06 Multivariate models: Granger causality, VAR and VECM models

Intuition on Unit roots

Andrius Buteikis, andrius.buteikis@mif.vu.lt
http://web.vu.lt/mif/a.buteikis/

Introduction

We will recap the intuition on how the parameters of an AR process relate to the roots of the AR lag function.

These relations are useful when we want to simulate specific cases, as well as visualize how the data may look like.

Univariate series: $\mathrm{AR}(1)$: $|z| > 1$ (stationary)

Assume that:

$$Y_t = 0.5Y_{t-1} + \epsilon_t$$

Then $\Phi(L) = 1 - 0.5L$. Hence, the root of $1 - 0.5z = 0$ is:

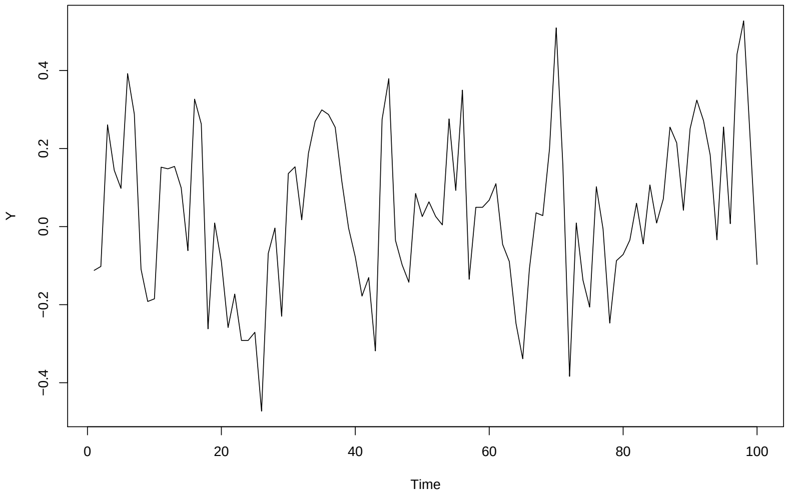
```
roots <- polyroot(c(1, -0.5))
paste0("Absolute: |z| = ", round(abs(roots), 4))
```

```
## [1] "Absolute: |z| = 2"
```

The above **stationary** process highlights the relationship between the root of the autoregression lag function and the coefficient values. We can simulate the series as follows:

```
set.seed(123)
N <- 100
phi <- 0.5
eps <- rnorm(N, mean = 0, sd = 0.2)
Y <- NULL
Y[1] <- eps[1]
for(i in 2:N){
  Y[i] = phi * Y[i - 1] + eps[i]
}
```

```r
plot.ts(Y)
```



Univariate series: AR(1): $|z| < 1$

Assume that:

$$Y_t = 1.1Y_{t-1} + \epsilon_t$$

Then $\Phi(L) = 1 - 1.1L$. Hence, the root of $1 - 1.1z = 0$ is:

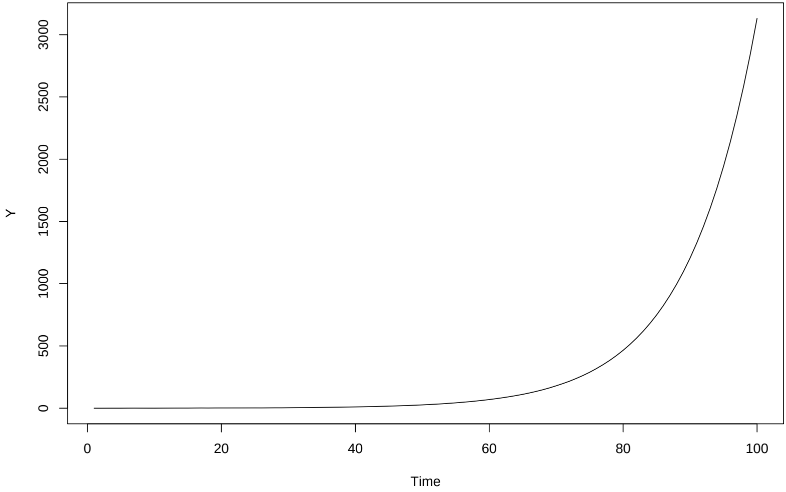
```
roots <- polyroot(c(1, -1.1))
paste0("Absolute: |z| = ", round(abs(roots), 4))
```

```
## [1] "Absolute: |z| = 0.9091"
```

The above **non-stationary** process highlights the relationship between the root of the autoregression lag function and the coefficient values. We can simulate the series as follows:

```
set.seed(123)
N <- 100
phi <- 1.1
eps <- rnorm(N, mean = 0, sd = 0.2)
Y <- NULL
Y[1] <- eps[1]
for(i in 2:N){
  Y[i] = phi * Y[i - 1] + eps[i]
}
```

```r
plot.ts(Y)
```



Univariate series: AR(1): $|z| = 1$ (unit root)

Assume that:

$$Y_t = 1 \cdot Y_{t-1} + \epsilon_t$$

Then $\Phi(L) = 1 - 1 \cdot L$. Hence, the root of $1 - 1 \cdot z = 0$ is:

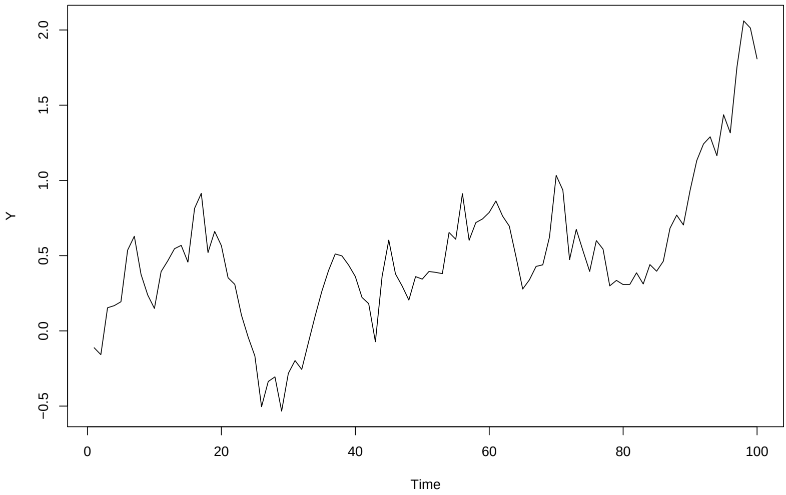
```
roots <- polyroot(c(1, -1))
paste0("Absolute: |z| = ", round(abs(roots), 4))
```

```
## [1] "Absolute: |z| = 1"
```

