Graphical output and 'first inspections' for time series data

This workshop introduces you to the handling of basic time series data in Stata. In this session, we generate single and two-way trend graphs, perform some basic time series smoothing, and explore the range of options available to extract as much information as possible from our visual output.

1.1. Loading and describing data

Open Stata by double-clicking on the Stata desktop icon, or by navigating to the programme through the windows start menu. Stata differs from SPSS as it requires you to type commands in the command line at the bottom of the screen in order to carry out activities. For example, to specify a basic linear regression model in SPSS, you would typically click on the 'Analyze' tab, then 'Regression', then 'Linear', or you would type in the relevant command lines into the SPSS syntax dialog.

In Stata, you would enter **`reg var1 var2`** into the command line in order to run a basic linear regression. The statistical results are the same as SPSS, but everything takes place on a single screen rather than multiple windows. Most activates and analyses are performed in the same way in Stata – by typing a command in the command line, followed by the variables you wish to perform the action/analysis on, and then pressing 'Enter'. Many of the commands you come across in this module will function the same way. In the worksheets that accompany this module, whenever you see text written in **`bolded font`**, this denotes Stata commands. Remember, Stata's syntax is case sensitive – commands are generally written in lower case, and if there are any capitals in your variable names, these must be entered as they appear in the dataset. If you are struggling to remember the basis of using Stata, you should refer to chapter 1 of Kohler and Kreuter's book 'Data Analysis Using Stata', a copy of which is available from the module Queen's Online (QOL) page.

Download and save the dataset for this session from the QOL module page (under the 'Resources' tab on the left-hand side, in the data folder, open and save the file titled '**worksheet1.dta'**). This dataset contains a range of time-series socioeconomic data for the United Kingdom from 1960-2010.

When you start a new session in Stata, it is good practice to set your working directory. Essentially, you are simply telling Stata which folder you will be working from. For this session, this will be the folder in which you saved your data file – this can be on your college drive, or on a USB drive. You can do this by typing:

```
cd "C:\Users\3048874\Documents\Datasets"
```

In the example above, I have indicated the folder I use on my own computer to store data; you should specify the folder from which you are working during this class. This is not an essential step, but is good working practice as it ensures any output can be sent to, or any data loaded from the same folder.

Check the contents of the folder by typing **`dir`**, which should display the file 'worksheet1.dta'. You can load the data file (or any other datafile in the folder) by typing **`use worksheet1`**

Once you have loaded the datafile, type **des** to view the contents of the dataset. You can get a more detailed summary of each case in the data by typing **list**
You can also specify which variables you want to include in the summary output – this is a useful way to get an overview of the contents of the variables. Type **list year union** to see the values for each observation in the dataset. You can also limit the range of output by using the **in** qualifier. Try the following commands to identify specific sets of observations:

```
list year union in 1
list year union in 1/10
list year union in 15/20
```

We can generate summary statistics for each variable by using the **sum** command. Remember your class on levels of measurement, and consider when it is appropriate to report mean and standard deviation. In the same way as above, you can combine **sum** with variable names to generate summaries for particular variables. To see more details for a particular variable, we can type **detail** after the command:

```
sum union, detail
```

You can also use the **if** qualifier to select certain cases or values. For example, if we wanted to see what unionisation levels were in 1960 compared to 2009, we could type:

```
list union if year==1960
list union if year==2009
```

You can include as many variables as you wish before **if** to see the values for that particular year.

We can graphically summarise our data using a range of graph commands. Whilst these graphs are not strictly suitable for time series data, they give us some indication of the graphical possibilities of Stata. To produce a histogram, we type:

```
histogram union
```

We can produce a boxplot by typing:

```
graph box union
```

Exercise 1

Identify three variables of interest to you from the dataset, and generate some descriptive output. Summarise their values for a particular year, find out their mean values within a certain timespan, and take a look at the histogram for each variable. What did you discover about your three variables?

If you are preparing to do a lengthy piece of work in Stata, it is a good idea to work with a do-file. A do-file is a text file on which you write and execute your commands. The benefit of using do-files is that you can write multiple commands before sending them to Stata to execute. You can open a new do-file by clicking on the following icon on the top icon bar:

Try entering a number of simple commands from those we tried above. Each line should contain only a single command. Once you have finished, you can highlight the ones you wish to run (or all of the commands in the file), and click the execute button:

When you are finished with a session in Stata, provided you have set your directory using the steps outlined above, you should save your data. You can do this by typing **`save worksheet1, replace`** to overwrite your file, or by typing **`save newname`** to give your datafile a new name. When you are finished, type **`exit, clear`** to close Stata.

