *X* axis. We can do this by typing:

```
twoway scatter VariableName2 VariableName1
```

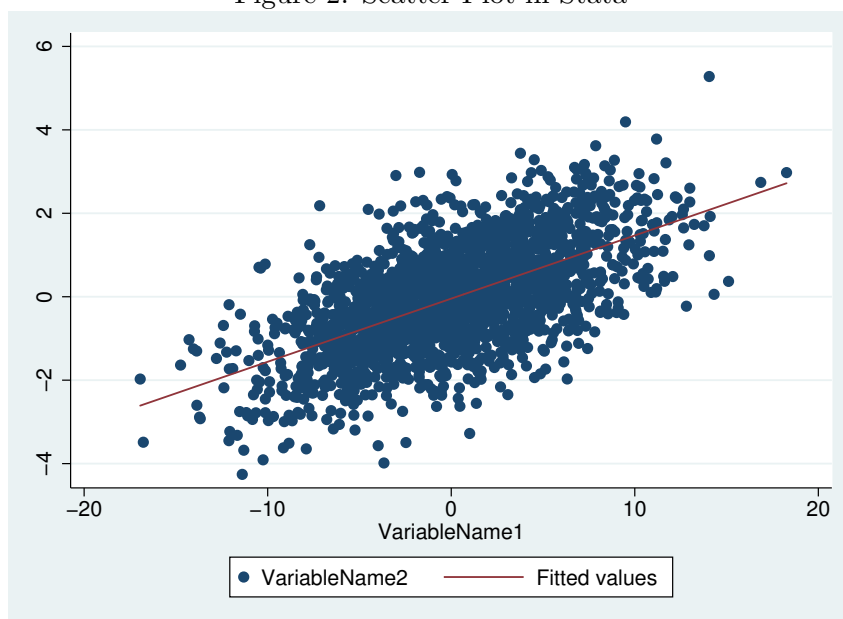
Figure 1: Scatter Plot in Stata



Sometimes it is helpful to fit a linear regression to these plots.

```
twoway (scatter VariableName2 VariableName1 ) (lfit VariableName2 VariableName1)
```

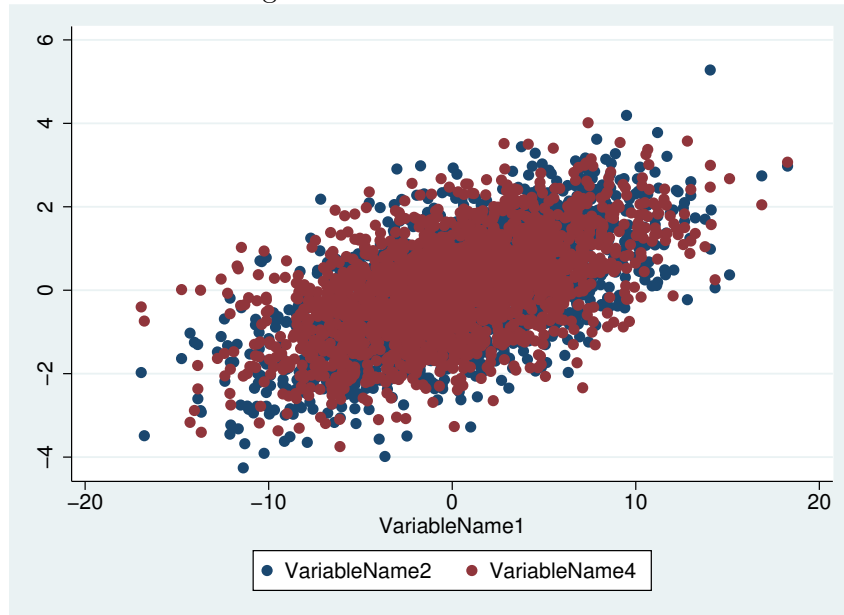
Figure 2: Scatter Plot in Stata



If we had another variable and wanted to include it in the scatter plot as well, we can just type it before we type *VariableName1*:

```
twoway (scatter VariableName2 VariableName1 ) (scatter VariableName4
VariableName1)
```