The above **unit-root** process highlights the relationship between the root of the autoregression lag function and the coefficient values. We can simulate the series as follows:

```
set.seed(123)
N <- 100
phi <- 1
eps <- rnorm(N, mean = 0, sd = 0.2)
Y <- NULL
Y[1] <- eps[1]
for(i in 2:N){
  Y[i] = phi * Y[i - 1] + eps[i]
}
```

```r
plot.ts(Y)
```



Univariate series: $\mathrm{AR}(2)$: $\exists i$: $|z_i| > 1$ (stationary)

Assume that the AR lag function is:

$$\Phi(L) = (1 - 0.5L)(1 - 0.8L) = 1 - 1.3L + 0.4L^2$$

So, our process is:

$$Y_t = 1.3Y_{t-1} - 0.4Y_{t-2} + \epsilon_t$$

the root of $1 - 1.3z + 0.4z^2 = 0$ is:

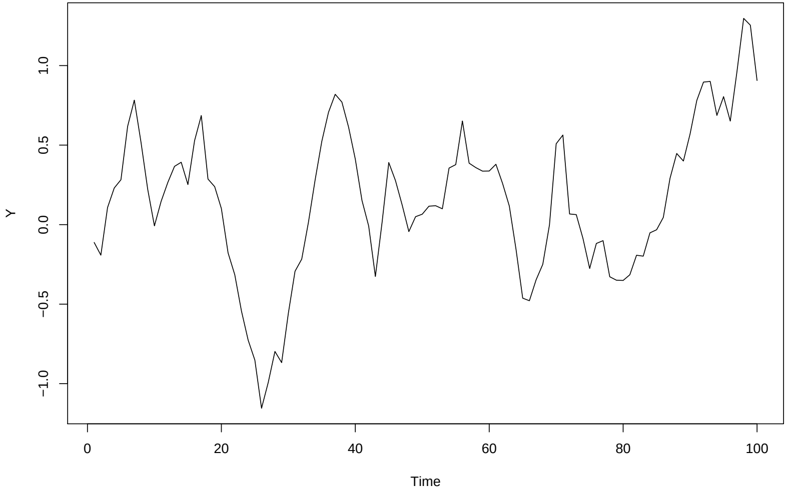
```
roots <- polyroot(c(1, -1.3, 0.4))
paste0("Absolute: |z_i| = ", round(abs(roots), 4))
```

```
## [1] "Absolute: |z_i| = 1.25" "Absolute: |z_i| = 2"
```

The above **stationary** process highlights the relationship between the root of the autoregression lag function and the coefficient values. We can simulate the series as follows:

```
set.seed(123)
N <- 100
phi <- c(1.3, -0.4)
eps <- rnorm(N, mean = 0, sd = 0.2)
Y <- NULL
Y[1] <- eps[1]
Y[2] <- phi[1] * Y[1] + eps[2]
for(i in 3:N){
  Y[i] = phi[1] * Y[i - 1] + phi[2] * Y[i - 2] + eps[i]
}
```

```
plot.ts(Y)
```



Univariate series: $\mathrm{AR}(2)$: $\forall i$: $|z_i| < 1$

Assume that the AR lag function is:

$$\Phi(L) = (1 - 0.5L)(1 - 1.2L) = 1 - 1.7L + 0.6L^2$$

So, our process is:

$$Y_t = 1.7Y_{t-1} - 0.6Y_{t-2} + \epsilon_t$$

the root of $1 - 1.7z + 0.6z^2 = 0$ is:

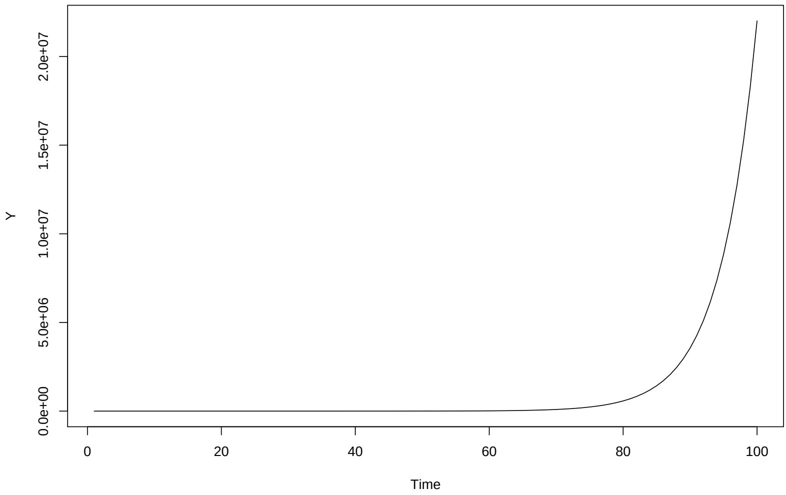
```
roots <- polyroot(c(1, -1.7, 0.6))
paste0("Absolute: |z_i| = ", round(abs(roots), 4))
```

```
## [1] "Absolute: |z_i| = 0.8333" "Absolute: |z_i| = 2"
```

The above **non-stationary** process highlights the relationship between the root of the autoregression lag function and the coefficient values. We can simulate the series as follows:

```
set.seed(123)
N <- 100
phi <- c(1.7, -0.6)
eps <- rnorm(N, mean = 0, sd = 0.2)
Y <- NULL
Y[1] <- eps[1]
Y[2] <- phi[1] * Y[1] + eps[2]
for(i in 3:N){
  Y[i] = phi[1] * Y[i - 1] + phi[2] * Y[i - 2] + eps[i]
}
```

```r
plot.ts(Y)
```



Univariate series: $\mathrm{AR}(2)$: $\exists i$: $z_i = 1$ (unit root)

Assume that the AR lag function is:

$$\Phi(L) = (1 - 0.5L)(1 - 1 \cdot L) = 1 - 1.5L + 0.5L^2$$

So, our process is:

$$Y_t = 1.5Y_{t-1} - 0.5Y_{t-2} + \epsilon_t$$

the root of $1 - 1.5z + 0.5z^2 = 0$ is:

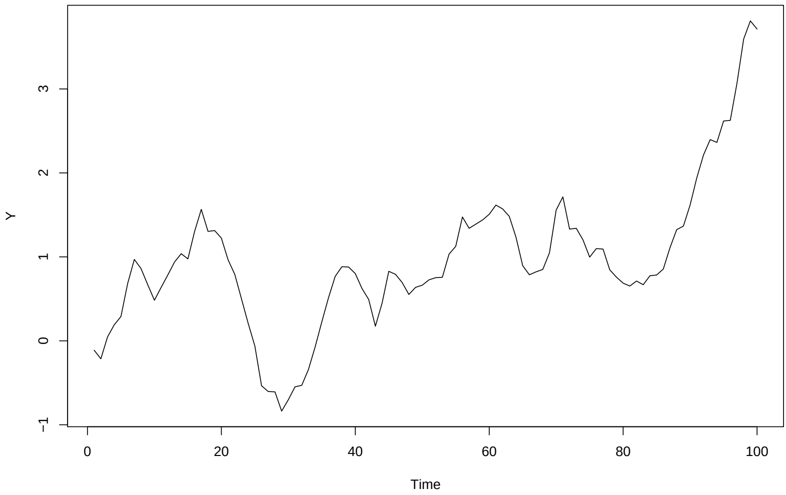
```
roots <- polyroot(c(1, -1.5, 0.5))
paste0("Absolute: |z_i| = ", round(abs(roots), 4))
```

```
## [1] "Absolute: |z_i| = 1" "Absolute: |z_i| = 2"
```

The above **unit-root** process highlights the relationship between the root of the autoregression lag function and the coefficient values. We can simulate the series as follows:

```
set.seed(123)
N <- 100
phi <- c(1.5, -0.5)
eps <- rnorm(N, mean = 0, sd = 0.2)
Y <- NULL
Y[1] <- eps[1]
Y[2] <- phi[1] * Y[1] + eps[2]
for(i in 3:N){
  Y[i] = phi[1] * Y[i - 1] + phi[2] * Y[i - 2] + eps[i]
}
```

```
plot.ts(Y)
```



Univariate series: $\mathrm{AR(p)}$: Stationarity and Unit root

Multivariate series: $\text{VAR}(1)$: $\forall i$: $|z_i| > 1$

Note: See "Symbolic Computation in R".

Assume that:

$$\begin{bmatrix} Y_{1,t} \\ Y_{2,t} \end{bmatrix} = \begin{pmatrix} 0.5 & 1.5 \\ 0 & 0.8 \end{pmatrix} \begin{bmatrix} Y_{1,t-1} \\ Y_{2,t-1} \end{bmatrix} + \begin{bmatrix} \epsilon_{1t} \\ \epsilon_{2t} \end{bmatrix}$$

where the absolute root values of $\det(\mathbf{I} - \Phi z) = (1 - 0.5z)(1 - 0.8z) = 0$ are:

```
z <- rSymPy::Var("z")
Theta <- rSymPy::sympy("Theta = Matrix([[0.5, 1.5], [0, 0.8]])")
cat(rSymPy::sympy("D = Matrix([[1, 0], [0, 1]]) - Theta * z"))
```

```
## [1 - 0.5*z,    -1.5*z]
## [        0, 1 - 0.8*z]
```

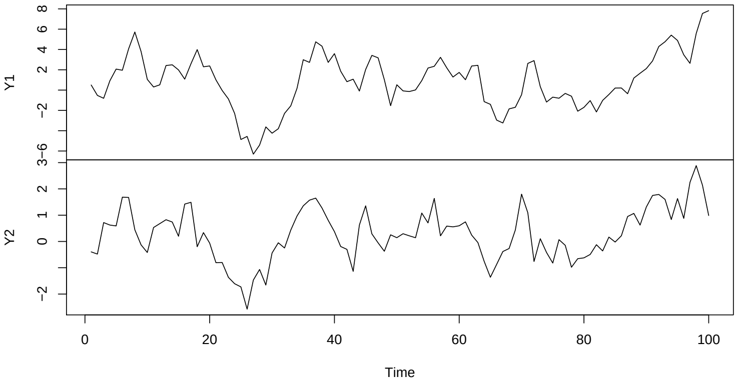
```
rSymPy::sympy("abs(roots(D.det(), z))")
```

```
## [1] "[1.25000000000000, 2.00000000000000]"
```

The above **stationary** example illustrates how the parameters from different equations relate to the autoregressive lag polynomial root calculation. We can simulate the process as well..

```
set.seed(123)
N <- 100
Theta_1 = matrix(c(0.5, 1.5, 0, 0.8), ncol = 2, byrow = TRUE)
e = MASS::mvrnorm(n = N, mu = c(0, 0), Sigma = diag(2) * 0.5)
Y <- matrix(data = 0, nrow = N, ncol = 2)
colnames(Y) <- c("Y1", "Y2")
Y[1, ] = e[1, ]
for(j in 2:N){
  Y[j, ] = Theta_1 %*% Y[j - 1, ] + e[j, ]
}
```

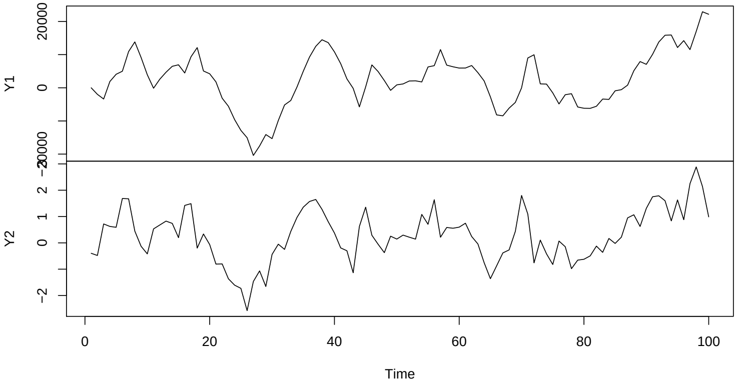
```
plot.ts(Y, plot.type = "multiple", oma.multi = c(5, 0, 1, 2))
```



