***EViews* Exercises for Chapter 5**

**EXAMPLE 5.1: Unit root tests on the spread and the $/£ exchange rate**

This example again uses the workfile interest\_rates.wf1 that was used in Examples 3.2 and 4.1. On generating the spread as spread = r20 - rs, the equivalent form of the AR(2) model can be estimated with

ls spread c spread(-1) d(spread(-1))

The ADF regression and accompanying statistics may be obtained automatically by opening spread and clicking ***View/Unit Root Tests/Standard Unit Root Test…***, checking ‘User Specified’ as the ‘lag length’ option and changing ‘4’ to ‘1’. (In fact, keeping the default ‘Automatic lag length’ selection will produce the same ADF regression.)

A unit root test for the $/£ exchange rate may be obtained by opening the workfile dollar.wf1, opening the series dollar and calculating the unit root test as above. The mean deviation from of the AR(2) process is obtained with

ls dollar c ar(1 to 2)

**EXAMPLE 5.2: Is there a unit root in global temperatures?**

This example uses the workfile global\_temps.wf1. A unit root test on the series temps may be constructed in the usual way with an automatic lag length setting.

**EXAMPLE 5.3: Trends in wine and spirits consumption**

This example uses the workfile wine\_spirits.wf1. Unit root tests of the series may be constructed as above but this time checking ‘Trend and intercept’ in the ‘Include in test equation’ option to obtain a test of the TS alternative against the DS null.

**EXAMPLE 5.4: Are U.K. equity prices trend or difference stationary?**

This example uses the workfile ftse.wf1. The logarithmic scale for Figure 5.4 is obtained by choosing ‘logarithmic scaling’ for the left axis in the Data Scaling options window in the graph view of the series price. The log price series is obtained with genr p = log(price).

**EXAMPLE 5.5: Are shocks to British GDP temporary or permanent?**

This example uses the workfile gdp.wf1. To construct Figure 5.6 the logarithms of real GDP per capita and a trend are first obtained as

smpl 1822 1913

genr y = log(gdp)

genr t = @trend

The TS model is then fitted (using the CLS estimation option) by

ls y c t ar(1)

The fitted trend line is then computed with

genr y\_tr = c(1) + c(2)\*t

Figure 5.6 is then constructed using the series y and y\_tr. The DS model is obtained with

ls d(y) c

and the ADF regression for y is obtained in the usual way

**EXAMPLE 5.6: Is Box and Jenkins’ Series C *I*(2)?**

This example uses the workfile bj\_series\_c.wf1. The plots of the ACFs and PACFs in Figure 5.6 are obtained as previously. The ADF regression testing for two unit roots is obtained by testing for a unit root in series\_c in the usual way but now checking ‘1st difference’ and ‘None’. The single unit root ADF regression is obtained in the usual way while the stationary AR(2) and ARIMA(1,1,0) models are obtained with

ls series\_c c ar(1 to 2)

ls d(series\_c) d(series\_c(-1))

**EXAMPLE 5.7: More unit root tests on the *All Share* index**

This example again uses workfile ftse.wf1 and the series p = log(price). The various unit root tests are obtained in the usual way (with ‘trend and intercept’ checked) but now selecting the appropriate statistic in the ‘Test Type’ drop-down list: ‘Phillips-Perron’ produces ; ‘Dickey-Fuller (ERS)’ *DF-GLS*; ‘Elliott-Rothenberg-Stock Point-Optimal’ ; and ‘Kwiatkowski-Phillips-Schmidt-Shin’ KPSS. Selecting ‘Ng-Perron’ produces four statistics: ‘MZa’ is ; ‘MZt’ is ; ‘MSB’ is ; and ‘MPT’ is .

**EXAMPLE 5.8: Estimating the trend is Central England temperatures robustly**

This example uses the workfile cet.wf1. The complete set of calculations required for the example may be obtained by running the program hlt.prg

genr y = cet

y.uroot(adf, exog=trend, lagmethod=sic,save=mout\_1)

y.uroot(kpss, exog=trend, lagmethod=sic,save=mout\_2)

scalar u = mout\_1(3)

scalar v = mout\_2(3)

scalar k = 0.00025

scalar cv = 1.96

equation eq\_levels.ls(cov=hac, covkern=bart,covbw=neweywest,covlag=a, covinfosel=aic) y @trend c

scalar b\_0 = @coefs(1)

scalar s\_0 = @stderrs(1)

equation eq\_diffs.ls(cov=hac, covkern=bart,covbw=neweywest,covlag=a,covinfosel=aic) d(y) c

scalar b\_1 = @coefs(1)

scalar s\_1 = @stderrs(1)

scalar lam = exp(-k\*(u/v)^2)

scalar den = ((1-lam)\*s\_1) + lam\*s\_0

scalar b\_lam = ((1-lam)\*b\_0\*s\_1 + lam\*b\_1\*s\_0)/den

scalar b\_lam\_se = s\_0\*s\_1/den

scalar b\_lam\_int = cv\*b\_lam\_se

scalar b\_lam\_upp = b\_lam + b\_lam\_int

scalar b\_lam\_low = b\_lam - b\_lam\_int

scalar marg\_sig\_lev = 1 - @cnorm(z\_lam)

**EXAMPLE 5.9: Persistence in the Nile river flow**

This example uses the workfile riverflow.wf1. Figure 5.8 may be constructed using the series flow in the usual way. The LM test for fractional differencing may be computed using the program fractdiff\_lm.prg

genr y = flow

ls y c y(-1) y(-2)

genr e = resid

stom(e,z)

scalar n = @rows(z)

matrix a = @identity(n)

for !i = 2 to n

for !j = !i to n

a(!j,!j-!i+1) = 1/!i

next

next

vector (n) z\_str = a\*z

mtos(z\_str, v\_str)

smpl 1872 1970

ls e v\_str(-1) flow(-1) flow(-2)

The ARFIMA(0, *d*, 0) model may be estimated with the command

ls flow c d

This regression will also provide the GPH estimate of *d*, since the estimation uses this estimate as an initial value of the fractional differencing parameter. To show the GPH estimate, check ‘Display settings in output’ in the ‘Equation Estimation’ window. In the ‘Stats’ view the GPH estimate is c(2) in the ‘Initial values’ list.

**EXAMPLE 5.10: Is there long memory in the *S&P 500* stock market index?**

This example uses the workfile sandp500.wf1. The following commands generate the various series used in the example:

genr lp = log(p)

genr ret = d(lp)

genr sq\_ret = ret^2

genr abs\_ret = abs(ret)

Figures 5.9 and 5.10 are constructed in the usual ways. The *R/S* statistics are computed using the program rs.prg, shown here for the series ret.

genr x = ret

genr xdev = x - @mean(x)

smpl 2 2

genr xdacc = xdev

smpl 3 17054

genr xdacc = xdacc(-1) + xdev

smpl 2 17054

scalar rstop = @max(xdacc) - @min(xdacc)

scalar lrvar = @var(x)\*(1 + (2/@obs(x))\*((4/5)\*@cor(x,x(-1)) + (3/5)\*@cor(x,x(-2)) + (2/5)\*@cor(x,x(-3)) + (1/5)\*@cor(x,x(-4))))

scalar rs\_stat = rstop/@sqrt(lrvar\*@obs(x))

GPH estimates of *d* were obtained as the initial values of the coefficient c(2) in the regressions

ls ret c d

ls sq\_ret c d

ls abs\_ret c d

The SACFs in Figure 5.11 may be computed using the program fractdiff\_acf.prg (shown here for abs\_ret)

smpl 1 200

scalar d\_hat = 0.47

scalar se = 0.06

vector (200) rho\_v

for !i = 1 to 200

rho\_v(!i) = (@gamma(1-d\_hat)/@gamma(d\_hat))\*(!i^-se)

next

mtos(rho\_v,rho\_d)