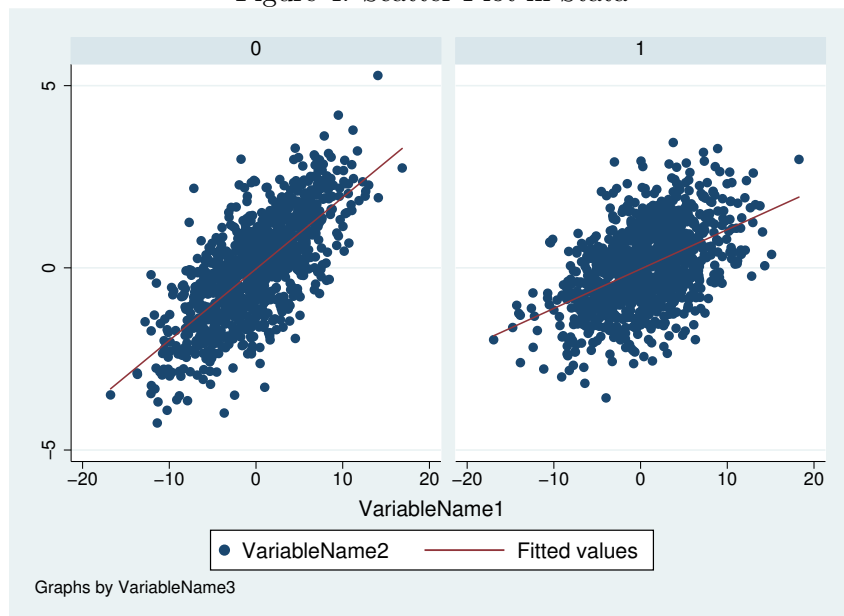
Figure 3: Scatter Plot in Stata



If we had another variable, and we wanted to do separate scatter plots of *VariableName2* on *VariableName1* and separate fit lines, we could do it by typing either:

```
twoway (scatter VariableName2 VariableName1) (lfit VariableName2
VariableName1), by(VariableName3)
```

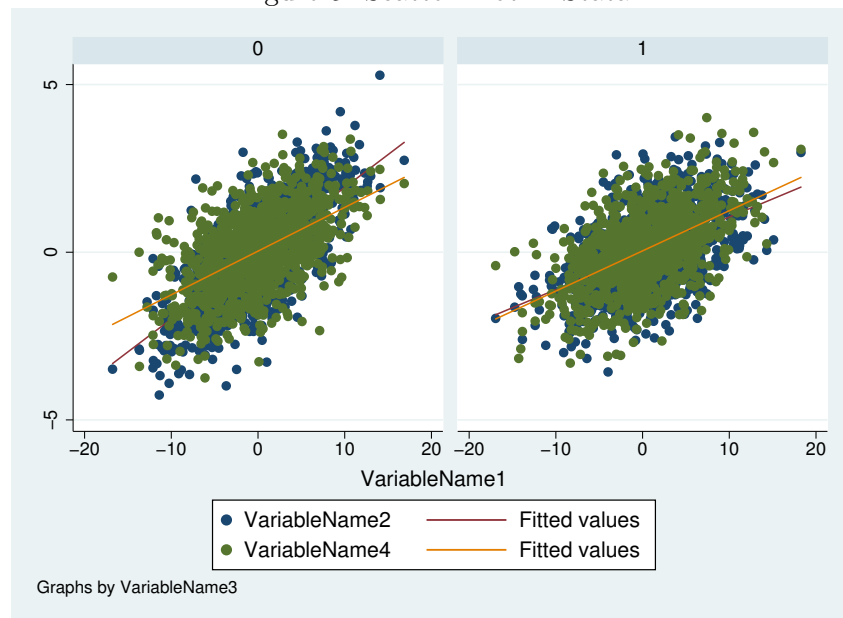
Figure 4: Scatter Plot in Stata



Note the form of the syntax here with these `twoway` plots. We can always add more objects to our graphs by adding additional statements in parentheses where the type of graph is listed first (either `scatter` or `lfit`) and then the y-axis variable is listed first followed by the x-axis variable. So, for example, we could write commands of the following form:

```
twoway (scatter VariableName2 VariableName1) (lfit VariableName2
VariableName1) (scatter VariableName4 VariableName1) (lfit VariableName4
VariableName1) , by(VariableName3)
```