An interesting point is that since $\phi_{21} = 0$ - even if our coefficient matrix was:

$$\begin{bmatrix} Y_{1,t} \\ Y_{2,t} \end{bmatrix} = \begin{pmatrix} 0.5 & 5000 \\ 0 & 0.8 \end{pmatrix} \begin{bmatrix} Y_{1,t-1} \\ Y_{2,t-1} \end{bmatrix} + \begin{bmatrix} \epsilon_{1t} \\ \epsilon_{2t} \end{bmatrix}$$

The VAR would still be stationary, the scale of $Y_{1,t}$ would change the most:



Assume that:

$$\begin{bmatrix} Y_{1,t} \\ Y_{2,t} \end{bmatrix} = \begin{pmatrix} 0.5 & -0.3 \\ 0.6 & 0.8 \end{pmatrix} \begin{bmatrix} Y_{1,t-1} \\ Y_{2,t-1} \end{bmatrix} + \begin{bmatrix} \epsilon_{1t} \\ \epsilon_{2t} \end{bmatrix}$$

where the absolute root values of
$\det(\mathbf{I} - \Phi z) = (1 - 0.5z)(1 - 0.8z) + 0.18z^2 = 0$ are:

```
z <- rSymPy::Var("z")
Theta <- rSymPy::sympy("Theta = Matrix([[0.5, -0.3], [0.6, 0.8]])")
cat(rSymPy::sympy("D = Matrix([[1, 0], [0, 1]]) - Theta * z"))
```

```
## [1 - 0.5*z,     0.3*z]
## [   -0.6*z, 1 - 0.8*z]
```

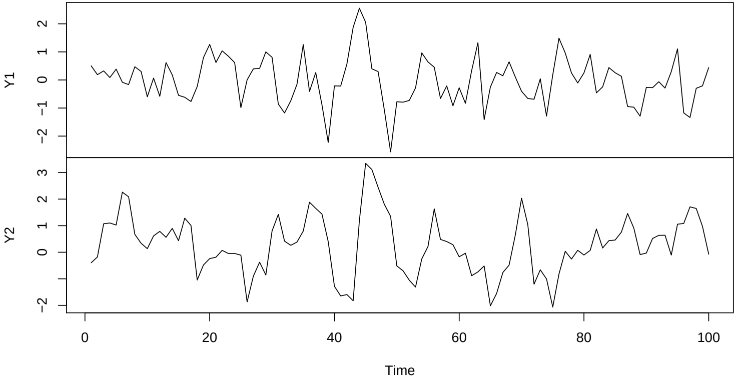
```
rSymPy::sympy("abs(roots(D.det(), z))")
```

```
## [1] "[1.31306432859723, 1.31306432859723]"
```

The above **stationary** example illustrates how the parameters from different equations relate to the autoregressive lag polynomial root calculation.

```r
set.seed(123)
N <- 100
Theta_1 = matrix(c(0.5, -0.3, 0.6, 0.8), ncol = 2, byrow = TRUE)
e = MASS::mvrnorm(n = N, mu = c(0, 0), Sigma = diag(2) * 0.5)
Y <- matrix(data = 0, nrow = N, ncol = 2)
colnames(Y) <- c("Y1", "Y2")
Y[1, ] = e[1, ]
for(j in 2:N){
  Y[j, ] = Theta_1 %*% Y[j - 1, ] + e[j, ]
}
```

```r
plot.ts(Y, plot.type = "multiple", oma.multi = c(5, 0, 1, 2))
```



Multivariate series: VAR(1): $\exists i$: $|z_i| < 1$

Assume that:

$$\begin{bmatrix} Y_{1,t} \\ Y_{2,t} \end{bmatrix} = \begin{pmatrix} 1.1 & 0.5 \\ 0.6 & 0.8 \end{pmatrix} \begin{bmatrix} Y_{1,t-1} \\ Y_{2,t-1} \end{bmatrix} + \begin{bmatrix} \epsilon_{1t} \\ \epsilon_{2t} \end{bmatrix}$$

where the absolute root values of
$\det(\mathbf{I} - \Phi z) = (1 - 1.1z)(1 - 0.8z) - 0.3z^2 = 0$ are:

```
z <- rSymPy::Var("z")
Theta <- rSymPy::sympy("Theta = Matrix([[1.1, 0.5], [0.6, 0.8]])")
cat(rSymPy::sympy("D = Matrix([[1, 0], [0, 1]]) - Theta * z"))
```

```
## [1 - 1.1*z,    -0.5*z]
## [   -0.6*z, 1 - 0.8*z]
```

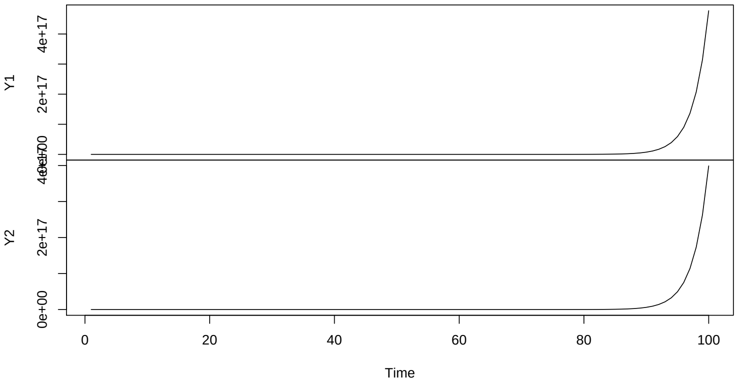
```
rSymPy::sympy("abs(roots(D.det(), z))")
```

```
## [1] "[0.658808905896505, 2.61705316306901]"
```

The above **non-stationary** example illustrates how the parameters from different equations relate to the autoregressive lag polynomial root calculation.

```r
set.seed(123)
N <- 100
Theta_1 = matrix(c(1.1, 0.5, 0.6, 0.8), ncol = 2, byrow = TRUE)
e = MASS::mvrnorm(n = N, mu = c(0, 0), Sigma = diag(2) * 0.5)
Y <- matrix(data = 0, nrow = N, ncol = 2)
colnames(Y) <- c("Y1", "Y2")
Y[1, ] = e[1, ]
for(j in 2:N){
  Y[j, ] = Theta_1 %*% Y[j - 1, ] + e[j, ]
}
```

```r
plot.ts(Y, plot.type = "multiple", oma.multi = c(5, 0, 1, 2))
```



