ECON90033 – QUANTITATIVE ANALYSIS OF FINANCE I

TUTORIAL 5

Read this handout and complete the tutorial exercises before your tutorial class so that you can ask for help during the tutorial if necessary.

Deterministic and Stochastic Trends

As we discussed in the week 3 lectures, a stochastic process is said to be (weakly or covariance) nonstationary if its mean and/or autocovariances change in time, and in week 4 we illustrated nonstationary processes with two simple examples.

1) $y_t = \mu_t + \varepsilon_t$, $\mu_t = \alpha_0 + \alpha_1 t + \alpha_2 t^2 + ... + \alpha_v t^v$, $\varepsilon_t : WN(0, \sigma^2)$

{y_t} is the sum of two terms, μ_t and ε_t. {μ_t} is a νth order[1] polynomial in the t time variable. It is a deterministic trend, as it is perfectly predictable from its own past. {ε_t} is the usual white noise error term, so it is stationary.

$E(\varepsilon_t) = 0$, thus $E(y_t) = \mu_t$. This expected value depends on t, so {μ_t} and {y_t} are both nonstationary. Their difference, however, is stationary since,

$$y_t - \mu_t = \varepsilon_t$$

Hence, {y_t} is not stationary, but the deviations from the expected values are. Since {y_t} can be made stationary by subtracting its deterministic trend, it is called a trend-stationary (*TS*) process.

2) $y_t = \varphi_1 y_{t-1} + \varepsilon_t$, ε_t , $\varepsilon_t : WN(0, \sigma^2)$

This is an *AR*(1) process, termed pure *AR*(1) process as it does not have any deterministic term. Whether this process is stationary or not, depends on the φ_1 parameter. Namely, it is stationary if $|\varphi_1| < 1$ and not stationary otherwise. Namely, it is an explosive process and its mean, variance and autocovariances all approach infinity (in absolute value) if $|\varphi_1| > 1$, while the mean is constant, but the variance and autocovariances approach infinity (in absolute value) if $|\varphi_1| = 1$.

The special case,

$$y_t = y_{t-1} + \varepsilon_t$$

is called a random walk or a unit-root process. What do these names actually mean?

[1] ν is small Greek letter *nu*.

The first name, random walk, refers to the fact that, as it will be illustrated in Exercise 1, $\{y_t\}$ follows a path that consists of a succession of random steps, while the second name, unit-root process, refers to the fact that φ_1 is the root of the corresponding characteristic equation.[2]

By backward iteration,

$$y_t = y_{t-1} + \varepsilon_t = y_{t-2} + \varepsilon_{t-1} + \varepsilon_t = y_{t-3} + \varepsilon_{t-2} + \varepsilon_{t-1} + \varepsilon_t = ... = y_0 + \sum_{i=0}^{t-1} \varepsilon_{t-i} = y_0 + \sum_{i=1}^{t} \varepsilon_i$$

Hence, y_t is the sum of the y_0 initial value and the sum of all ε error terms from time 1 to t.

From the previous formula,

$$E(y_t) = y_0 \ \ , \ Var(y_t) = \sum_{i=1}^{t} \sigma^2 = \sigma^2 t$$

so the (unconditional) expected value of y_t is constant, but its unconditional variance is not. Consequently, a random walk or a unit-root process is not stationary. However, its first differences

$$y_t - y_{t-1} = \varepsilon_t$$

is a white noise and thus stationary. Since the first difference of $\{y_t\}$ is stationary, a random walk or a unit-root process is termed difference-stationary.

Assume that the information set, i.e., the collection of all available information, at time t – 1 is $\Omega_{t-1} = (\varepsilon_{t-1}, \varepsilon_{t-2}, ..., \varepsilon_1)$. This means that at time t – 1, y_0 and $\varepsilon_{t-1}, \varepsilon_{t-2}, ..., \varepsilon_1$ are known constants, but ε_t is not, it is yet to be realized. Consequently, the conditional expected value of y_t is

$$E(y_t \mid \Omega_{t-1}) = E(y_0 \mid \Omega_{t-1}) + \sum_{i=1}^{t-1} E(\varepsilon_i \mid \Omega_{t-1}) + E(\varepsilon_t \mid \Omega_{t-1}) = y_0 + \sum_{i=1}^{t-1} \varepsilon_i$$

This shows that the conditional expected value of y_t is driven by the $(\varepsilon_t\}$ stochastic shocks, and the cumulative sum of these shocks, $\Sigma\varepsilon$, is called stochastic trend. The main feature of a stochastic trend is that both its level and slope can and does change randomly.

Exercise 1

The purpose of this exercise is to illustrate the difference between deterministic and stochastic trends.

a) Draw and plot a random sample of 200 normal random numbers, *eps*, with 0 expected value and 5 standard deviation.

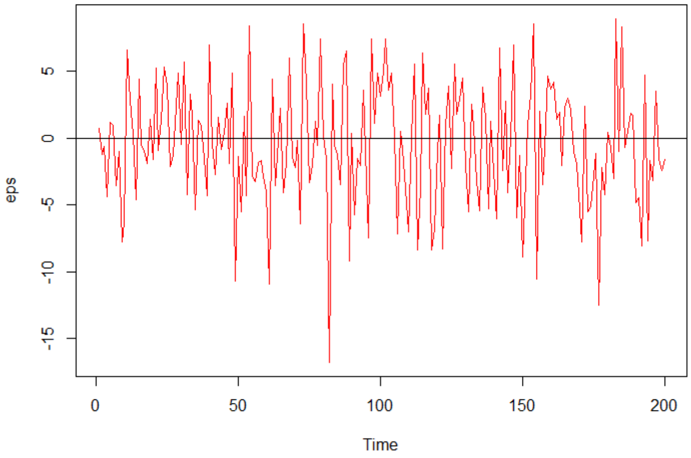
[2] In this course we do not have time to discuss characteristic equations, but if you wish to see some details, have a look at the *Homogeneous Linear Difference Equations* handout.

Launch *RStudio*, create a new project and script, and name both *t5e1*.

Execute the following commands[3]

```
set.seed(6082023)
eps = ts(rnorm(200, mean = 0, sd = 5))
plot(eps, col = "red")
abline(h = 0)
```

to get



b) Use this random series to simulate the following trend stationary process:

$$y_t = 50 - 0.5t + \varepsilon_t$$

We can simulate this process with a *for* loop that has the following syntax:

```
for (var in vector)
   {statement(s)}
```

where var takes on each value of the vector during the loop and in each iteration, the statements are evaluated.

We perform the task in three steps. First we create a time variable,

```
t = ts(1:200)
```

[3] Note that to ensure that we all get the same printouts, we use a common seed again.

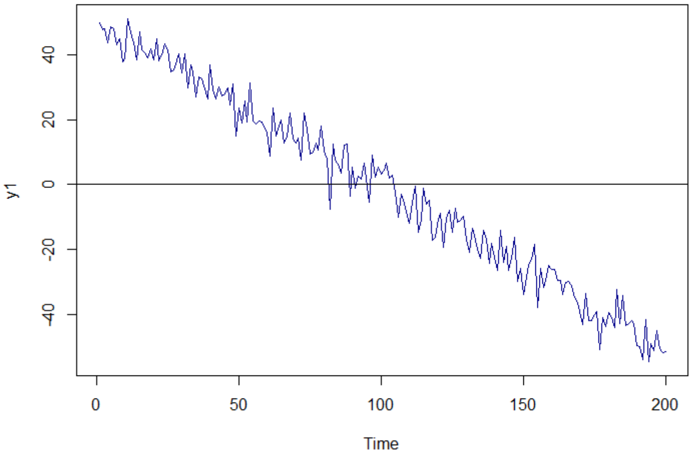
then the *y1* series,

```
y1 = ts(50, start = 1, end = 200)
for (t in 2:200)
  {y1[t] = ts(50 - 0.5*t + eps[t])}
```

and finally plot the simulated series,

```
plot.ts(y1, col = "darkblue")
abline(h = 0)
```

We get



The simulated $\{y_{1t}\}$ series appears to be fluctuating randomly with constant variance around the 50 – 0.5*t* deterministic linear trend.

c) Calculate and plot the cumulative sum of the *eps* random series.

The relevant *R* function is:

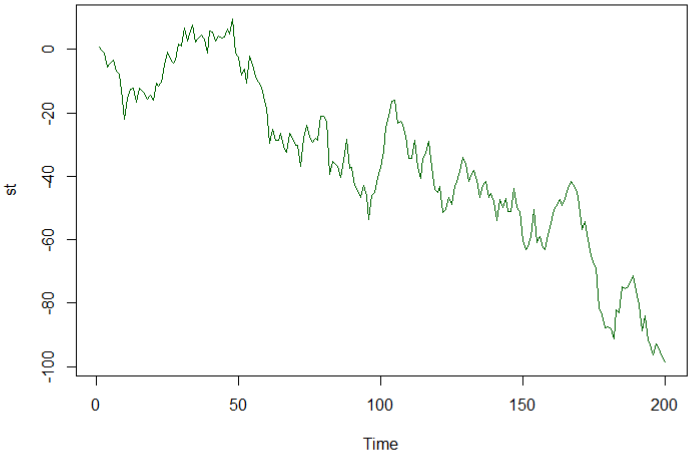
cumsum(x)

returns the cumulative sum of vector x.