Figure 5: Scatter Plot in Stata



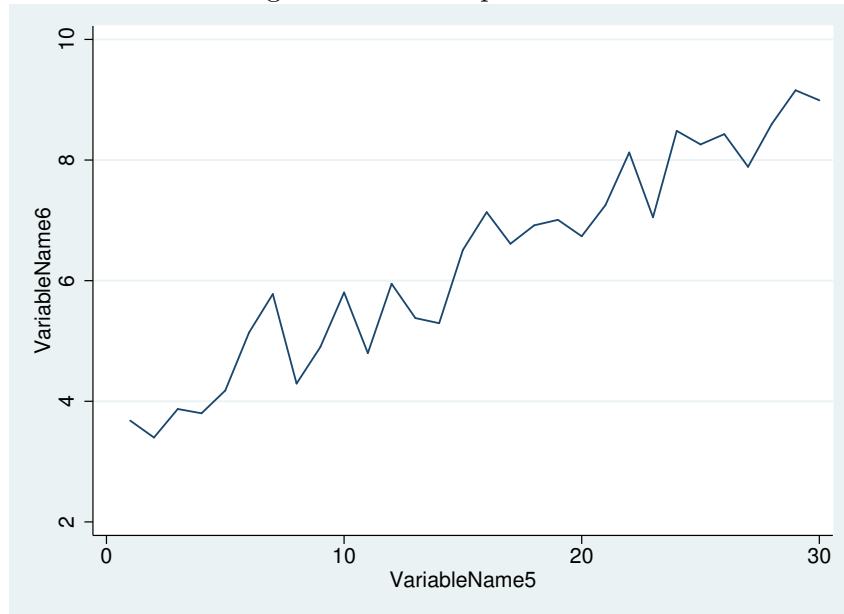
Line Graphs

Sometimes, you may have data with single observations of some variable for single observations of another variable. For example, GDP for the U.S. by year, or mean earnings of individuals by education level. In such settings, a scatter graph would not be particularly informative, as a scatter graph would not make clear that there are links between observations over time. A better graph to represent this form of data would be a line graph.

We can graph *VariableName6* on the y-axis and *VariableName5* on the x-axis by typing:

```
twoway (line VariableName6 VariableName5)
```

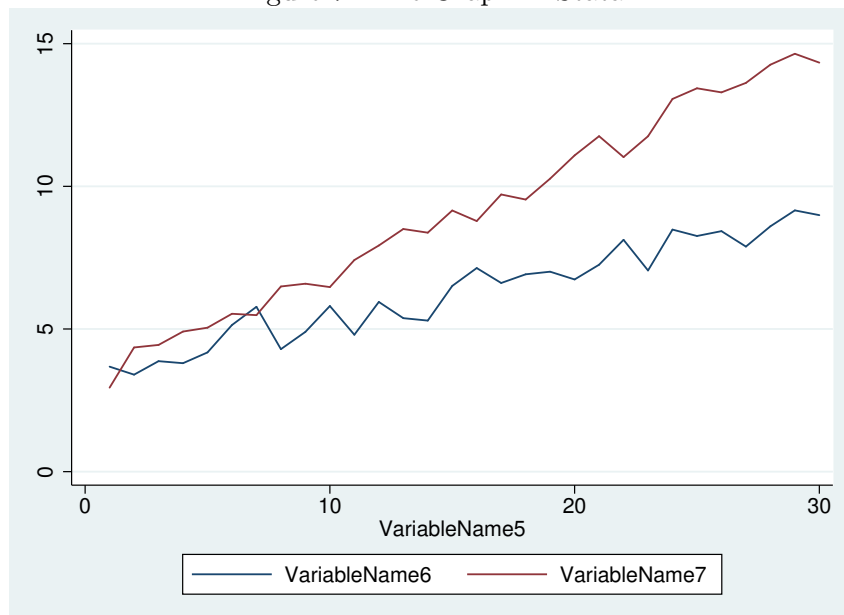
Figure 6: Line Graph in Stata



Similar to previously, we can add other graphing objects by adding additional statements in parentheses. So, we could graph the same line as in the last case and an additional line of *VariableName3* on *VariableName1* by typing:

```
twoway (line VariableName6 VariableName5) (line VariableName7 VariableName5)
```