Multivariate series: $\mathrm{VAR}(1)$: $\exists i$: $z_i = 1$

To simulate a $\mathrm{VAR}(1)$ with a unit root, it would make it easier if we start with its stationary representation in terms of the differences and fix a couple of parameters. See the example in the lecture files for higher lag orders. For $p = 1$, we can specify the following process

Assume that:

$$\begin{bmatrix} Y_{1,t} \\ Y_{2,t} \end{bmatrix} = \begin{pmatrix} 1 & 0.5 \\ 0 & 0.8 \end{pmatrix} \begin{bmatrix} Y_{1,t-1} \\ Y_{2,t-1} \end{bmatrix} + \begin{bmatrix} \epsilon_{1t} \\ \epsilon_{2t} \end{bmatrix}$$

where the absolute root values of
$\det(\mathbf{I} - \Phi z) = (1 - 1 \cdot z)(1 - 0.8z) - 0.3z^2 = 0$ are:

```
z <- rSymPy::Var("z")
Theta <- rSymPy::sympy("Theta = Matrix([[1, 0.5], [0, 0.8]])")
cat(rSymPy::sympy("D = Matrix([[1, 0], [0, 1]]) - Theta * z"))
```

```
## [1 - z,    -0.5*z]
## [    0, 1 - 0.8*z]
```

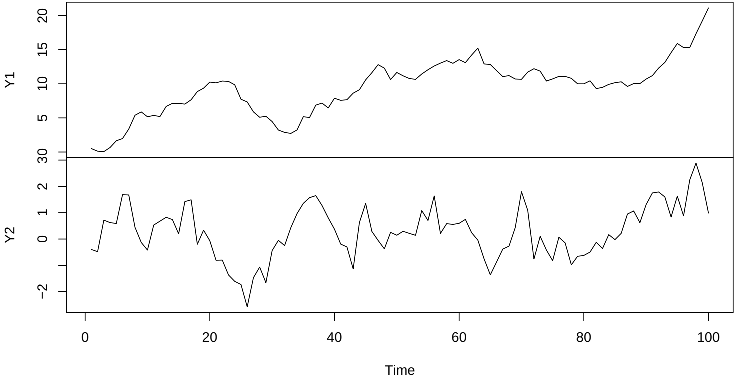
```
rSymPy::sympy("abs(roots(D.det(), z))")
```

```
## [1] "[1.00000000000000, 1.25000000000000]"
```

The above **unit root** example illustrates how the parameters from different equations relate to the autoregressive lag polynomial root calculation.

```r
set.seed(123)
N <- 100
Theta_1 = matrix(c(1, 0.5, 0, 0.8), ncol = 2, byrow = TRUE)
e = MASS::mvrnorm(n = N, mu = c(0, 0), Sigma = diag(2) * 0.5)
Y <- matrix(data = 0, nrow = N, ncol = 2)
colnames(Y) <- c("Y1", "Y2")
Y[1, ] = e[1, ]
for(j in 2:N){
  Y[j, ] = Theta_1 %*% Y[j - 1, ] + e[j, ]
}
```

```r
plot.ts(Y, plot.type = "multiple", oma.multi = c(5, 0, 1, 2))
```



VAR(1): unit root and cointegration

From the previous example we have that:

$$\Pi = \sum_{k=1}^{p} \Theta_j - \mathbf{I} = \begin{pmatrix} 1-1 & 0.5 \\ 0 & 0.8-1 \end{pmatrix} = \begin{pmatrix} 0 & 0.5 \\ 0 & -0.2 \end{pmatrix}$$

So $rank(\Pi) = 1$. This means that the system has one cointegration relation.

```
print(Matrix::rankMatrix(Theta_1 - diag(2))[1])
```

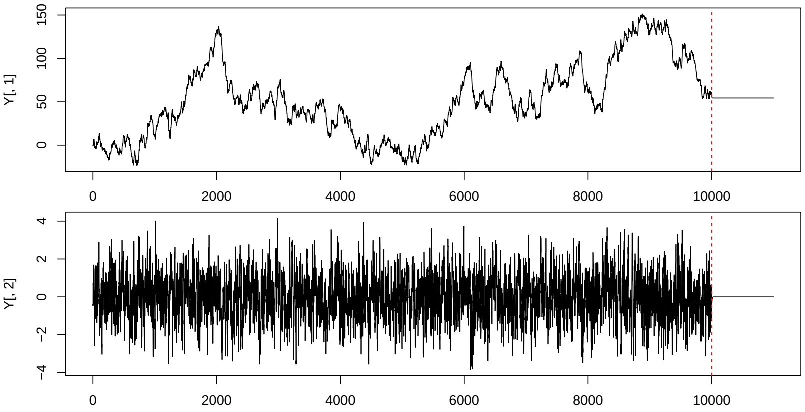
```
## [1] 1
```

How does a *long-run* equilibrium relationship look like in practice? Since we are are simulating data - we can try just that!

In the long-run there are **no more shocks**, so assume that the shocks stop after 10^3 observations and we simulate a total of $10^3 + 1000$ observations.

```
set.seed(123)
N <- 1e4 + 1000
Theta_1 = matrix(c(1, 0.5, 0, 0.8), ncol = 2, byrow = TRUE)
e = MASS::mvrnorm(n = N, mu = c(0, 0), Sigma = diag(2) * 0.5)
e[(1e4 + 1):N, ] <- 0
Y <- matrix(data = 0, nrow = N, ncol = 2)
colnames(Y) <- c("Y1", "Y2")
Y[1, ] = e[1, ]
for(j in 2:N){
  Y[j, ] = Theta_1 %*% Y[j - 1, ] + e[j, ]
}
```

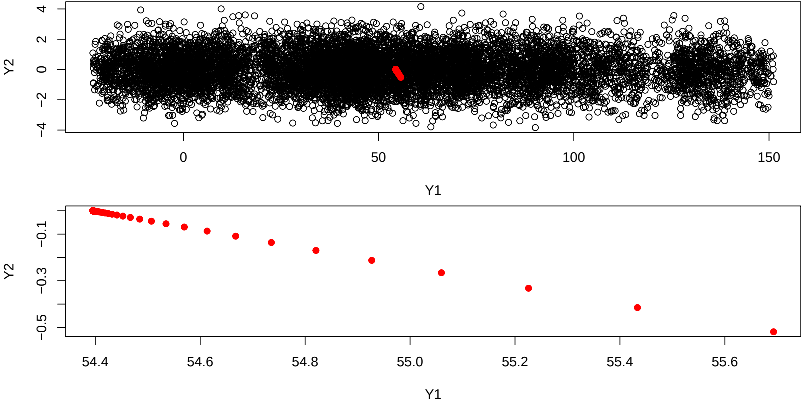
```r
par(mfrow = c(2, 1), oma = c(0, 0, 0, 0), mar = c(2, 4, 0.5, 1))
plot.ts(Y[, 1])
abline(v = 1e4, col = "red", lty = 2)
plot.ts(Y[, 2])
abline(v = 1e4, col = "red", lty = 2)
```



It appears that both series are somewhat constant, but is there a **linear relationship?**.