1.2. Exploring and graphing time series data

Load your datafile worksheet1.dta using the steps outlined above. As this is time series data, we need to tell Stata that we are working with time-ordered observations. The dataset contains a year variable, which identifies the units of time – this is also referred to as the resolution of the data. Data can be found at various levels of resolution, most commonly in the social sciences, we deal with yearly data. Economists often work with quarterly or monthly data, financial analysts with daily data, or even minute-by-minute data of stock price. In order to declare to Stata that we are working with time series data, we specify the unit of time and the interval by typing:

```
tsset year, yearly
```

Since our data are time-ordered, it makes sense to summarise the distribution graphically by retaining each observation in order. Your data are sorted by year when you open the dataset, and once the time series interval variable and period has been set, you can use the **`tsline`** command to generate line graphs:

```
tsline union
tsline open
tsline top1
```

We can also insert a reference line into our graph to examine fluctuations or trends around a key value (i.e. the mean):

```
tsline union, yline(39.662)
```

Remember one of the great techniques of how to lie with statistics is axis scaling: by manipulating axes, we can make ordinarily small trends appear large. By default, Stata specifies the minimum and maximum value of the variable as the upper and lower bands of the graph. We can properly scale the graph by specifying a full range of Y-axis values:

```
tsline union, yscale(range (0 100))
```

While the trend is clearly present, we can see it now appears less steep than before. Why do you think unionisation began a sharp decline in the 1980's?

How can we depict a relationship between two variables graphically? This is easy when both variables are measured with the same units, as we do not have to worry about different scales of measurement since both are expressed as percentages:

```
tsline union labourshare
```

We can also generate this graph as a connected-dot graph, remembering to end the command with our time variable:

```
twoway connected union labourshare year
```

This does not work as well with two time series with different scales of measurement. Try running each of these commands separately:

```
tsline union
tsline pcgdp
tsline union pcgdp
```

Why is the union variable now flat? We can rectify this by setting the right-hand axis to the units of our second variable:

```
twoway (tsline union) (tsline pcgdp, yaxis(2))
```

There is a lot we can tell from simple inspections of time series data. Consider the following variable:

```
tsline migie, yline(0)
```

While this graph reveals much about the dynamics of migration over the last century, we can start to introduce additional pieces of information to tell us more about the factors which may have influenced migration trends:

```
twoway (tsline migie) (tsline gdpie, yline(2))
```

What does this graph tell us about the relationship between migration and economic growth? We can explore this further through correlation and plotting:

```
scatter migie gdpie
```

We can then quantify the association by typing:

```
pwcorr migie gdpie, sig
```

Finally, we can plot the relationship by fitting a linear trend to the scatterplot:

```
graph twoway (scatter migie gdpie) (lfit migie gdpie)
```

You can then add the year to the plot:

```
graph twoway (scatter migie gdpie, mlabel(year)) (lfit migie gdpie)
```

As we will see, this is a crude (yet somewhat) effect first pass at exploring relationships amongst variables. Nonetheless, it starts to tell us useful information about the form of the relationship between variables, we well as flagging important periods, events, or epochs we may wish to investigate more closely later on.

While the syntax of Stata is powerful enough to cope with minute graphic alterations, this is quite cumbersome for a casual user (of which I count myself). You can start the graph editor by clicking on 'File – Start Graph Editor' after running a graph command. Try opening graph editor for the following graph:

```
twoway connected union year
```

By navigating through the various options in graph editor, we can customise any aspect of the chart we wish without having to resort to excessively cumbersome syntax.

Exercise 2

Select two variables from the dataset which you suspect may be related. Choose a suitable method of graphing, and generate some summary output. What do you conclude?

So far, we have looked at variables with relatively simple dynamics – either strongly trended, or showing little volatility. This is quite typical of institutional variables in the social sciences which are often slow to change (think about this intuitively – things like unionisation rates, income inequality, educational participation, infant mortality do not tend to change very quickly in the absence of severe events). Some variables may exhibit more complex dynamics making it difficult to discern any underlying trend. This is where smoothing might be helpful, but should generally be used sparingly, and only for illustration.

We can generate a simple moving-average smoothed series from an existing variable by using the following command. This smoother uses two lagged values, the current observation, and two future values to convert each data point to a five-year average:

```
tssmooth ma sm1=union, window(2 1 2)
```

We can then display the new series on its own, or overlaid onto the existing series:

```
tsline fdi sm1
```

Additional graph commands

```
tsline union, name(union, replace)
tsline labourshare, name(labourshare, replace)
graph combine union labourshare
graph combine union labourshare, rows(2)
```

1.3. Getting started with modelling

The base commands for conducting regression analysis in Stata are quite simple. To run a basic bivariate ordinary least squares regression, we type reg followed by our dependent and independent variable. Specify the following model to explore the relationship between unionisation and income inequality:

```
reg gininet union
```

Looking at our model diagnostics (t-statistics, R^2, F) would you say this is a good model? Remember, in terms of time series model specifications, this is what we call a 'static' model – it has no dynamics, and assumes the impact of the independent on dependent variable is instantaneous. This is not always an accurate specification for a system of this kind, although it will suffice for illustration of Stata's regression commands.