Hence, execute

```
st = ts(cumsum(eps))
plot(st, col = "darkgreen")
```

to get



If you did not know how this series was generated, you could think that just like in part (b) it is fluctuating around some downward sloping straight line with constant variance. This is, however, a stochastic trend whose variance ($\sigma^2 t$ = 25*t*) is increasing in time.

d) Use the same random series as before to simulate the following difference stationary process:

$$y_t = -0.5 + y_{t-1} + \varepsilon_t$$

Execute the following commands to generate and plot the simulated series:

```
y2 = ts(50, start = 1, end = 200)
for (t in 2:200)
  {y2[t] = ts(-0.5 + y2[t-1] + eps[t])}
plot(y2, col = "orange")
abline(h = 0)
```

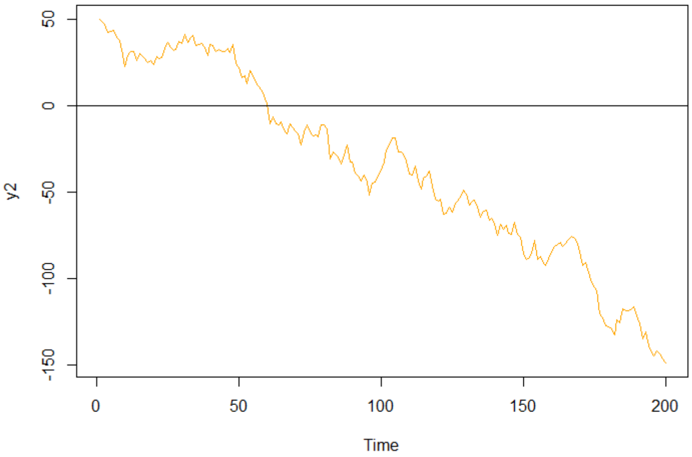
You should get the plot shown on the next page.

Just like in part (c), if you did not know how this series was generated, you could think that it is fluctuating around some downward sloping straight line with constant variance. This series, however, has both a deterministic and a stochastic trend, since by backward iteration,

$$y_t = -0.5 + y_{t-1} + \varepsilon_t = -0.5 + (-0.5 + y_{t-2} + \varepsilon_{t-1}) + \varepsilon_t = -0.5 \times 2 + (-0.5 + y_{t-3} + \varepsilon_{t-2}) + \varepsilon_{t-1} + \varepsilon_t$$

$$= \ldots = -0.5t + y_0 + \sum_{i=1}^{t} \varepsilon_i$$

and due to the stochastic trend, its variance ($\sigma^2 t$ = 25*t*) is increasing in time.



Spurious Regression

In general, spurious or nonsense regression means a statistically significant regression of totally unrelated variables. It typically occurs when highly persistent non-stationary variables are regressed on each other without removing the deterministic or stochastic trends in these variables. The purpose of the first exercise is to illustrate this phenomenon.

Exercise 2

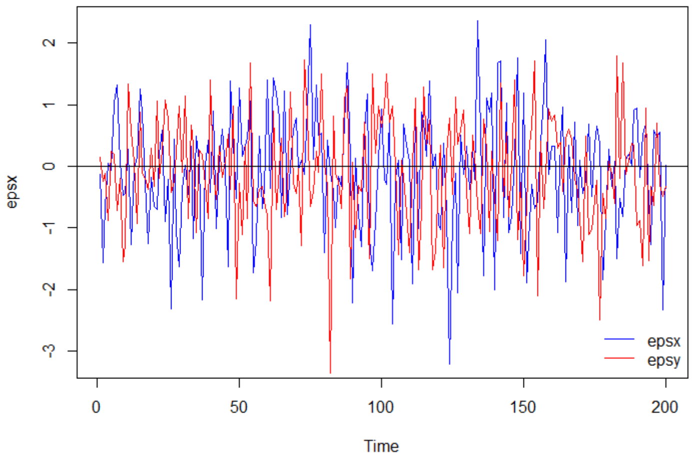
a) Draw two independent random samples of 200 standard normal random numbers, *epsx* and *epsy*, and illustrate them using a time-series plot and a scatter diagram (*epsy* against *epsx*).

Launch *RStudio*, create a new project and script, and name both *t5e2*.

Execute the following commands

```
set.seed(6082023)
epsy = ts(rnorm(200))
epsx = ts(rnorm(200))

plot.ts(epsx, col = "blue")
lines(epsy, col = "red")
legend("bottomright", bty="n", lty=c(1,1), col=c("blue", "red"),
          legend = c("epsx", "epsy"))
abline(h = 0)
```

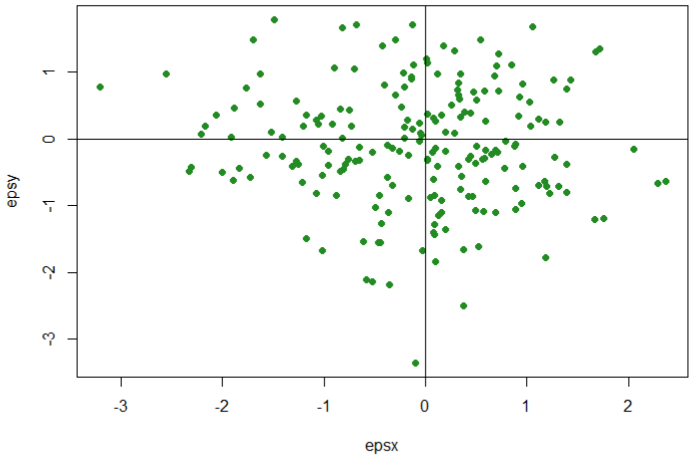
to get the following time series plot:



The scatter plot can be developed, for example, by the *plot.zoo()* function. Execute,

```
library(zoo)
plot.zoo(epsx, epsy, pch = 19, col = "forestgreen")
abline(h = 0)
abline(v = 0)
```

to obtain



On this scatterplot the dots that represent the (*epsx*, *epsy*) pairs of observations seem to be scattered randomly around the origin, i.e., the (0, 0) point. This suggests that they are not related to each other, and that's what one should expect, since *epsx* and *epsy* are two independent random samples drawn from two different populations.

b) Using the two white noise series from part (a), generate two pure random walks, *x1* and *y1*, with zero initial values. Graph them on a time-series plot and on a scatter diagram.

The pure random walks

$$x_t = x_{t-1} + \varepsilon_{x,t} \quad , \quad y_t = y_{t-1} + \varepsilon_{y,t}$$

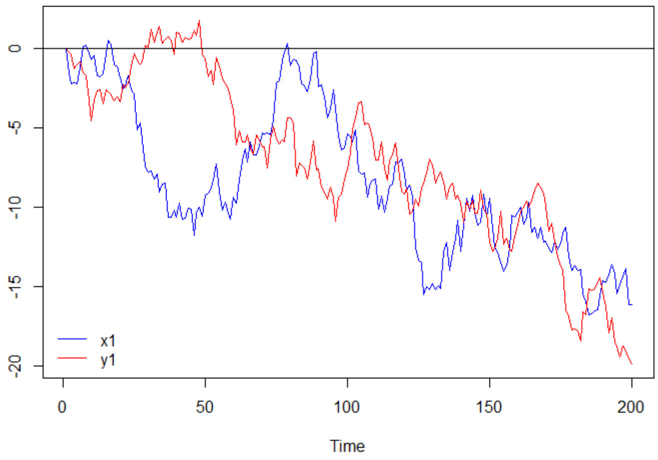
Are generated by the following commands:

```
y1 = ts(0, start = 1, end = 200)
x1 = ts(0, start = 1, end = 200)
for (t in 2:200) {
  y1[t] = ts(y1[t-1] + epsy[t])
  x1[t] = ts(x1[t-1] + epsx[t])
}
```

To plot them, execute

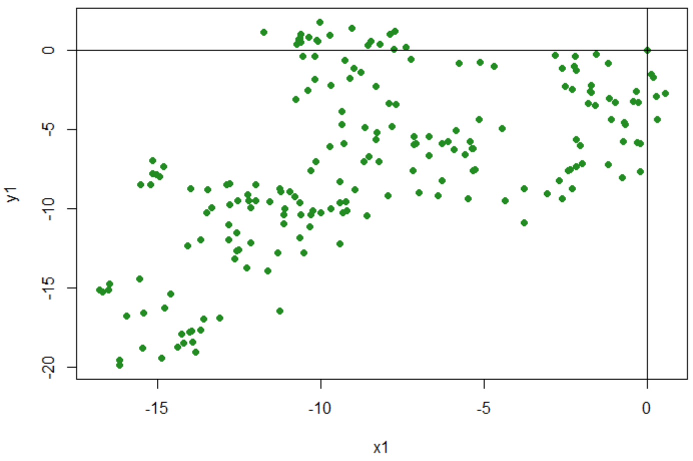
```
ts.plot(x1, y1, col = c("blue", "red"))
legend("bottomleft", bty="n", lty=c(1,1), col=c("blue", "red"),
     legend = c("x1", "y1"))
abline(h = 0)

plot.zoo(x1, y1, pch = 19, col = "forestgreen")
abline(h = 0)
abline(v = 0)
```

The time series plot



illustrates the stochastic trends that the simulated random walks have.