A simple cross-sectional plot can be used to visualize this:

```
par(mfrow = c(2, 1), oma = c(0, 0, 0, 0), mar = c(4, 4, 0.5, 1))
plot(Y)
points(Y[-c(1:1e4), ], pch = 19, col = "red")
plot(Y[-c(1:1e4), ], pch = 19, col = "red", xlab = "Y1")
```



It appears that there is a clear linear relationship. This is because initially past values of Y_1 and Y_2 still contain shocks so the system adjusts until equilibrium.

But how would it look like if there was no cointegration?

VAR(1): unit root and no cointegration

Assume that:

$$\begin{bmatrix} Y_{1,t} \\ Y_{2,t} \end{bmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{bmatrix} Y_{1,t-1} \\ Y_{2,t-1} \end{bmatrix} + \begin{bmatrix} \epsilon_{1t} \\ \epsilon_{2t} \end{bmatrix}$$

```
z <- rSymPy::Var("z")
Theta <- rSymPy::sympy("Theta = Matrix([[1, 0], [0, 1]])")
rSymPy::sympy("D = Matrix([[1, 0], [0, 1]]) - Theta * z", retclass = "NULL")
rSymPy::sympy("abs(roots(D.det(), z))")
```

```
## [1] "[1]"
```

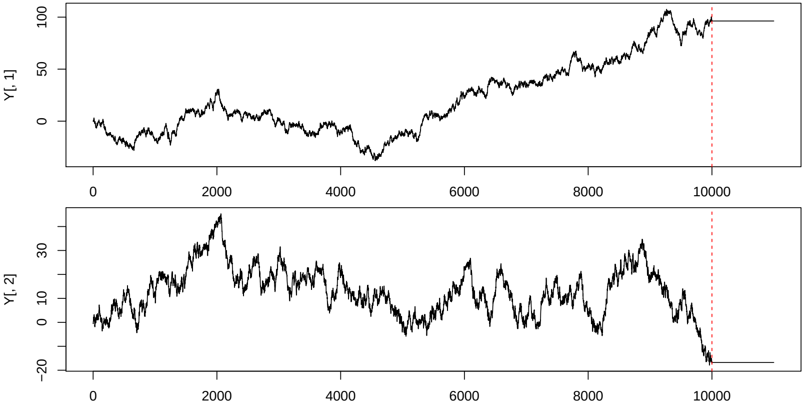
We have that:

$$\Pi = \sum_{k=1}^{p} \Theta_j - \mathbf{I} = \begin{pmatrix} 1-1 & 0 \\ 0 & 1-1 \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$$

So $rank(\Pi) = 0$. This means that the system has nocointegration relation.

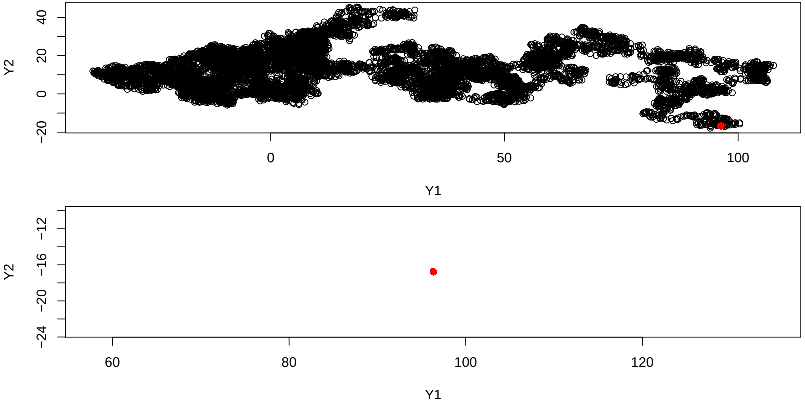
```
set.seed(123)
N <- 1e4 + 1000
Theta_1 = matrix(c(1, 0, 0, 1), ncol = 2, byrow = TRUE)
e = MASS::mvrnorm(n = N, mu = c(0, 0), Sigma = diag(2) * 0.5)
e[(1e4 + 1):N, ] <- 0
Y <- matrix(data = 0, nrow = N, ncol = 2)
colnames(Y) <- c("Y1", "Y2")
Y[1, ] = e[1, ]
for(j in 2:N){
  Y[j, ] = Theta_1 %*% Y[j - 1, ] + e[j, ]
}
```

```r
par(mfrow = c(2, 1), oma = c(0, 0, 0, 0), mar = c(2, 4, 0.5, 1))
plot.ts(Y[, 1])
abline(v = 1e4, col = "red", lty = 2)
plot.ts(Y[, 2])
abline(v = 1e4, col = "red", lty = 2)
```



Even when there is no cointegration, it appears that both of the series are somewhat constant, but is there a **linear relationship?**.

```r
par(mfrow = c(2, 1), oma = c(0, 0, 0, 0), mar = c(4, 4, 0.5, 1))
plot(Y)
points(Y[-c(1:1e4), ], pch = 19, col = "red")
plot(Y[-c(1:1e4), ], pch = 19, col = "red", xlab = "Y1")
```



Both of the series are equal to their last value in the long-run as there are no more shocks.