Figure 7: Line Graph in Stata

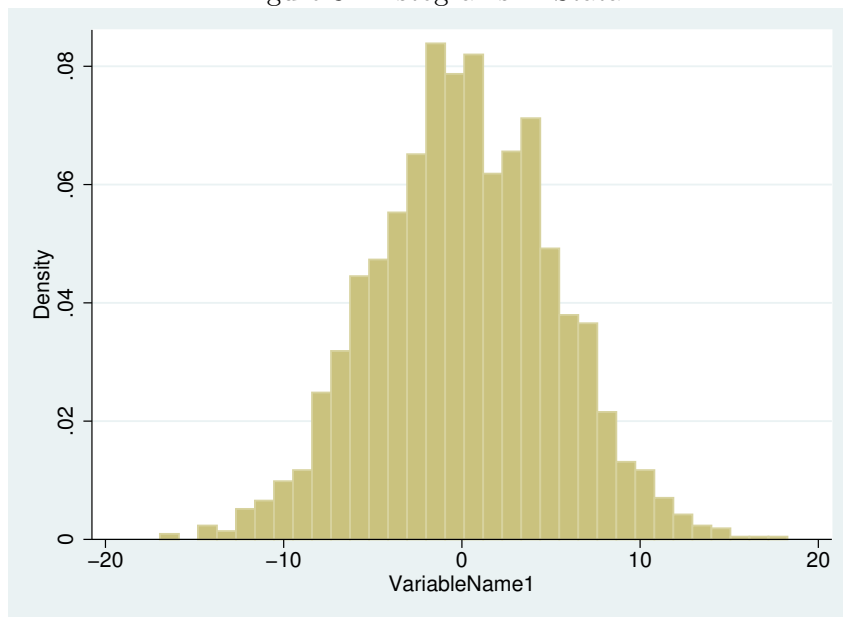


Histograms and Kernel Density Estimates

If we want to just look at the density of a single variable, then a scatter plot is not particularly informative as it does not clearly offer an indication of the percentage of observations that have values of the variable in a certain range. A much more informative graph would be a histogram or a kernel density estimate. Histograms bin data into uniformly sized groups of a single variable and compute the fraction of overall observations that have values of that variable that fall into those ranges. We can tell STATA to do this by typing:

```
histogram VariableName1
```

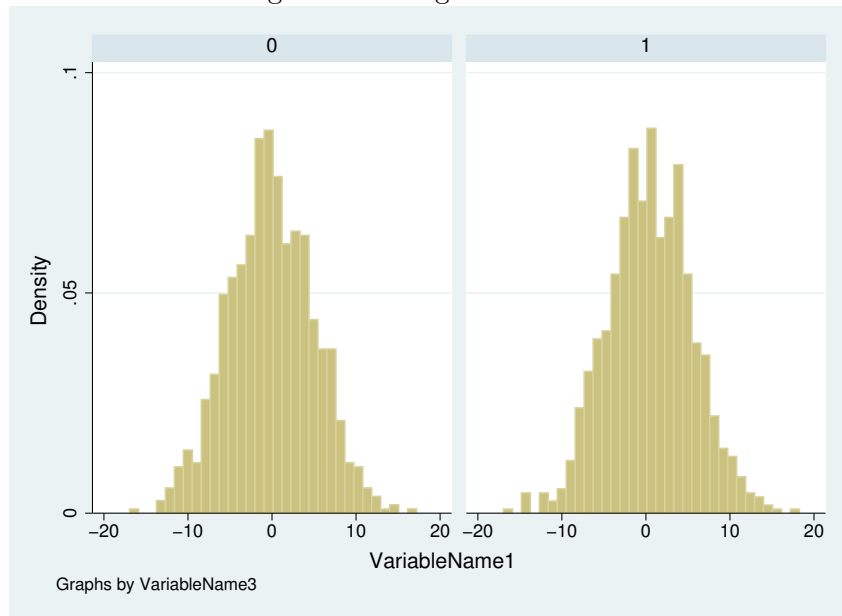
Figure 8: Histograms in Stata



If we want to create histograms separately for different subgroups defined by *VariableName3*, we can type:

```
histogram VariableName1, by(VariableName3)
```