We can also estimate the model with standardised coefficients. To do this, we simple add the option **,b** after the model specification:

```
reg gininet union, b
```

We can also run the model for a specific time interval by using the **if** qualifier:

```
reg gininet union if year>1969 & year <1991
```

We can check for heteroscedastic errors by producing a plot of the model residuals versus fitted values, or by running Cook-Weisberg test for constant error variance. These commands should be run directly after the regression model is specified:

```
reg gininet union
rvfplot
hettest
```

We can run multiple models and store our estimates for tabulation afterwards by using the estimates store command:

```
reg gininet union
estimates store model1
reg labourshare union
estimates store model2
estimates table model1 model2, star stats (N r2 r2_a)
```

This allows us to quickly compare model coefficients and diagnostics. In practice, we can include as many pieces of information from the model as we wish. To view the full list of stored scalars, types ereturn list after running the model:

```
reg gininet union
ereturn list
```

Before we move on to dealing with the specific issues of time series analysis, it is worth exploring the possibilities available with careful use of trends, trend dummies, and shift dummies. In substantive terms, we refer to the presence of acute changes in the trending behaviour of our data as 'structural breaks'. Although these present certain statistical issues which need to be rectified, they also represent an important issue of substance for social scientists – they are the registering of sudden change or new trends, often linked to a change in the composition or behaviour of a given social system. As such, they are of much substantive importance (see lecture slides for further details on structural breaks). Although there are specific tests we can use to identify the precise moments of a structural break, we can begin to explore variable dynamics within a time series using simple OLS techniques.

The most basic model we can specify is a regression of a dependent variable on a trend dummy. In our case, this will be the year, but in practice it can be any sequence of equal-interval numbers (i.e. 1, 2, 3, 4…) In practice, a trend regression gives us an approximation of the rate of change of the variable over the lifespan of the series:

```
reg union year
```

However, as we can see from our line graph, this is not an accurate representation of the behaviour of the series from 1960-2010. There are periods of stagnation, increase, and decrease, yet the trend has simply picked up the average rate of change over the entire series. We can view this by fitting a linear trend to the unionisation variable:

```
twoway (tsline union) (lfit union year)
```

We can overcome this difficulty by fitting restricted trend dummies to the time series. Open the data editor using the **edit** command, and take a look at variables t1 to t10. What do you think will happen if we replace year in the above regression with t1 or t2?

```
reg union t1
reg union t2
```

Trend dummies are thus a useful way of capturing and describing the variability of a trended time series with discrete changes in trend direction. Take a look at the Flaherty and Ó Riain (2015) chapter included in the online resources and your reading list to see how we compared trend dummies in order to quantify the rate of change in labour's share of GDP between Ireland and Denmark. You could also use this technique to compare variable rates of change across any number of countries or cases.

Exercise 3

Graph a time series of interest to you from the dataset, list its values, and identify a point of change in the series. Generate two trend dummies to summarise its behaviour. You can generate a blank variable by using the **gen** *command*:

```
gen variablename=.
```

This will generate a blank variable which you can fill using the data editor.

Exercise 4

Go to the data resources folder on the module Queen's Online page. Thinking about your own research interests, try to identify some variables, either from the supplied dataset for this class, or from the resources provided online that may be of relevance to you. This does not have to be strictly related to your research topic, but it might be useful to include in your dissertation as a context to your research question. Is there an important country-level variable that might be of relevance? If your research question likely to be influenced by, or a product of, patterns of long-term change? Could you compare two countries over time?

Try comparing regressions on the original and differenced series for **gininet** and **union**:

```
reg gininet union
reg d.gininet d.union
```

What do you notice?

Now try specifying the model in lags of 1 and 2:

```
reg gininet union
reg gininet l.union
reg gininet l2.union
reg gininet l3.union
```

What can you conclude about the impact of unionisation on inequality? Is there a temporal effect? If so, how would you characterise it?

We can visualise this progressive lagged effect by plotting a cross-correlogram of the correlation between the dependent variable and various lags of the independent variable (note for this graph, the independent variable must come first – this is a rare exception to the general rule):

```
xcorr union gininet
```

When is the impact of unionisation strongest (hint: start from lag 0 and look back toward the negative values)?

Next week, we take a closer look at diagnosing autocorrelation and non-stationarity in our models and variables, and examine some more detailed approached to dealing with structural breaks. We also consider a number of model specifications which help us grasp the dynamics of social and political systems, whilst dealing with issues of non-stationarity and autocorrelation.