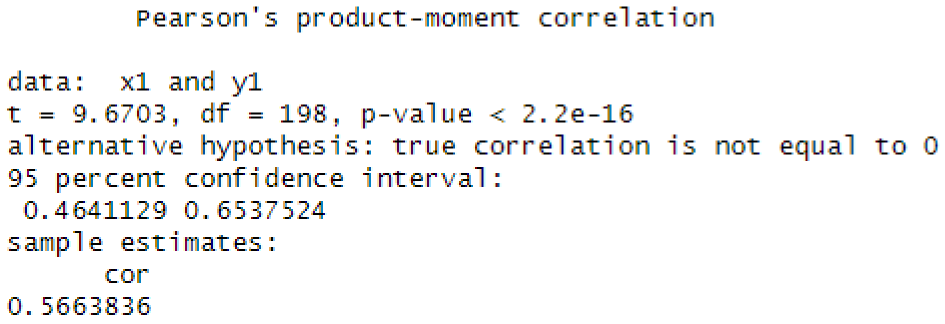
The scatter plot



suggests that there is some positive correlation between the two simulated random walks.

This observation is confirmed by the (Pearson) correlation coefficient and the *t*-test on it:

cor.test(x1, y1)



```
        Pearson's product-moment correlation

data:  x1 and y1
t = 9.6703, df = 198, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.4641129 0.6537524
sample estimates:
      cor
0.5663836
```

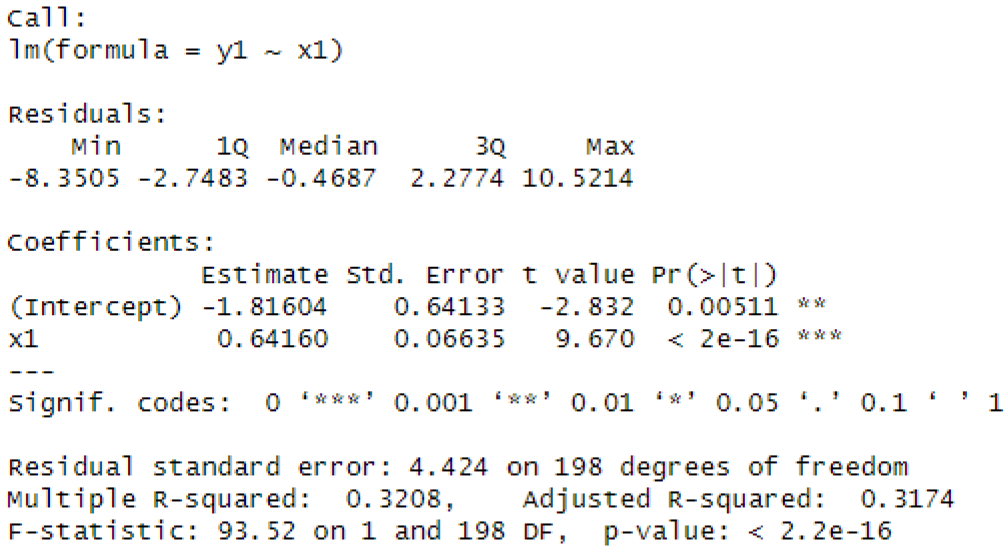
The sample correlation coefficient, 0.5663836, is positive and half of the reported *p*-value of the *t*-test for the null hypothesis of no correlation against the alternative hypothesis of nonzero correlation is $2.2\text{x}10^{-16} / 2 = 1.1\text{x}10^{-16} \approx 0$. Hence, we can reject H_0 and conclude that there is significantly positive linear relationship between the two simulated random walks.

c) Estimate this relationship by regressing *y1* on a constant and *x1*.

The

m1 = lm(y1 ~ x1)
summary(m1)

commands return the following printout:



```
Call:
lm(formula = y1 ~ x1)

Residuals:
    Min      1Q  Median      3Q     Max 
-8.3505 -2.7483 -0.4687  2.2774 10.5214 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -1.81604    0.64133  -2.832  0.00511 ** 
x1           0.64160    0.06635   9.670  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.424 on 198 degrees of freedom
Multiple R-squared:  0.3208,    Adjusted R-squared:  0.3174 
F-statistic: 93.52 on 1 and 198 DF,  p-value: < 2.2e-16
```

It shows that this simple linear regression is significant (*F-statistic* = 93.52, *p-value* ≈ 0) and it accounts for about 32% of the total variation of *y1* in this sample (*Multiple R-squared* = 0.3208).

This regression, however, is spurious because *x1* and *y1* had been generated completely independently of each other. This does not mean that the correlation detected between the two series does not exist, they are indeed correlated with each other because the two random walks behind them are correlated.[4] It just means that the revealed relationship is a purely statistical relationship, a co-movement, and it does not reflect any real relationship between the two variables.

d) Simulate two random walks with drift, *x2* and *y2*, using the same *epsx* and *epsy* series as before, assuming that the drift term is 0.5 for *x2* and -0.3 for *y2*, and that the initial values are zero. Regress *y2* on a constant and *x2*. What do you notice?

You need to execute similar commands as before. Namely,

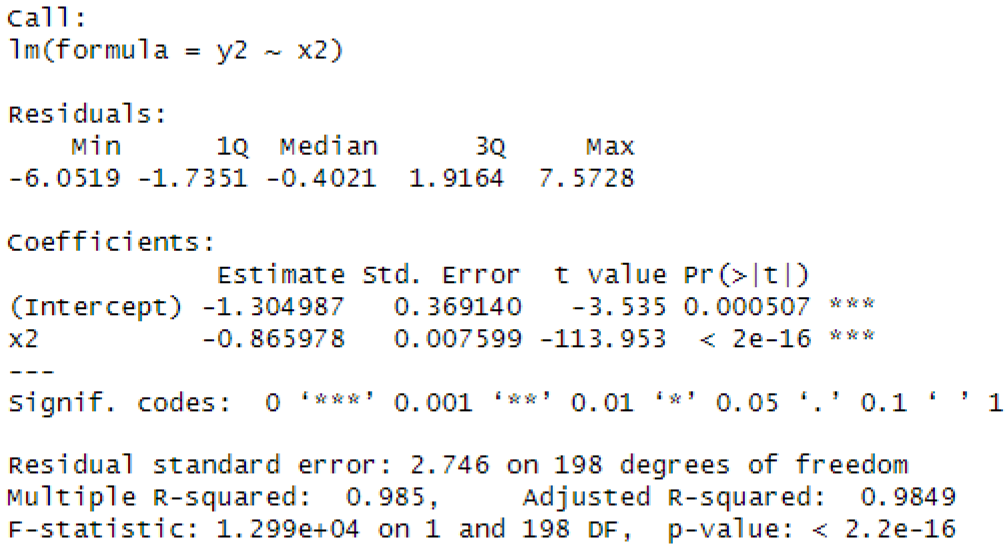
```
y2 = ts(0, start = 1, end = 200)
x2 = ts(0, start = 1, end = 200)
for (t in 2:200) {
  y2[t] = ts(-0.3 + y2[t-1] + epsy[t])
  x2[t] = ts(0.5 + x2[t-1] + epsx[t])
}
```

generate the two random walks, and

[4] Two trends, deterministic or stochastic, are always correlated with each other to some degree, because any change in one of them is accompanied with a somewhat systematic change in the other.

m2 = lm(y2 ~ x2)
summary(m2)

regress *y2* on a constant and *x2*, and return the following printout:



```
Call:
lm(formula = y2 ~ x2)

Residuals:
    Min      1Q  Median      3Q     Max 
-6.0519 -1.7351 -0.4021  1.9164  7.5728 

Coefficients:
             Estimate Std. Error  t value Pr(>|t|)    
(Intercept) -1.304987   0.369140   -3.535 0.000507 ***
x2          -0.865978   0.007599 -113.953  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.746 on 198 degrees of freedom
Multiple R-squared:  0.985,	Adjusted R-squared:  0.9849 
F-statistic: 1.299e+04 on 1 and 198 DF,  p-value: < 2.2e-16
```

This regression is also significant, and it explains about 98.5% of the total sample variations of *y2*. Nevertheless, this is again a spurious regression as the extremely good fit to the data is due entirely to the correlation between the two random walks.