VAR(2) with unit root and no cointegration

This example is similar to the one from the Summary lecture:

$$\begin{pmatrix} Y_t \\ X_t \end{pmatrix} = \begin{pmatrix} 0.4 & 0.5 \\ -0.3 & 0.8 \end{pmatrix} \begin{pmatrix} Y_{t-1} \\ X_{t-1} \end{pmatrix} + \begin{pmatrix} 0.6 & -0.5 \\ 0.3 & 0.2 \end{pmatrix} \begin{pmatrix} Y_{t-2} \\ X_{t-2} \end{pmatrix} + \begin{pmatrix} \epsilon_{1,t} \\ \epsilon_{2,t} \end{pmatrix}$$

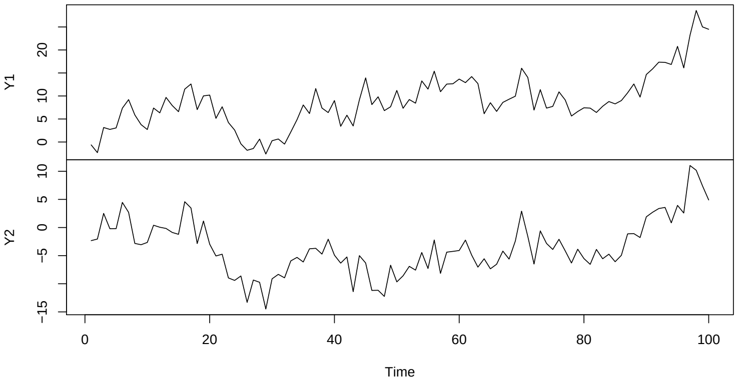
We can verify that there is no cointegration:

$$\Pi = \Theta_1 + \Theta_2 - I = \begin{pmatrix} 0.4 + 0.6 - 1 & 0.5 - 0.5 \\ -0.3 + 0.3 & 0.8 + 0.2 - 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$$

(The lecture summary shows that the series contains two unit roots).

```
set.seed(123)
N = 100
Theta_1 = matrix(c(0.4, 0.5, -0.3, 0.8), ncol = 2, byrow = TRUE)
Theta_2 = matrix(c(0.6, -0.5, 0.3, 0.2), ncol = 2, byrow = TRUE)
Sigma = matrix(c(8, 6, 6, 9), ncol = 2, byrow = TRUE)
e = MASS::mvrnorm(n = N, mu = c(0, 0), Sigma = Sigma)
Y <- matrix(data = 0, nrow = N, ncol = 2)
colnames(Y) <- c("Y1", "Y2")
Y[1, ] = e[1, ]
Y[2, ] = Theta_1 %*% Y[1, ] + e[2, ]
for(j in 3:N){
  Y[j, ] = Theta_1 %*% Y[j-1, ] + Theta_2 %*% Y[j-2, ] + e[j, ]
}
```

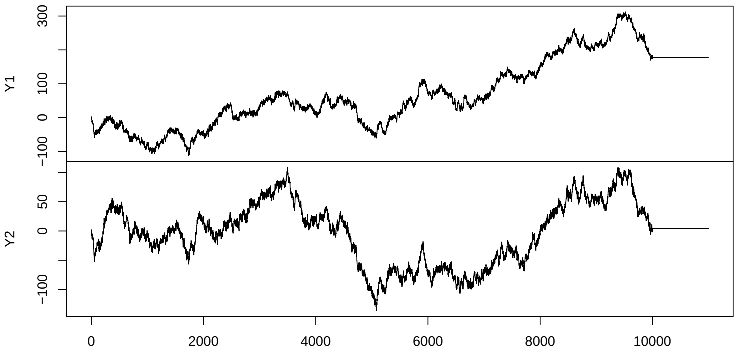
```
plot.ts(Y, plot.type = "multiple", oma.multi = c(5, 0, 1, 2))
```



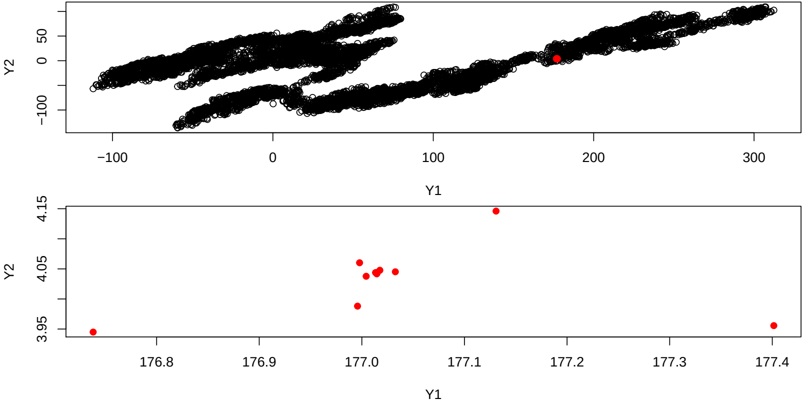
Let us now examine the long-run relationship:

```
set.seed(1234)
N = 1e4 + 1000
Sigma = matrix(c(8, 6, 6, 9), ncol = 2, byrow = TRUE)
e = MASS::mvrnorm(n = N, mu = c(0, 0), Sigma = Sigma)
e[(1e4 + 1):N, ] <- 0
Y <- matrix(data = 0, nrow = N, ncol = 2)
colnames(Y) <- c("Y1", "Y2")
Y[1, ] = e[1, ]
Y[2, ] = Theta_1 %*% Y[1, ] + e[2, ]
for(j in 3:N){
  Y[j, ] = Theta_1 %*% Y[j-1, ] + Theta_2 %*% Y[j-2, ] + e[j, ]
}
```

```
plot.ts(Y, plot.type = "multiple", oma.multi = c(5, 0, 1, 2))
```



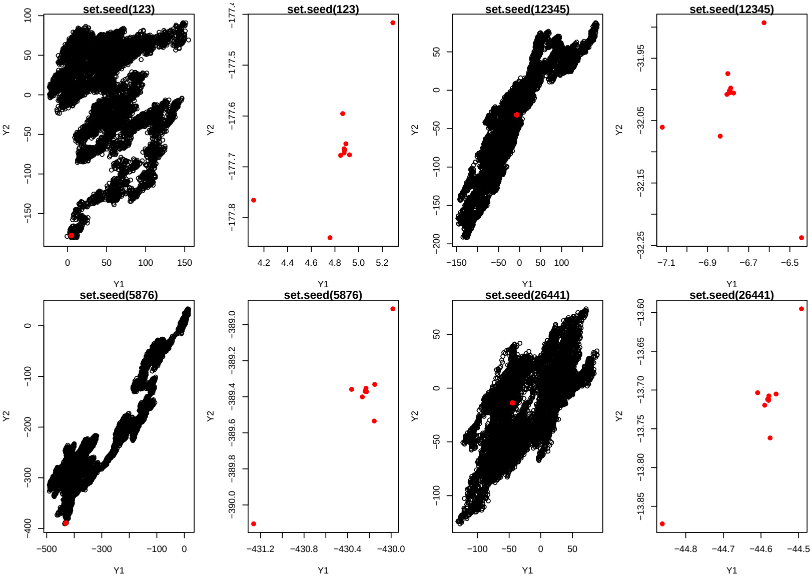
```
par(mfrow = c(2, 1), oma = c(0, 0, 0, 0), mar = c(4, 4, 0.5, 1))
plot(Y)
points(Y[-c(1:1e4), ], pch = 19, col = "red")
plot(Y[-c(1:1e4), ], pch = 19, col = "red", xlab = "Y1")
```



Even though both of the series **appear** to be constant in the long-run (just like in the case of cointegration) - there is no linear relationship!

