1.1.1 Defining Time-Series in Stata

In order to take advantage of Stata’s many built-in functions for analyzing time-series data, one has to declare the data in the set to be a time-series. Since time-series are ordered in time their position relative to the other observations must be maintained. It is, after all, their temporal relationships that make analysis of this kind of data different from cross-sectional analysis.

If the data you have do not already have a proper date to identify the time period in which the observation was collected, then adding one is a good idea. This makes identification of historical periods easier and enhances the information content of graphs considerably. The data sets distributed with your book have not been declared to be time series and most do not contain the relevant dates in the set of variables. So, the first order of business is to add this information to the data set and then to use the dates to identify the observations as time-series and indicates the period of time that separates the individual observations (e.g., daily, monthly, quarterly, yearly). In analyzing the time dependencies in the data, this is vital information as will be explained below.

Before getting to the specific examples from the text, something should be said about how Stata handles dates and times. Basically, Stata treats each time period as an integer. The integer records the number of time units (whatever you define them to be) that have passed from an agreed-upon base, which for Stata is 1960.

For example, for 100 quarterly data observations that start in 1960 we could generate Stata dates using

```
set obs 100
generate date = tq(1960q1) + _n - 1
```

The `tq(1960q1)` is referred to as a pseudofunction. They are called pseudofunctions because they translate what you type into their integer equivalents. The integer equivalent of 1960q1 is 0—that is how many quarters have passed since the first one in1960. The second quarter is set to 1 and so on. Adding `_n - 1` is done to increment the observations by 1. Listing the first 5 observations of date reveal:

```
. list date in 1/5

    date
1.    0
2.    1
3.    2
4.    3
5.    4
```

which is exactly what we expect.

To make this meaningful for people, these need to be converted to strings that make it easy for us to tell what date is 20 quarters from 1960. This is done using a `format` command.

```
format %tq date
```

Translates the integers contained in `date` back into year-quarter form.
Notice that the `%tq` function has converted the original integers 0, 1, 2, and 3 into 1960q1, 1960q2, and so on. Finally, the observations are declared to be time-series using the `tsset` command followed by the variable name that identifies the time variable.

```
tset date
```

Once the data are declared to be time-series, Stata prints out important information about your time-series. It identifies the name of the time variable, the dates it covers, and the \texttt{delta} or the period of time that elapses between observations. Check this carefully whenever generating dates to make sure that those created match what is desired.

Stata includes other functions and pseudofunctions for defining weekly (\texttt{%tw}), monthly (\texttt{%tm}), yearly (\texttt{%ty}) and others. To see other options and to learn more about how they operate type

```
help dates and times
```

at the \texttt{Command} window and Stata open a \texttt{viewer} window and carry you to the relevant information.

\textit{Okun data set}

The first thing to do is to change the directory to the one containing your data, open a new log file, and load your data. In this exercise we'll be using the \texttt{okun.dta} data.

```
cd c:\data\poe4stata
use okun, clear
```

This data set contains two variables, \texttt{g} and \texttt{u}, that are quarterly observations on the percentage change in Gross Domestic Product and the unemployment rate for the U.S. from 1985q2 to 2009q3, respectively. Once the data are loaded, a date is assigned using the generate command. Stata includes special functions for creating dates which translate the way Stata treats dates (integers) and the way people do (days, months, years, etc.).

The quarterly data begin in the second quarter of 1985. To establish dates and convert all of the variables to time-series use:

```
generate date = tq(1985q2) + _n - 1
list date in 1
format %tq date
list date in 1
```

Notice that the \texttt{tq} function has converted the original integers 0, 1, 2, and 3 into 1960q1, 1960q2, and so on. Finally, the observations are declared to be time-series using the `tsset` command followed by the variable name that identifies the time variable.
### Time-Series Plots

Once the data are loaded, the time variable generated, formatted and the variables declared as time-series, you are ready to begin the initial phases of analysis. With time-series, there is no better place to start than plotting the variables against time. This will reveal important features of the data (e.g., stationarity, trends, structural breaks, etc.).

To plot the unemployment rate and GDP growth rates the `tsline` plot is used. In order to get the labels of both plots on the same graph, the labels are shortened using the `label var` commands. Then `tsline`, which is an abbreviation of `graph twoway tsline`, plots both series in the same graph.

```
label var u "% Unemployed"
lable var g "% GDP growth"
tsline (u) (g)
```

The two time-series graphs are overlaid since each of the series to be graphed are enclosed in parentheses. Other options can be used, but we will keep it simple at this point.

The Stata graphs appear below:
The Unemployment series (top) shows a wider range of variation than GDP growth, but less variance from one time period to the next. There are no obvious trends, breaks, or other features that would suggest that either of the variables is nonstationary. Therefore, these variables are probably well-suited for the traditional regression techniques discussed in this chapter. In Chapter 12 more formal tests are developed to explore the possible nonstationarity of the series. For now it is assumed that they are stationary.

1.1.3 Stata’s Lag and Difference Operators

As seen before, the list command is used to print variables from the data set to the screen. In this case it is used with in 1/5 and 96/98 to limit the observations. The variables that are printed use another instance of Stata’s unary operators that were first explored in Chapter 5.

Stata includes special unary operators that can be used to make taking lags and differences of time-series data very easy and efficient. These operators are documented in the Stata User’s Manual under the heading Time-series varlists. Here is a list of operators and their meanings:

<table>
<thead>
<tr>
<th>Operator</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>L.</td>
<td>lag (x_{t-1})</td>
</tr>
<tr>
<td>L2.</td>
<td>2-period lag (x_{t-2})</td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
<tr>
<td>F.</td>
<td>lead (x_{t+1})</td>
</tr>
<tr>
<td>F2.</td>
<td>2-period lead (x_{t+2})</td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
<tr>
<td>D.</td>
<td>difference (x_t - x_{t-1})</td>
</tr>
<tr>
<td>D2.</td>
<td>difference of difference (x_t - 2x_{t-1} + x_{t-2})</td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>
S.  "seasonal" difference (x_t - x_{t-1})
S2.  lag-2 (seasonal) difference (x_t - x_{t-2})
...

These (unary) operators operate on the variable that follows the period. For instance, L. u takes the variable u and lags it one period. Similarly, D. u takes the one period time difference \( u_t - u_{t-1} \). The lag and difference operators are linear and can be used together in any order. For instance to take the lagged difference between the observations in u (i.e., \( ldu_t = u_{t-1} - u_{t-2} \)) one can use L. D. u. This works right to left: take the difference of u and then lag it one period. Linearity in operations implies this is equivalent to D. L. u—lag u one period and then difference. It is also true L. L=2. To lag the variable u two periods, then use L. L. u or, more simply, L2. u. The number following L indicates how many periods in the past to lag the variable. Thus L2. u lags u two periods (i.e., \( = u_{t-2} \)). There are time-series operators that create leads (F) and seasonal differences (S). Just as in the case of the unary operators for factor variables, these time-series operators save one from having to separately generate variables to include in a model. There are several other shortcuts that will be discussed below.

To demonstrate the use of these operators the variables, lags and differences are listed below for observations at the beginning and end of the data set. In general, it is often good practice to print a few observations to ensure that the contents of the series make sense and that the time periods have been assigned to the correct variables. Below the date, u, the change in u, g, and several lags are printed using the time-series operators. These match the observations in Table 9.1 in Principles of Econometrics, 4th edition (POE4).

```
list date u L.u D.u g L1.g L2.g L3.g in 1/5
list date u L.u D.u g L1.g L2.g L3.g in 96/98
```

<table>
<thead>
<tr>
<th></th>
<th>date</th>
<th>L.</th>
<th>D.</th>
<th>L.</th>
<th>L2.</th>
<th>L3.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1985q2</td>
<td>7.3</td>
<td>.</td>
<td>1.4</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>2</td>
<td>1985q3</td>
<td>7.2</td>
<td>7.3</td>
<td>-.1</td>
<td>2</td>
<td>1.4</td>
</tr>
<tr>
<td>3</td>
<td>1985q4</td>
<td>7</td>
<td>7</td>
<td>-.2</td>
<td>1.4</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>1986q1</td>
<td>7</td>
<td>7</td>
<td>0</td>
<td>1.5</td>
<td>1.4</td>
</tr>
<tr>
<td>5</td>
<td>1986q2</td>
<td>7.2</td>
<td>7</td>
<td>-.9</td>
<td>1.5</td>
<td>1.4</td>
</tr>
</tbody>
</table>

```
list date u L.u D.u g L1.g L2.g L3.g in 96/98
```

<table>
<thead>
<tr>
<th></th>
<th>date</th>
<th>L.</th>
<th>D.</th>
<th>L.</th>
<th>L2.</th>
<th>L3.</th>
</tr>
</thead>
<tbody>
<tr>
<td>96</td>
<td>2009q1</td>
<td>8.1</td>
<td>6.9</td>
<td>1.2</td>
<td>-.2</td>
<td>.3</td>
</tr>
<tr>
<td>97</td>
<td>2009q2</td>
<td>9.3</td>
<td>8.1</td>
<td>1.2</td>
<td>-.2</td>
<td>.3</td>
</tr>
<tr>
<td>98</td>
<td>2009q3</td>
<td>9.6</td>
<td>9.3</td>
<td>.8</td>
<td>-.2</td>
<td>-.2</td>
</tr>
</tbody>
</table>