Figure 9: Histograms in Stata

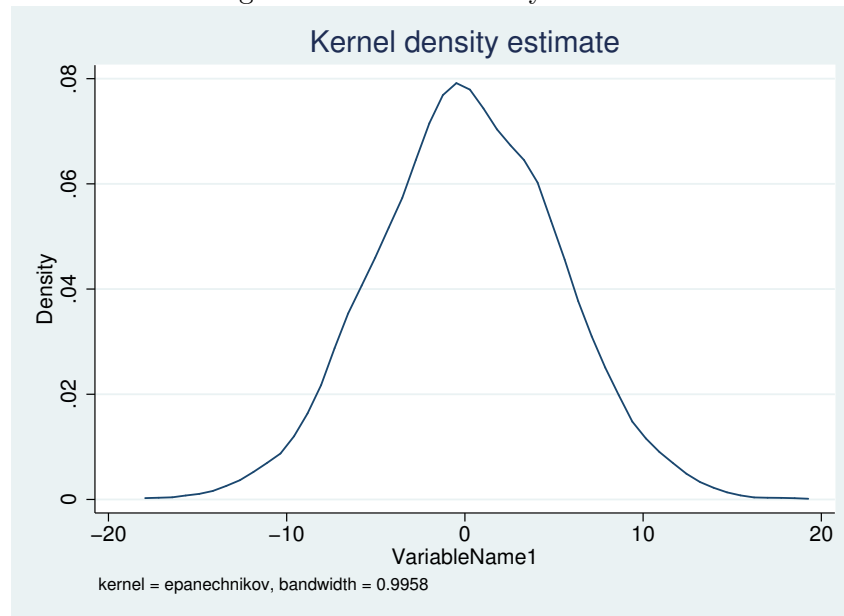


Note that STATA has automatically chosen the bin size here, as evidenced by the width of the rectangles. We can choose these parameters by specifying `bin()` or `width()`, but it is generally best to let STATA choose these parameters automatically.

An alternative to a histogram that offers a more revealing picture of the distribution of data for a variable is to estimate a kernel density function. Whereas the histogram is a simple representation of data, a kernel density applies assumptions about the data generating process to create an estimate of the shape of the density function (for data that represent some non-discrete variable, e.g. income). To apply kernel density estimation, we simply type:

```
kdensity VariableName1
```

Figure 10: Kernel Density in Stata



As it turns out, under some assumptions about tuning parameters hidden in the background, this process is consistent for the density of a continuous random variable.

Formating Graphs

There are many ways we can clean up these graphs to make them easier to understand. If we wanted to add axis titles, we could

STATA uses the variable labels as axis names unless told otherwise. So, if I wanted to relabel the axes I could just type in:

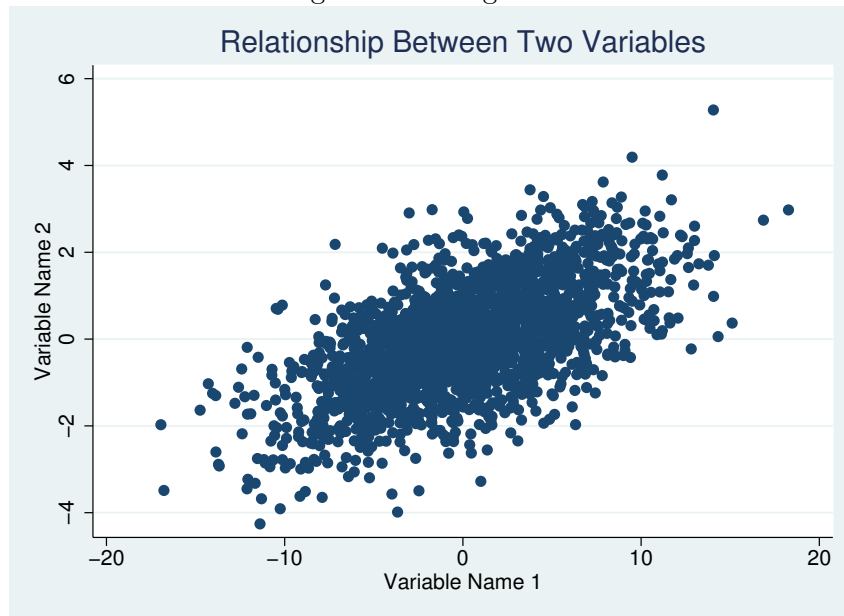
```
twoway scatter VariableName2 VariableName1, xtitle("Variable Name 1")
yttitle("Variable Name 2")
```

We could add an overall title to this graph by typing:

```
twoway scatter VariableName2 VariableName1, xtitle("Variable Name 1")
yttitle("Variable Name 2") title("Relationship Between Two Variables")
```

The outcome of this command in STATA with our data would be:

Figure 11: Using Titles



Exporting Graphics

After having created a graph, you can save it by using the `graph export` command. After creating the graph, we simply also put in:

```
graph export "PUT IN FILE PATHWAY HERE/graphname.pdf"
```