But why is R^2 so much higher this time? As we saw in part (d) of Exercise 1, due to the constant terms (also called drift terms), the random walks have both deterministic and stochastic trends. Namely,

$$y_{2t} = -0.3 + y_{2,t-1} + \varepsilon_{yt} = -0.3t + y_0 + \sum_{i=1}^{t} \varepsilon_{y,i} \quad , \quad x_{2t} = 0.5 + x_{2,t-1} + \varepsilon_{xt} = 0.5t + x_0 + \sum_{i=1}^{t} \varepsilon_{x,i}$$

In part (b) we had no intercept terms, so the two random walks had only stochastic trends, $\Sigma\varepsilon_{y,i}$ and $\Sigma\varepsilon_{x,I}$, which can be strongly, but definitely imperfectly correlated with each other. This time, however, due to the intercept terms the random walks also have deterministic trends, -0.3*t* and 0.5*t*, which are perfectly correlated with each other.

This exercise illustrated the importance of carefully studying time series data before using them to estimate a regression model. In particular, it is important to find out whether the stochastic processes that generated the sample data are stationary, trend stationary or difference stationary. This can be done by performing tests for a unit root versus stationarity.

Testing for a Unit Root / Stationarity

There are many different tests for a unit root versus stationarity, but none of them is superior in all applications. For this reason, in practice one should always use several alternative tests. In this course, due to lack of time, we consider only two tests, the Dickey-Fuller (*DF*) τ test for a unit root and the Kwiatkowski-Phillips-Schmidt-Shin (*KPSS*) test for stationarity.

DF τ test

The test regression is an *AR*(1) regression augmented with an intercept and linear trend,

$$y_t = a_0 + \varphi_1 y_{t-1} + a_2 t + \varepsilon_t \qquad \text{or equivalently} \qquad \Delta y_t = a_0 + \gamma y_{t-1} + a_2 t + \varepsilon_t \;\; , \;\; \gamma = \varphi_1 - 1$$

and depending on the a_0 and a_2 parameters, we distinguish three possibilities:

Model 1: $a_0 = a_2 = 0$

Model 2: $a_0 \neq 0$, $a_2 = 0$

Model 3: $a_0 \neq 0$, $a_2 \neq 0$

In each case ε_t is supposed to be a white noise error.

The hypotheses can be given either in terms of the φ_1 slope parameter, or in terms of γ. Namely,

$$H_0 : \varphi_1 = 1 \quad vs \quad H_A : |\varphi_1| < 1 \qquad \text{or} \qquad H_0 :\; \gamma = 0 \quad vs \quad H_A :\; \gamma < 0$$

Under the null hypothesis the characteristic root of the *AR*(1) process is equal to one, so there is a unit root, and the process is not stationary; while under the alternative hypothesis the characteristic root is between -1 and 1, hence the process is stationary.[5]

No matter which model is estimated, the *DF* τ test statistic is always the *t*-ratio of the slope, i.e.,

$$\tau = \frac{\hat{\varphi}_1 - 1}{s_{\hat{\varphi}_1}} = \frac{\hat{\gamma}}{s_{\hat{\varphi}_1}}$$

but it is denoted as τ for Model 1, τ_μ for Model 2 and τ_τ for Model 3, and these three test statistics have different sampling distributions.

The proper specification of the test regression can be crucial because a missing deterministic term, intercept or trend, might prevent the test to reject the null hypothesis even if it is false.

In addition to the deterministic terms, the other major issue is the assumption that the ε_t error term in the *DF* test regression is a white noise, in particular it is serially uncorrelated. This requirement is definitely violated when the data generating process has a higher order *AR*

[5] Note that stationarity also requires -2 < γ, a requirement not checked by the *DF* test. In practice, however, it is almost always satisfied, and if it were not, a simple time series plot would most likely reveal that.

component, i.e., *AR*(*p*), where *p* > 2. In that case, the test statistic must be augmented with sufficiently long lags of the first difference of the variable, i.e.,

$$\Delta y_t = \gamma y_{t-1} + \sum_{i=2}^{k} \beta_i \Delta y_{t-i+1} + \varepsilon_t \quad , \quad k > p$$

leading to the augmented *DF*, i.e., *ADF* test.

In practice, *p* is usually unknown, and *k* is selected on the basis of some model selection criterion. However, no matter which criterion is used, it is important to check whether the residuals from the test regression are serially uncorrelated, as required by the *ADF* test.

The *ADF* test can be performed in several ways in *R*. One option is the

```
ur.df(y, type = "none" or “drift” or “trend, selectlags = "AIC" or "BIC" or "Fixed)
```

from the urca package, where y is the time series to be tested for a unit root, type refers to the deterministic term(s) in the test regression and it can be "none" (Model 1), "drift" (Model 2) or "trend" (Model 3), and selectlags specifies whether the lag length is selected by minimizing “AIC” or “BIC”, or it is set equal to some "Fixed" value.

Exercise 3

In Exercise 1 of tutorial 3 you estimated CAPM for Exxon and IBM. In the first regression the dependent and independent variables were the excess returns on the Exxon stocks and on the whole market (S&P 500), respectively. In order to see whether that regression is spurious, in this exercise you will subject the two variables to the *ADF* test.

a) Launch *RStudio*, create a new project and script, and name both *t5e3*. Open the data from the *t3e1.RData* file, resave it in the new project as *t5e3.RData*, and attach it to your project.

   Calculate the two excess return series as you did in tutorial 3, i.e., by executing the following commands:

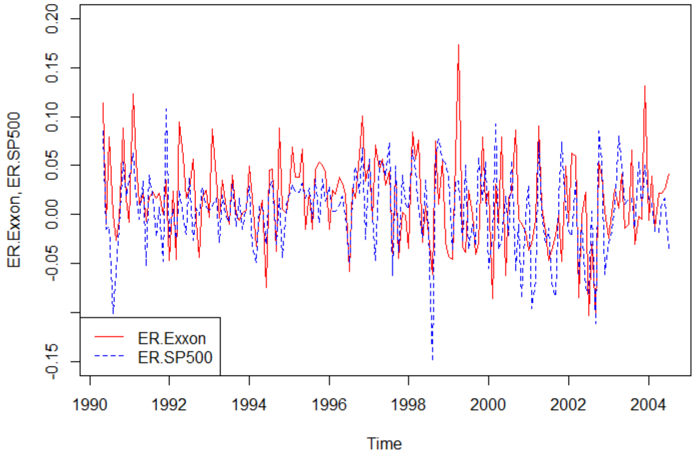
```
attach(t3e1)
Exxon = ts(Exxon, start = c(1990, 4), end = c(2004, 7),
        frequency = 12)
R.Exxon = Exxon / lag(Exxon,-1) - 1
SP500 = ts(SP500, start = c(1990, 4), end = c(2004, 7),
        frequency = 12)
R.SP500 = SP500 / lag(SP500,-1) - 1
Tbill = ts(Tbill, start = c(1990, 4), end = c(2004, 7),
        frequency = 12)
R.Tbill = Tbill / 12
```

```
ER.Exxon = R.Exxon - R.Tbill
ER.SP500 = R.SP500 - R.Tbill
```

b) Test the excess return series for a unit root with the *ur.df* function. Select the lag length by minimizing *BIC*.

```
plot.ts(ER.Exxon, ylab = "ER.Exxon, ER.SP500", col = "red", ylim = c(-0.15, 0.20))
lines(ER.SP500, col = "blue", lty = 2)
legend("bottomleft", legend = c("ER.Exxon", "ER.SP500"),
          col=c("red", "blue"), lty=1:2)
```

return the following time series plot:



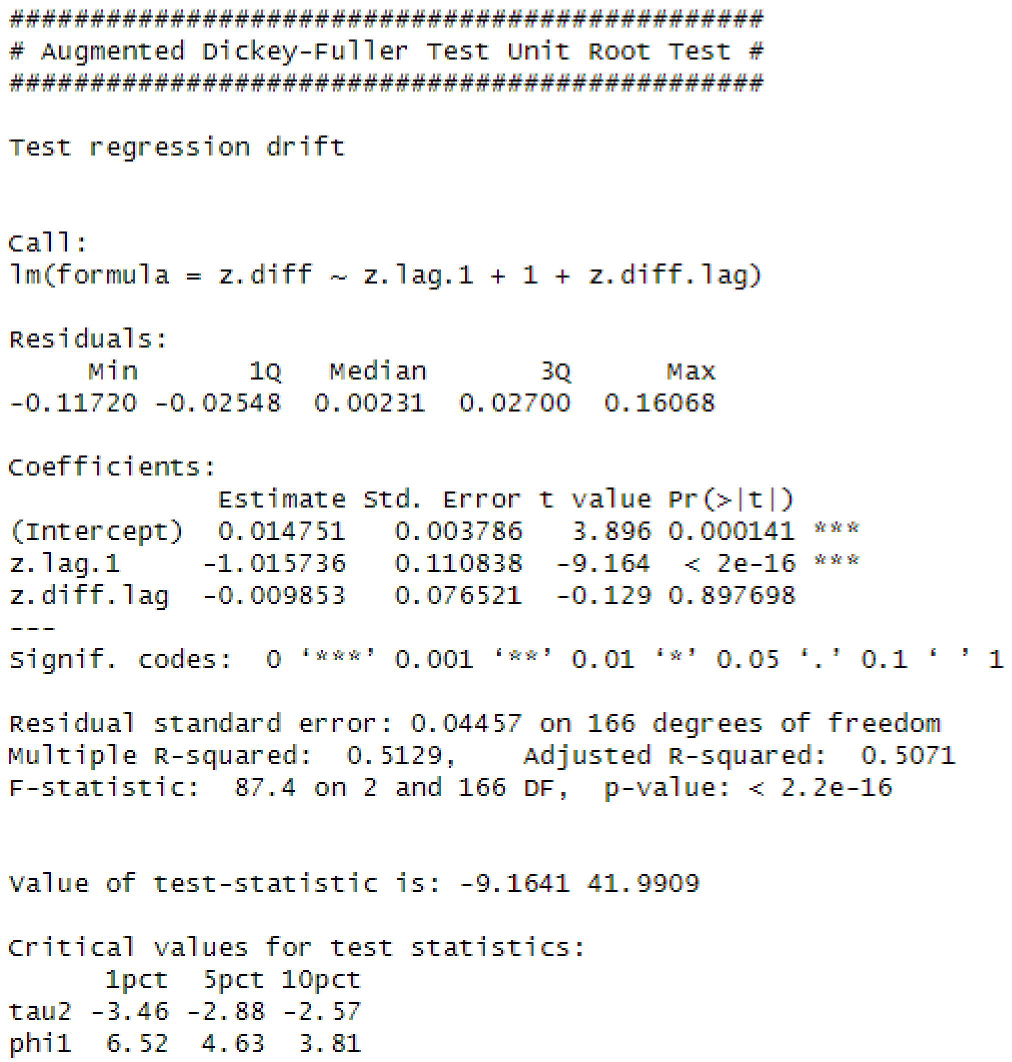
It shows that neither series has a trend, they rather fluctuate around some constants. Since these constants might be different from zero, it is better to include an intercept in the test regression, i.e., to estimate Model 2 rather than Model 1.