If we suspect that there is an equilibrium relationship, which is masked by the presence of various shocks (i.e. innovations) - this is additional information, which we **omitting** if we do not take into account the cointegration.

We can repeat the above with different seeds to verify:



VAR(2) with unit root and cointegration

$$\begin{pmatrix} Y_t \\ X_t \end{pmatrix} = \begin{pmatrix} 1 & 0.5 \\ 0 & 0.5 \end{pmatrix} \begin{pmatrix} Y_{t-1} \\ X_{t-1} \end{pmatrix} + \begin{pmatrix} 0.0 & -0.5 \\ 0.0 & 0.3 \end{pmatrix} \begin{pmatrix} Y_{t-2} \\ X_{t-2} \end{pmatrix} + \begin{pmatrix} \epsilon_{1,t} \\ \epsilon_{2,t} \end{pmatrix}$$

```r
z <- rSymPy::Var("z")
rSymPy::sympy("Theta_1 = Matrix([[1, 0.5], [0, -0.5]])", retclass = "NULL")
rSymPy::sympy("Theta_2 = Matrix([[0, 0.5], [0, 0.3]])", retclass = "NULL")
cat(rSymPy::sympy("D = Matrix([[1, 0], [0, 1]]) - Theta_1*z - Theta_2*(z**2)"))
```

```
## [1 - z,    -0.5*z - 0.5*z**2]
## [    0, 1 + 0.5*z - 0.3*z**2]
```

Note that there is one unit root in the series:

```r
rSymPy::sympy("abs(roots(D.det(), z))")
```

```
## [1] "[1, 2.84026576313205, 1.17359909646538]"
```

We can verify that $rank(\Pi) = 1$:

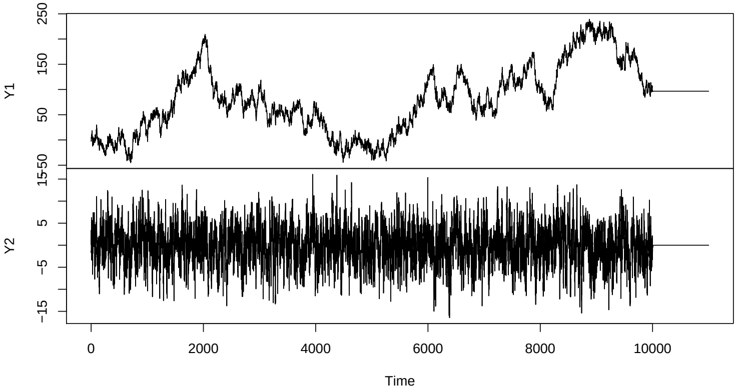
```r
Theta_1 = matrix(c(1, 0.5, 0, 0.5), ncol = 2, byrow = TRUE)
Theta_2 = matrix(c(0, -0.5, 0, 0.3), ncol = 2, byrow = TRUE)
```

```r
print(Matrix::rankMatrix(Theta_1 + Theta_2 - diag(2))[1])
```

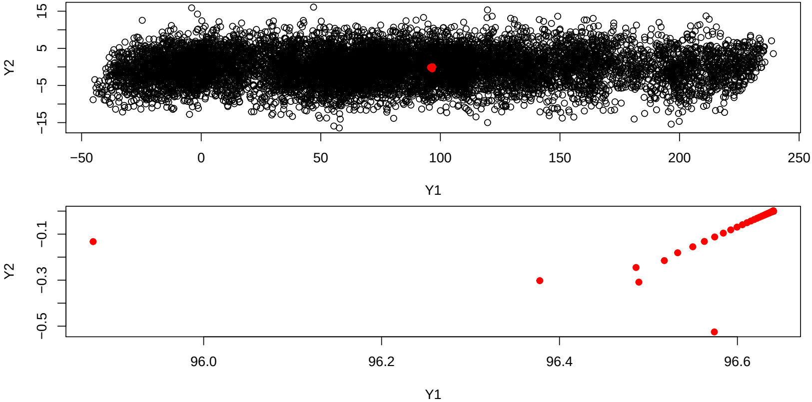
```
## [1] 1
```

```r
set.seed(123)
N = 1e4 + 1000
Sigma = matrix(c(8, 6, 6, 9), ncol = 2, byrow = TRUE)
e = MASS::mvrnorm(n = N, mu = c(0, 0), Sigma = Sigma)
e[(1e4 + 1):N, ] <- 0
Y <- matrix(data = 0, nrow = N, ncol = 2)
colnames(Y) <- c("Y1", "Y2")
Y[1, ] = e[1, ]
Y[2, ] = Theta_1 %*% Y[1, ] + e[2, ]
for(j in 3:N){
  Y[j, ] = Theta_1 %*% Y[j-1, ] + Theta_2 %*% Y[j-2, ] + e[j, ]
}
```

```r
plot.ts(Y, plot.type = "multiple", oma.multi = c(5, 0, 1, 2))
```



```r
par(mfrow = c(2, 1), oma = c(0, 0, 0, 0), mar = c(4, 4, 0.5, 1))
plot(Y)
points(Y[-c(1:1e4), ], pch = 19, col = "red")
plot(Y[-c(1:1e4), ], pch = 19, col = "red", xlab = "Y1")
```



After there are no more shocks - when $\epsilon_{i,t} = 0\ \forall i, j$ and for $t > \tilde{T}$, then the cointegration relation means that the series adjusts to the equilibrium relationship.

We can repeat the above with different seeds to verify: