***EViews* Exercises for Chapter 2**

**Distributional transformations**

**[2.3]** To obtain the histograms shown in Figure 2.1, open the workfile rpi.wf1 and then, on opening the variable rpi, click ***View/Descriptive Statistics & Tests/Histogram and Stats***. To remove the ‘stats box’ click within it and then press the keyboard ‘delete’ key. To obtain the histogram for the logarithms of the RPI, issue the command

genr p = log(rpi)

and repeat the procedure. Figure 2.2 is a graph of p. The ratio of cumulative standard deviations shown in Figure 2.3 may be obtained by issuing the command

genr ratio\_sd = @cumstdev(rpi)/@cumstdev(p)

**[2.4]** The plots shown in Figure 2.4 use the workfile rainfall.wf1. The Box-Cox transformed rainfall series is obtained by issuing the command

genr rain\_bc = (rainfall^0.5 -1)/0.5

Clearly, a value other than 0.5 may be used if desired. To obtain the empirical density shown in the left side-bars of the plots, check ‘Kernel Density’ in the ‘Axis borders’ drop-down box in the graph options.

**[2.7]** The ML estimate of the Box-Cox parameter was obtained using the *EViews* program boxcox\_rain.prg

genr y = rainfall

genr t = @trend

for !1 = 1 to 12

genr s{!1} = @seas(!1)

next

matrix (100,2) logl

!count = 0

for !j = 0.01 to 1 step 0.01

scalar lambda = !j

genr ybc = (y^lambda - 1)/lambda

ls ybc s1 s2 s3 s4 s5 s6 s7 s8 s9 s10 s11 s12 s1\*t s2\*t s3\*t s4\*t s5\*t s6\*t s7\*t s8\*t s9\*t

s10\*t s11\*t s12\*t

scalar ll = -(@regobs/2)\*log(@ssr) + (lambda-1)\*@sum(log(y))

!count = !count + 1

logl(!count,1) = !j

logl(!count,2) = ll

next

This sets up a trend and a set of seasonal dummies and then defines a matrix containing two columns. The first column contains values of the Box-Cox parameter, which takes on values between 0 and 1 in steps of 0.01 (this can be changed if desired). The second is the associated log-likelihood computed from the model as defined in **§2.7**. From this matrix the ML estimate of the Box-Cox parameter may be extracted and a relevant confidence interval constructed.

**[2.8]** The data for Figure 2.5 is contained in the workfile Kefalonia.wf1. The ML estimate of the GP transformation may be obtained with the program gp\_kef.prg:

genr y = rainfall

for !1 = 1 to 12

genr s{!1} = @seas(!1)

next

matrix (100,2) logl2

!count = 0

for !j = 0.01 to 1 step 0.01

scalar lambda = !j

genr ybc = ((y+1)^lambda - 1)/lambda

ls ybc s1 s2 s3 s4 s5 s6 s7 s8 s9 s10 s11 s12

scalar ll = -(@regobs/2)\*log(@ssr) + (lambda-1)\*@sum(log(y+1))

!count = !count + 1

logl(!count,1) = !j

logl(!count,2) = ll

next

Similarly, the IHS estimate is obtained using the program ihs\_kef.prg, although to obtain the upper bound of the 95% confidence interval the range of the IHS parameter needs to be extended:

genr y = rainfall

for !1 = 1 to 12

genr s{!1} = @seas(!1)

next

matrix (200,2) logl

!count = 0

for !j = 0.01 to 2 step 0.01

scalar lambda = !j

genr ihs = log(lambda\*y + @sqrt((lambda^2)\*(y^2) + 1))/lambda

ls ihs s1 s2 s3 s4 s5 s6 s7 s8 s9 s10 s11 s12

scalar a1 = -(@regobs/2)\*log(@ssr)

series s = log(1 + (lambda^2)\*(y^2))

scalar a3 = -0.5\*@sum(s)

scalar ll = a1 + a3

!count = !count + 1

logl(!count,1) = !j

logl(!count,2) = ll

next

**Stationarity inducing transformations**

**[2.9-2.10]** The first differences of wine and spirits consumption plotted in Figure 2.6 may be obtained using the workfile wine\_spirits.wf1 by issuing the commands

genr d\_w = d(wine)

genr d\_s = d(sprits)

Figures 2.7 and 2.8 uses the workfile bj\_series\_c.wf1 with the second differences and two-period differences constructed using the commands

genr d2\_c = d(series\_c,2)

genr d\_c\_2 = d(series\_c,0,2)

**[2.11]** Using the workfile gold.wf1, the monthly percentage return shown in Figure 2.9 is calculated using the command

genr return = @pc(gold)

This is equivalent to the command

genr return =100\*d(gold)/gold(-1)

**[2.12-2.14]** Using the workfile rpi.wf1, the annual rate of inflation of the RPI, shown in Figure 2.10, is calculated using the command

genr infl = @pcy(rpi)

This is equivalent to the command

genr infl = 100\*d(rpi,0,12)/rpi(-12)

The annualised monthly rate of inflation is calculated using the command @pca(rpi).

Figure 2.11 uses the monthly rate of inflation, calculated as

genr infl\_mon = @pc(rpi)

or

genr infl\_mon = 100\*d(rpi)/rpi(-1)

If, as in **§2.14**, the log difference is preferred as a rate of change, then the commands

genr infl = 100\*dlog(rpi,0,12)

and

genr infl\_mon = 100\*dlog(rpi)

may be used.

**Decomposing a time series and smoothing transformations**

**[2.18]** Figure 2.12 uses the workfile dollar.wf1. The moving averages are calculated as

genr doll\_ma\_c\_251 = @movavc(dollar,251)

genr doll\_ma\_60 = @movav(dollar,60)

**[2.19]** Figure 2.13 uses the workfile global\_temps.wf1. To obtain the H-P trend filter, click ***Proc/Hodrick-Prescott Filter…*** and change ‘Lambda’ to 100000 before OK-ing. The figure plots temp and the trend, named by default hptrend01.

**[2.21-2.22]** The seasonal adjustment example uses the workfile beer.wf1. To obtain the trend component, issue the command

genr trend = 0.125\*(beer(-2) + beer(2)) + 0.25\*(beer(-1) + beer + beer(1))

The seasonal factors are calculated by opening beer and clicking ***Proc/Seasonal Adjustment/ Moving Average Methods***, checking ‘Difference from moving average – Additive’, and typing factors (say) in the ‘Factors (optional)’ box. The irregular component may then be calculated with

genr irreg = beer – trend – factors

Figure 2.14 is constructed from individual figures plotting trend, factors and irreg while Figure 2.15 plots the seasonally adjusted series, named by default as beersa.