To perform the *ADF* test on *ER.Exxon*, execute the following commands:

```
library(urca)
DF.ER.Exxon = ur.df(ER.Exxon, type = "drift", selectlags = "BIC")
summary(DF.ER.Exxon)
```

The first command calls the *urca* package, the second runs the ADF test and saves it as *DF.ER.Exxon*, and the third displays the test regression and the test results. This printout is shown on the next page.

On this printout the variable to be tested is denoted as *z*. In this case it is *ER.Exxo*. Accordingly, *z.diff* is the first difference of *ER.Exxo*, *z.lag.1* is *ER.Exxo* lagged by one and *z.diff.lag* is the lagged first difference of *ER.Exxo*.



```
###############################################
# Augmented Dickey-Fuller Test Unit Root Test #
###############################################

Test regression drift


Call:
lm(formula = z.diff ~ z.lag.1 + 1 + z.diff.lag)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.11720 -0.02548  0.00231  0.02700  0.16068 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept)  0.014751   0.003786   3.896 0.000141 ***
z.lag.1     -1.015736   0.110838  -9.164  < 2e-16 ***
z.diff.lag  -0.009853   0.076521  -0.129 0.897698    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.04457 on 166 degrees of freedom
Multiple R-squared:  0.5129,	Adjusted R-squared:  0.5071 
F-statistic:  87.4 on 2 and 166 DF,  p-value: < 2.2e-16


Value of test-statistic is: -9.1641 41.9909 

Critical values for test statistics: 
      1pct  5pct 10pct
tau2 -3.46 -2.88 -2.57
phi1  6.52  4.63  3.81
```

Hence, using the y_t = *ER.Exxon$_t$* notation, the estimated test regression is

$$\widehat{\Delta y}_t = \hat{a}_0 + \hat{\gamma} y_{t-1} + \hat{\beta}_1 \Delta y_{t-1} = 0.014751 - 1.015736 y_{t-1} - 0.009853 \Delta y_{t-1}$$

It shows that based on *BIC*, the *ur.df* function performed the *ADF* test with a single lag.

Below the test regression you can see two test statistic values. This is because *ur.df* actually performs two tests when the test regression is Model 2. The first (*tau2*) is the *ADF* τ_μ test, and the second (*phi1*) is a joint test for the null hypothesis that there is a unit root but no intercept. In this course we do not have time to learn about the second test, so just ignore it.

The first test statistic value is -9.1641 and, as you can see on the printout that, it is the usual *t* statistic for the null hypothesis of $\gamma = 0$. However, ignore the corresponding *p*-value on the printout (< *2e-16*) because it is based on a *t* distribution, not on the sampling distribution of the *ADF* τ_μ test statistic.

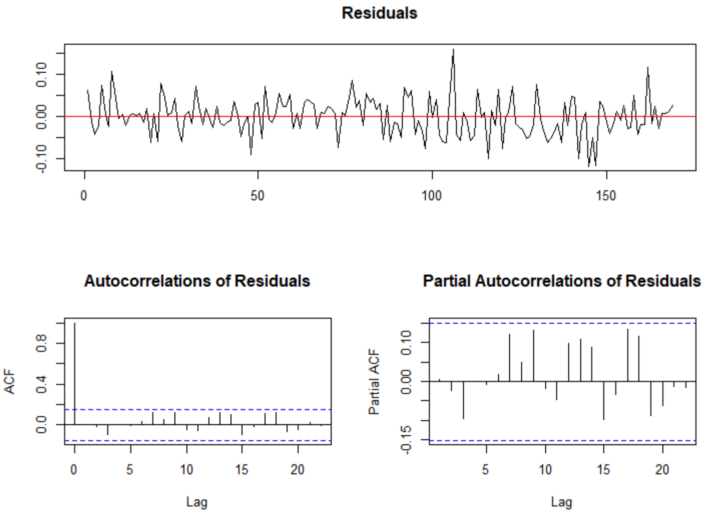
Instead, compare this test statistic value to the *tau2* critical values in the bottom part of the printout, recalling that the τ tests are left-tail tests, so the rejection region is left to the relevant critical value.

In this case the observed τ_μ test statistic value (-9.1641) is smaller than the 1% *tau2* critical value (-3.46), so the null hypothesis of a unit root can be rejected at the 1% significance level, implying that *ER.Exxon* is stationary.

This conclusion, however, can be incorrect if the error term in the test regression is autocorrelated. To check whether the error term is serially uncorrelated, execute

plot(DF.ER.Exxon)

It returns the following plot:

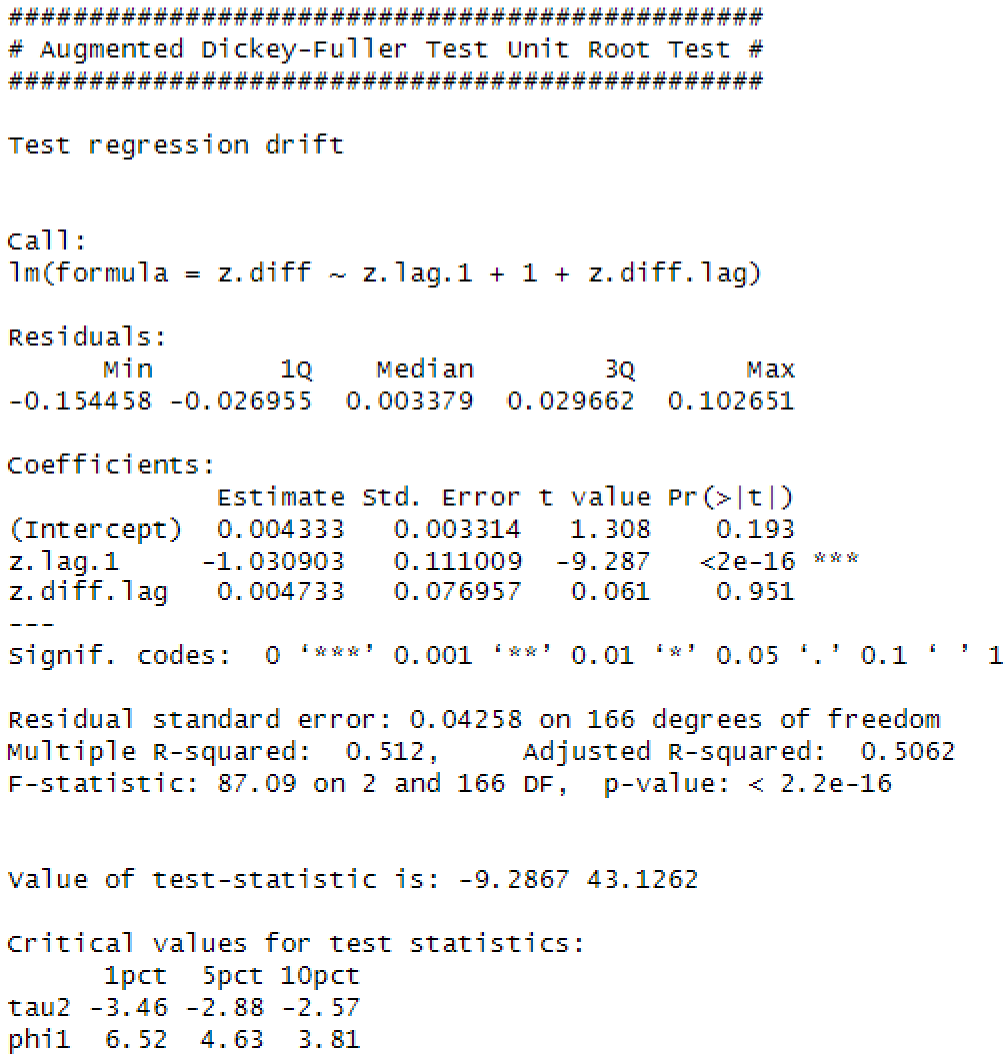


The top panel is a residual plot, and the bottom panel consists of the sample correlograms for the residuals. The residuals look like a random series and on the sample correlograms none of the sample autocorrelation and partial autocorrelation coefficients is significant (for *k* > 0). Hence, there is no reason to assume that the error term in the test regression is autocorrelated and we can rely on the *ADF* test result.

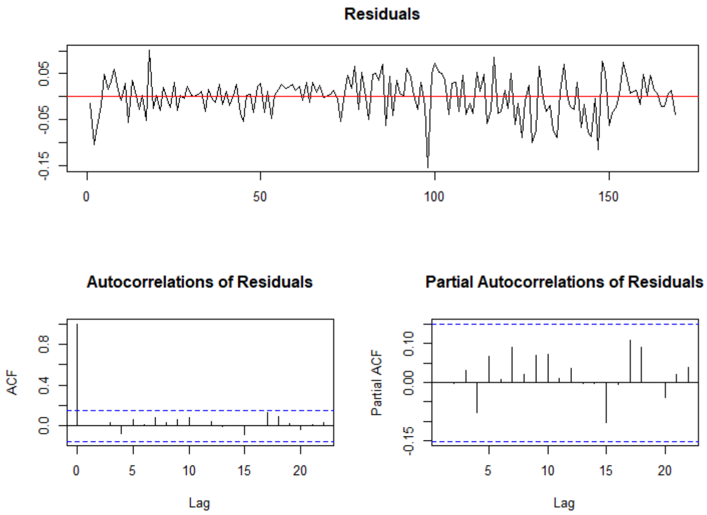
You can perform the *ADF* test on *ER.SP500* similarly. Execute

DF.ER.SP500 = ur.df(ER.SP500, type = "drift", selectlags = "BIC")
summary(DF.ER.SP500)
plot(DF.ER.SP500)

to get the following printouts:



```
###############################################
# Augmented Dickey-Fuller Test Unit Root Test #
###############################################

Test regression drift


Call:
lm(formula = z.diff ~ z.lag.1 + 1 + z.diff.lag)

Residuals:
      Min        1Q    Median        3Q       Max 
-0.154458 -0.026955  0.003379  0.029662  0.102651 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept)  0.004333   0.003314   1.308    0.193    
z.lag.1     -1.030903   0.111009  -9.287   <2e-16 ***
z.diff.lag   0.004733   0.076957   0.061    0.951    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.04258 on 166 degrees of freedom
Multiple R-squared:  0.512,	Adjusted R-squared:  0.5062 
F-statistic: 87.09 on 2 and 166 DF,  p-value: < 2.2e-16


Value of test-statistic is: -9.2867 43.1262 

Critical values for test statistics: 
      1pct  5pct 10pct
tau2 -3.46 -2.88 -2.57
phi1  6.52  4.63  3.81
```



Without repeating all the details, these printouts show that

i. There is only one lag in the test regression.

ii. The τ_μ test statistic value (-9.2867) is smaller than the 1% *tau2* critical value (-3.46), so the null hypothesis of a unit root can be rejected at the 1% significance level. This implies that *ER.SP500* might be also stationary.

iii. The residuals are serially uncorrelated, so one lag is sufficient.

Based on these *ADF* test results it is tempting to conclude that both excess return series are stationary. It is important to acknowledge though, that the *ADF* test for a single unit root might incorrectly reject the null hypothesis when there are actually two unit roots and thus two levels of differencing are required to achieve stationarity. For this reason, irrespectively of the outcome of the *ADF* test on the level series, it is recommended to repeat the test on the differenced series as well, because the difference of a stationary variable is also stationary, while the difference of a variable that is nonstationary due to a unit root, might still be nonstationary due to a second unit root.[6]

c) Test the first differences of the excess return series for a unit root with the *ADF* test.

When we performed the *ADF* tests on the level series in part (b), we had an intercept term in the test regression. Without the lagged first difference, it looked like

$$\Delta y_t = a_0 + \gamma y_{t-1} + \varepsilon_t$$

which is equivalent to

$$y_t = a_0 + \varphi_1 y_{t-1} + \varepsilon_t \quad , \quad \varphi_1 = \gamma + 1$$

From this,

$$y_{t-1} = a_0 + \varphi_1 y_{t-2} + \varepsilon_{t-1}$$

so the first difference of y_t is

$$\Delta y_t = y_t - y_{t-1} = \varphi_1 (y_{t-1} - y_{t-2}) + \varepsilon_{t-1} - \varepsilon_{t-2} = \varphi_1 \Delta y_{t-1} + \Delta\varepsilon_{t-1}$$

This shows that as a result of differencing, the constant term drops out.

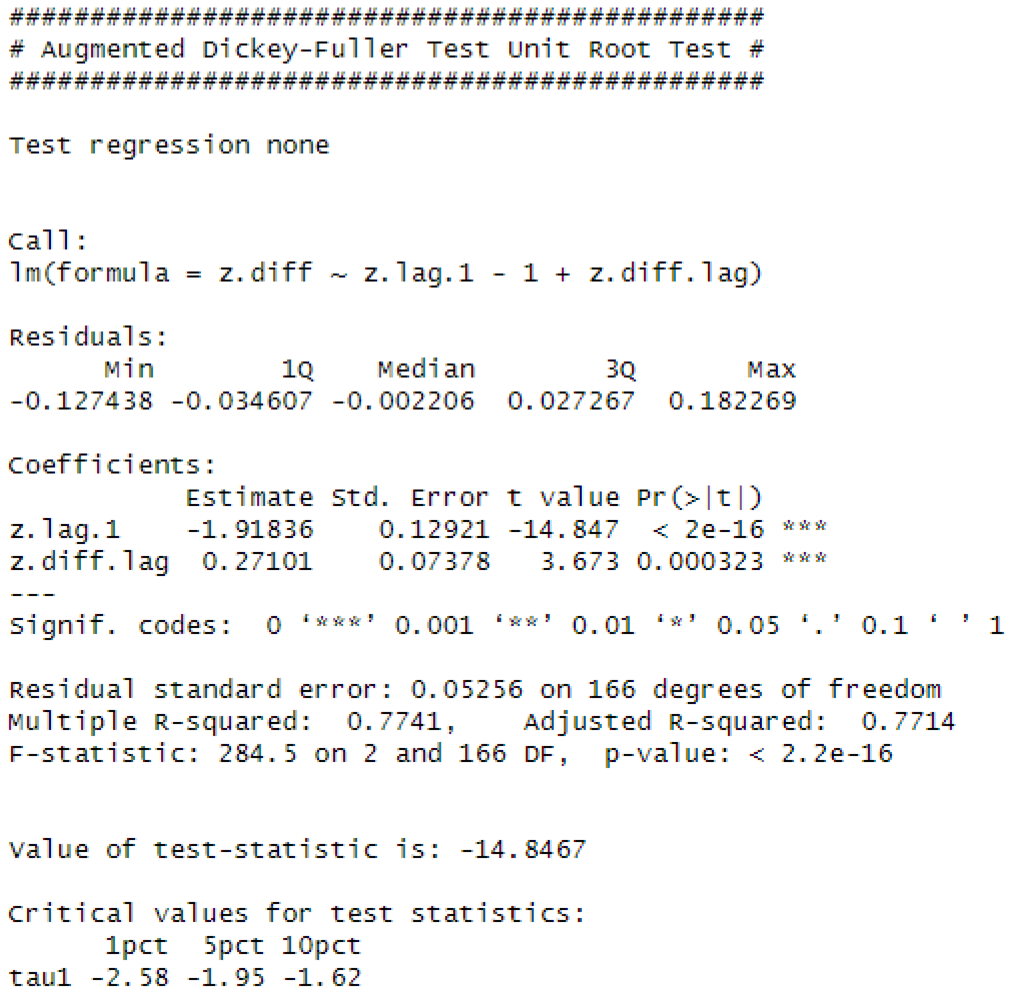
Hence, if we use Model 2 in the *ADF* test on variable *y*, we have to use Model 1 in the *ADF* test on Δy because differencing wipes out the constant. It can be shown similarly, that if we use Model 3 in the *ADF* test on variable *y*, we have to use Model 2 in the *ADF* test on Δy because differencing wipes out the deterministic linear trend.

Otherwise, the *ADF* tests on the first difference of *ER.Exxon* and *ER.SP500* can be performed just like on the level series.

[6] Without going into details, occasionally it might be reasonable to perform further tests on higher differences as well.

DF2.ER.Exxon = ur.df(diff(ER.Exxon), type = "none", selectlags = "BIC")
summary(DF2.ER.Exxon)

return the following printout:



```
###############################################
# Augmented Dickey-Fuller Test Unit Root Test #
###############################################

Test regression none


Call:
lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)

Residuals:
      Min        1Q    Median        3Q       Max 
-0.127438 -0.034607 -0.002206  0.027267  0.182269 

Coefficients:
           Estimate Std. Error t value Pr(>|t|)    
z.lag.1    -1.91836    0.12921 -14.847  < 2e-16 ***
z.diff.lag  0.27101    0.07378   3.673 0.000323 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.05256 on 166 degrees of freedom
Multiple R-squared:  0.7741,	Adjusted R-squared:  0.7714 
F-statistic: 284.5 on 2 and 166 DF,  p-value: < 2.2e-16


Value of test-statistic is: -14.8467 

Critical values for test statistics: 
      1pct  5pct 10pct
tau1 -2.58 -1.95 -1.62
```

On this printout *tau1* is the *DF* τ test statistic. It is equal to -14.8467, smaller than the 1% critical vale, -2.58, so the null hypothesis of a unit root can be rejected at the 1% significance level. This implies that the first difference of *ER.Exxon* is stationary.

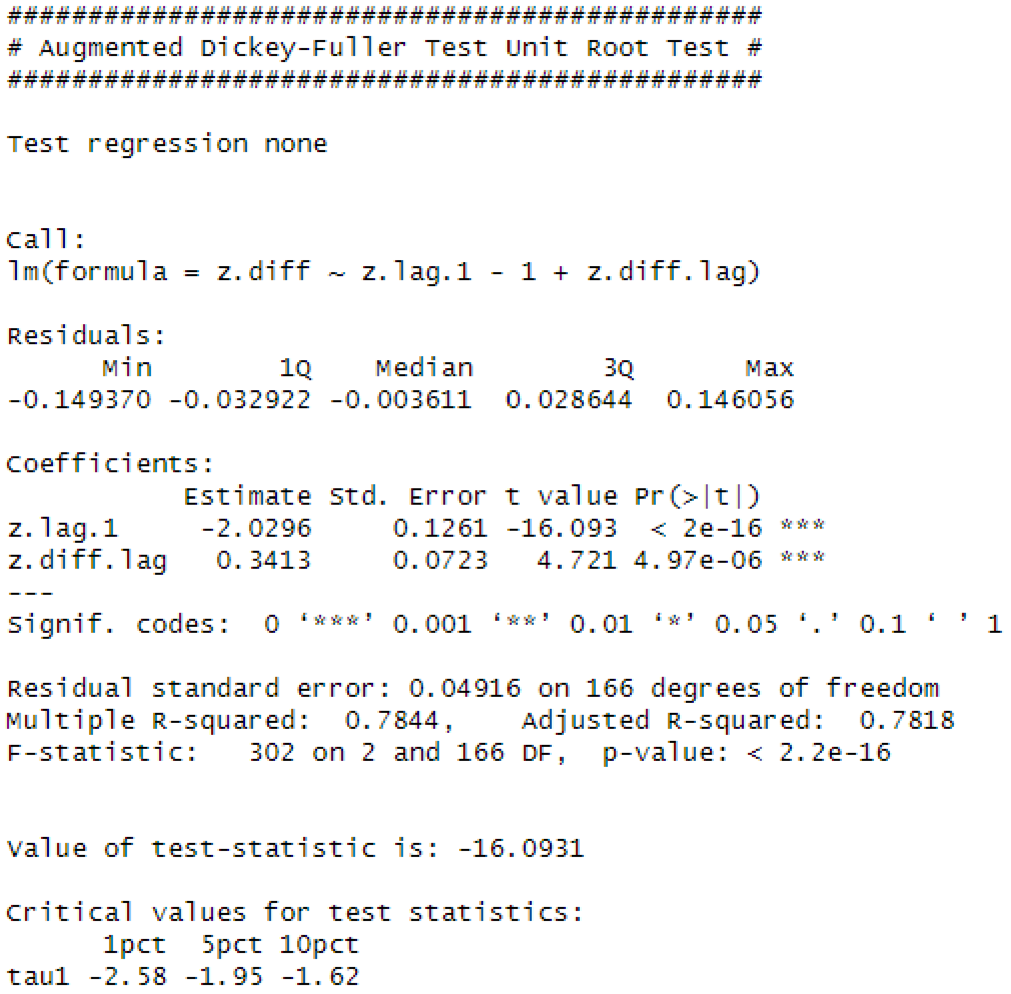
Since the *ADF* test rejected the unit root null hypothesis both for the level and for the first difference of *ER.Exxon*, we conclude that according to this test *ER.Exxon* is stationary.[7]

[7] Note that the first difference of a stationary variable is also stationary. Hence, performing the *ADF* test on the level and on the first difference of a variable, a rejected null hypothesis for the level series followed by a maintained null hypothesis for the differenced series would be a contradictory outcome.

Likewise,

DF2.ER.SP500 = ur.df(diff(ER.SP500), type = "none", selectlags = "BIC")
summary(DF2.ER.SP500)

return



```
###############################################
# Augmented Dickey-Fuller Test Unit Root Test #
###############################################

Test regression none


Call:
lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)

Residuals:
      Min        1Q    Median        3Q       Max 
-0.149370 -0.032922 -0.003611  0.028644  0.146056 

Coefficients:
           Estimate Std. Error t value Pr(>|t|)    
z.lag.1     -2.0296     0.1261 -16.093  < 2e-16 ***
z.diff.lag   0.3413     0.0723   4.721 4.97e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.04916 on 166 degrees of freedom
Multiple R-squared:  0.7844,    Adjusted R-squared:  0.7818 
F-statistic:   302 on 2 and 166 DF,  p-value: < 2.2e-16


Value of test-statistic is: -16.0931 

Critical values for test statistics: 
      1pct  5pct 10pct
tau1 -2.58 -1.95 -1.62
```

The *DF* τ test statistic is *tau1* = -16.0931. It is well below the 1% critical vale, -2.58, so the null hypothesis of a unit root can be rejected at the 1% significance level. All in all, we conclude that *ER.SP500* is stationary.

Since we found both *ER.Exxon* and *ER.SP500* stationary, we can conclude that the CAPM we estimated for Exxon is not spurious.

Consider now the *KPSS* test.

KPSS test

In the *ADF* test, just like in every test based on the classical approach, a rejected null hypothesis is a much stronger outcome than a maintained one because in the former case it is reasonable to assume that the alternative hypothesis is true, i.e., the data generating process is stationary, but in the latter case it is a mistake to conclude that the null hypothesis is true, i.e., the data generating process is a random walk and hence it is not stationary. For this reason, it is a good idea to complement the *ADF* test with the *KPSS* test, where the null and alternative hypotheses are swapped.

We do not discuss the details of this test; it is enough to keep in mind two things about the *KPSS* test. First, in this case the hypotheses are H_0: stationarity and H_A: unit root; and second, in terms of the deterministic terms, there are only two versions, Model 2 and Model 3.

Using *R*, the *KPSS* test can be performed with the following function from the *urca* library:

```
ur.kpss(y, type = "mu" or "tau", lags = "short" or "long" or "nil")
```

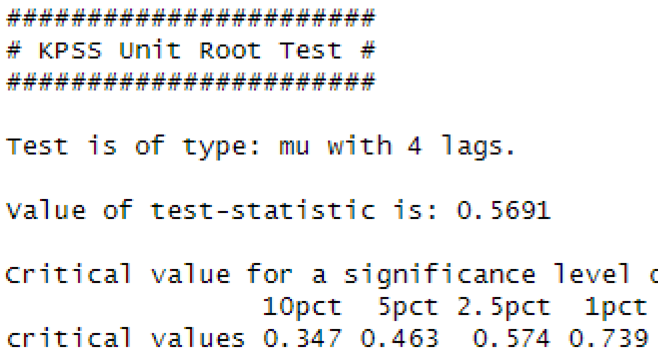
where y is the time series to be tested, type refers to the deterministic term(s) in the test regression and it can be "mu" (Model 2) or "tau" (Model 3), and lags specifies whether the lag length is zero ("nil") or "short" = $(4T/100)^{0.25}$ or "long" = $(12T/100)^{0.25}$. Alternatively, one can specify a user specified number of lags by the optional use.lag argument.

d) Perform the *KPSS* test on the two excess return series.

Like in the *ADF* tests on the level series, we again use Model 2, and perform the *KPSS* test on *ER.Exxon* by executing the following commands:

```
KPSS.ER.Exxon = ur.kpss(ER.Exxon, type = "mu", lags = "short")
summary(KPSS.ER.Exxon)
```

They return this printout:



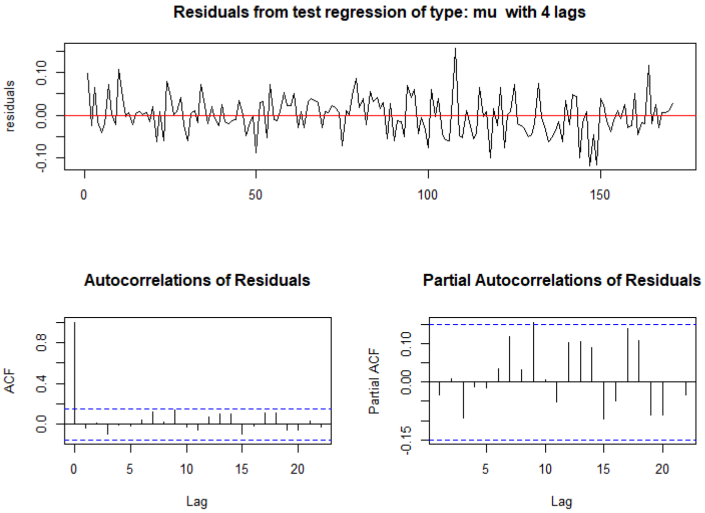
```
#######################
# KPSS Unit Root Test #
#######################

Test is of type: mu with 4 lags.

Value of test-statistic is: 0.5691

Critical value for a significance level of:
                10pct  5pct 2.5pct  1pct
critical values 0.347 0.463  0.574 0.739
```

The observed test statistic value is 0.5691. If you compare the reported critical values to each other, you can see that they are all positive and the 10% critical value is the smallest and the 1% critical value is the largest. This suggests that the rejection region is right to the respective critical value. Since the test statistic (0.5691) is between the 5% critical value (0.463) and the 2.5% critical value (0.574), the *p*-value of the test is smaller than 0.05 but larger than 0.025, so the null hypothesis is rejected at the 5% significance level but maintained at the 2.5% level.

Just like after the *ADF* test, we can again plot the residuals and their correlograms to check whether the error term is likely a white noise by executing the

plot(KPSS.ER.Exxon)

command. It displays



This residual plot and the two correlograms look fine, so we can accept the *KPSS* test result. At the 2.5% or stronger significance level it supports the *ADF* test result since both imply stationarity. At the 5% significance level, however, the *ADF* and *KPSS* test results go against each other since the former test implies stationarity while the second implies a unit root.

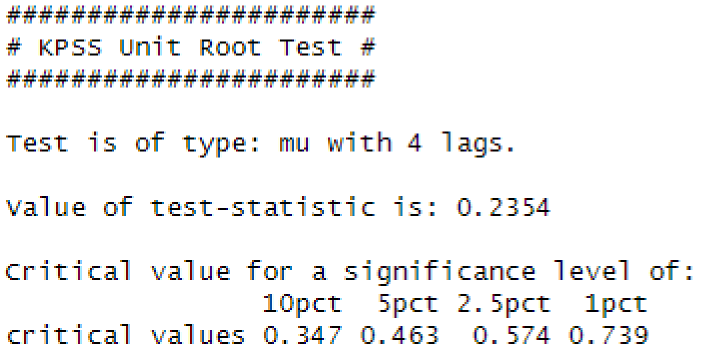
Unfortunately, similar contradictions occur in practice time to time. What can one do in a situation like this one? One option is to perform further unit root / stationarity tests, but in this course we have no time to explore this path further. The other option is to consider the consequences of incorrectly concluding stationarity or a unit root, and choosing the less harmful of the two.

Although an incorrect conclusion or decision is never desirable, in general, when there is uncertainty, it is better to assume that there is a unit root, when in reality there is none, and attempt to eliminate it by differencing, than falsely assuming stationarity.

Perform now the *KPSS* test on *ER.SP500* by executing the following commands:

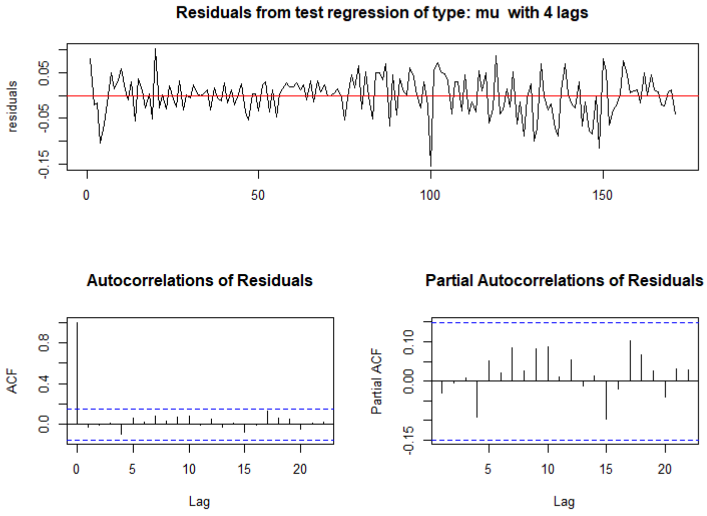
KPSS.ER.SP500 = ur.kpss(ER.SP500, type = "mu", lags = "short")
summary(KPSS.ER.SP500)
plot(KPSS.ER.SP500)

You should get



```
#######################
# KPSS Unit Root Test #
#######################

Test is of type: mu with 4 lags.

Value of test-statistic is: 0.2354

Critical value for a significance level of:
                10pct  5pct 2.5pct  1pct
critical values 0.347 0.463  0.574 0.739
```

and



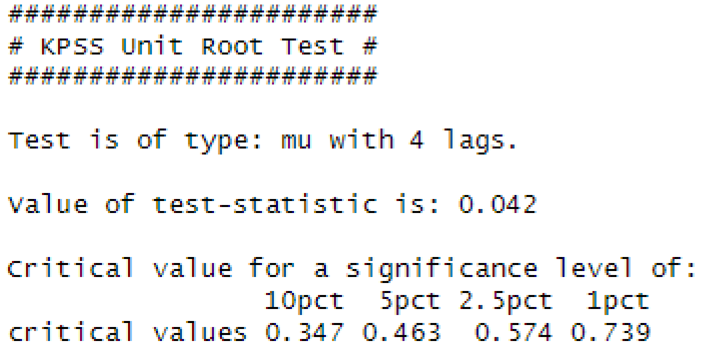
This time the test statistic is 0.2354, smaller than even the 10% critical value, so the *KPSS* test maintains the null hypothesis of stationarity supporting the *ADF* test result. In addition, the residual plot and the correlograms look fine, they do not cast doubt on the *KPSS* test.

e) Test the first differences of the excess return series for stationarity with the *KPSS* test.

Since in the *KPSS* test regression we always have to use at least a constant, we keep using Model 2.

```
KPSS2.ER.Exxon = ur.kpss(diff(ER.Exxon), type = "mu", lags = "short")
summary(KPSS2.ER.Exxon)
```

produce

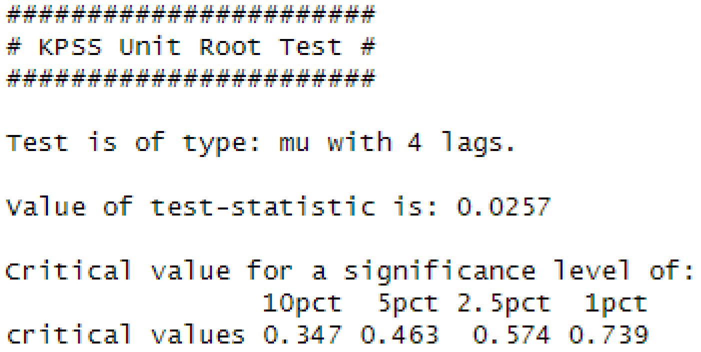


```
#######################
# KPSS Unit Root Test #
#######################

Test is of type: mu with 4 lags.

Value of test-statistic is: 0.042

Critical value for a significance level of:
                10pct  5pct 2.5pct  1pct
critical values 0.347 0.463  0.574 0.739
```

and

```
KPSS2.ER.SP500 = ur.kpss(diff(ER.SP500), type = "mu", lags = "short")
summary(KPSS2.ER.SP500)
```

produce



```
#######################
# KPSS Unit Root Test #
#######################

Test is of type: mu with 4 lags.

Value of test-statistic is: 0.0257

Critical value for a significance level of:
                10pct  5pct 2.5pct  1pct
critical values 0.347 0.463  0.574 0.739
```

As you can see, on both printouts the observed test statistic value (0.0420 and 0.0257, respectively) is smaller that the 10% critical value (0.347), maintaining the null hypothesis of stationarity. This suggests that the first differences of the two excess return series are stationary.

This is reassuring since in the *ADF* and *KPSS* tests alike, a stationary outcome on the level series and a unit-root outcome on the differenced series would be incomprehensible, because the first difference of any random walk that has only one unit root and the first difference of any stationary series are always both stationary.

Finally, returning to the CAPM model, just a brief remark. In spite of the potentially (i.e., depending on the preferred significance level) contradicting *ADF* and *KPSS* tests results on the level of *ER.Exxon*, we can be quite confident that the CAPM regression is not spurious.

The reason for this is that spurious regressions in general fit to the data reasonably well but they suffer from positive first order autocorrelation. If you review pages 6-8 of tutorial 3, you can see that the CAPM model has a modest R^2 and there is not any sign of autocorrelation.

Save your *R* code and quit *RStudio*.