# Cointegração

*Henri Makika*

*6/18/2019*

## Cointegração

Pour résumer le cours jusqu'à présent, on a vu :

i. ARIMA
ii. VAR
iii. SVAR
iv. Cointegração

Pour VAR la restriction est faite en utilisant la décomposition de Cholesky, c'est-à-dire on fait ordonner les variables selon que : exogène vs endogène.

Pour le modèle SVAR la restriction est fait sur base de la matrice A (Amat), voir aula 11. Il sied à noter que toutes les restrictions sont réalisées sur base de théories économiques à la base. VAR ne traite que les variables stationnaires, sinon il faut faire le test de RU et applique la différence dans le cas de une racine unitaire, applique encore la difference dans le cas de plus de racine unitaire. (même raisonnement pour SVAR).

VAR donne les informations de court terme tandisque Cointégration donne l'information de long terme, et VECM donne les deux.

La Cointégration est la combinaison linéaire des variables non stationnaires et des variables qui sont intégrées de même ordre.

## Cointégration à la Engle-Granger et Phillips-Ouliares

```
library(urca)
library(MASS)
library(vars)
library(lmtest)
#library(MTS)
```

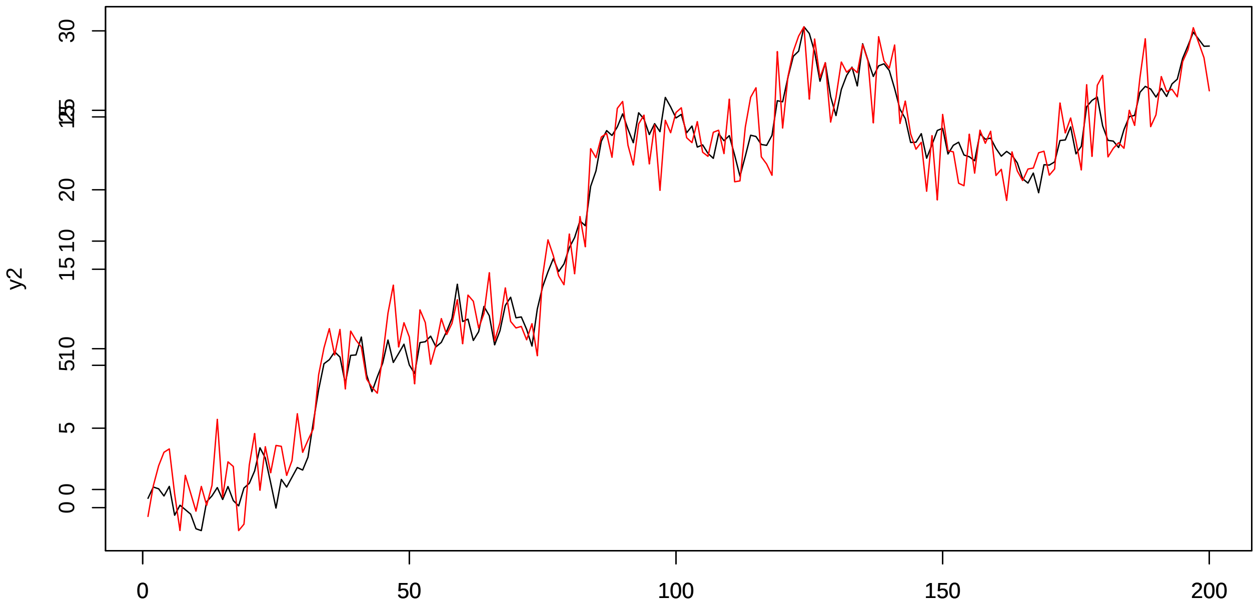
## Séries sintéticas

```
set.seed(12345)
e1 = rnorm(200)
e2 = rnorm(200)

y1 = cumsum(e1)
y2 = 0.6*y1 + e2

plot.ts(y1, xlab = "", ylab = "", main = "Cointegration")
par(new = T)
plot.ts(y2, col = "red")
```

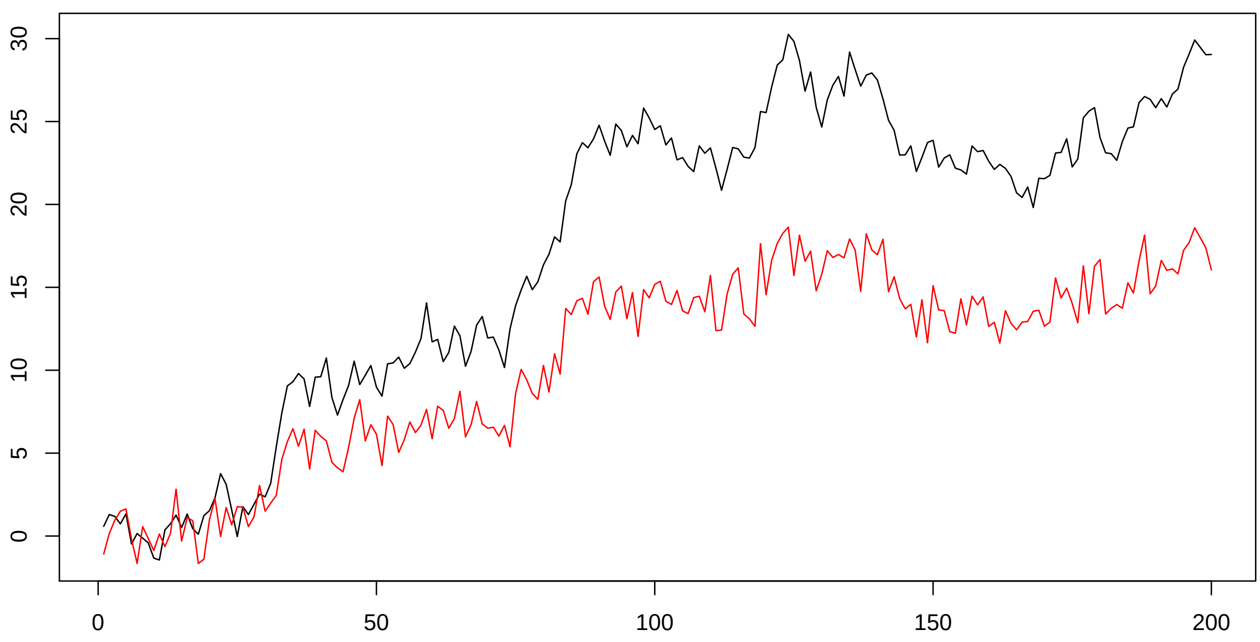
**Cointegration**




```
# La meilleure manière de présenter ce meme graphique est :

plot.ts(y1, xlab = "", ylab = "", main = "Cointegration")
lines(y2, col = "red")
```

**Cointegration**



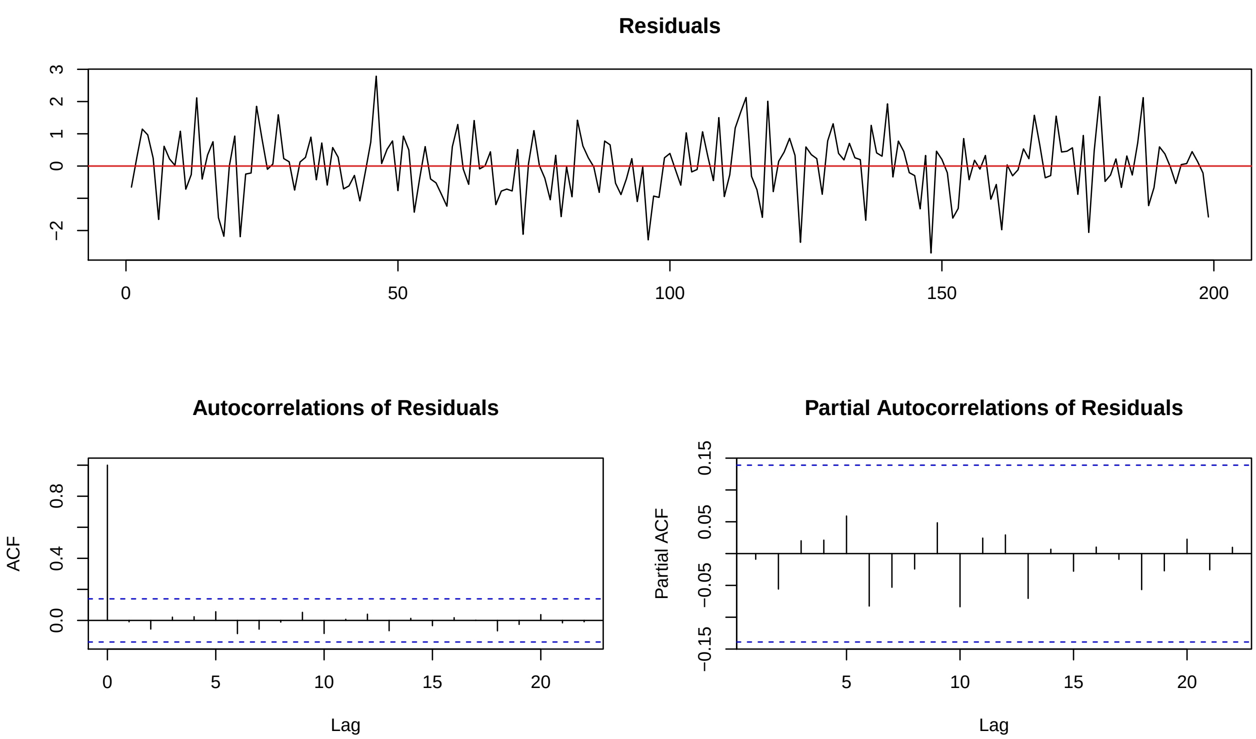
# Procedimento Engle-Granger

```r
mod.reg = lm(y2 ~ y1)
summary(mod.reg)
```

```
##
## Call:
## lm(formula = y2 ~ y1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.72396 -0.56977  0.09572  0.56988  2.73839
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.074274   0.143930  -0.516    0.606
## y1           0.609118   0.007294  83.515   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9569 on 198 degrees of freedom
## Multiple R-squared:  0.9724, Adjusted R-squared:  0.9723
## F-statistic:  6975 on 1 and 198 DF,  p-value: < 2.2e-16
```

```r
erro = residuals(mod.reg)

erro.df = ur.df(erro, type = "none", lags = 0)
plot(erro.df)
```




```r
summary(erro.df)
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression none
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 - 1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.69748 -0.56896  0.06992  0.56949  2.78661
##
## Coefficients:
##         Estimate Std. Error t value Pr(>|t|)
## z.lag.1 -1.06436    0.07104  -14.98   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.95 on 198 degrees of freedom
## Multiple R-squared:  0.5314, Adjusted R-squared:  0.529
## F-statistic: 224.5 on 1 and 198 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic is: -14.9835
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau1 -2.58 -1.95 -1.62
```

La valeur du test -14.98 est hors interval de confiance, alors on rejet l'hypotèse nulle de non cointégration, la série est cointégrée.

```
erro.lag = erro[-c(1, 200)]
dy1 = diff(y1)
dy2 = diff(y2)

diff.dat = data.frame(embed(cbind(dy1, dy2), 2))
colnames(diff.dat) = c("dy1", "dy2", "dy1.1", "dy2.1")

ecm.reg = lm(dy2 ~ erro.lag + dy1.1 + dy2.1, data = diff.dat)
summary(ecm.reg)
```

```
##
## Call:
## lm(formula = dy2 ~ erro.lag + dy1.1 + dy2.1, data = diff.dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.81326 -0.83773  0.02054  0.76891  3.15603
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)  0.097276   0.084670   1.149    0.252
## erro.lag    -1.118505   0.129633  -8.628 2.23e-15 ***
## dy1.1       -0.001373   0.098874  -0.014    0.989
## dy2.1       -0.002074   0.088734  -0.023    0.981
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.18 on 194 degrees of freedom
## Multiple R-squared:  0.4512, Adjusted R-squared:  0.4428
## F-statistic: 53.18 on 3 and 194 DF,  p-value: < 2.2e-16
```

```r
erro1 = erro[-1]
erro1.1 = cbind(erro1)

dados = ts(cbind(dy1, dy2))

modelo = VAR(dados, p = 1, type = "const", exogen = erro1.1)
summary(modelo)
```

```
##
## VAR Estimation Results:
## =========================
## Endogenous variables: dy1, dy2
## Deterministic variables: const
## Sample size: 198
## Log Likelihood: -486.386
## Roots of the characteristic polynomial:
## 0.4951 0.007721
## Call:
## VAR(y = dados, p = 1, type = "const", exogen = erro1.1)
##
##
## Estimation results for equation dy1:
## ====================================
## dy1 = dy1.l1 + dy2.l1 + const + erro1
##
##        Estimate Std. Error t value Pr(>|t|)
## dy1.l1  0.04930    0.08133   0.606   0.5451
## dy2.l1 -0.08847    0.05513  -1.605   0.1102
## const   0.14087    0.07678   1.835   0.0681 .
## erro1   0.04483    0.08000   0.560   0.5758
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 1.071 on 194 degrees of freedom
## Multiple R-Squared: 0.01495, Adjusted R-squared: -0.0002811
## F-statistic: 0.9815 on 3 and 194 DF,  p-value: 0.4025
##
##
## Estimation results for equation dy2:
## ====================================
## dy2 = dy1.l1 + dy2.l1 + const + erro1
##
##        Estimate Std. Error t value Pr(>|t|)
```

```
## dy1.l1  0.35085    0.06938   5.057 9.84e-07 ***
## dy2.l1 -0.55210    0.04703 -11.738  < 2e-16 ***
## const   0.07078    0.06549   1.081    0.281
## erro1   1.08870    0.06825  15.952  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.9133 on 194 degrees of freedom
## Multiple R-Squared: 0.6715,  Adjusted R-squared: 0.6665
## F-statistic: 132.2 on 3 and 194 DF,  p-value: < 2.2e-16
##
##
##
## Covariance matrix of residuals:
##        dy1    dy2
## dy1 1.1462 0.6858
## dy2 0.6858 0.8341
##
## Correlation matrix of residuals:
##        dy1    dy2
## dy1 1.0000 0.7013
## dy2 0.7013 1.0000
```

### Procedimento Phillips-Ouliares

```r
X = data.frame(y1, y2)
cointest_X = ca.po(X, demean = "constant", lag = "short", type = "Pu")
summary(cointest_X)
```

```
##
## ########################################
## # Phillips and Ouliaris Unit Root Test #
## ########################################
##
## Test of type Pu
## detrending of series with constant only
##
##
## Call:
## lm(formula = z[, 1] ~ z[, -1])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.6002 -0.8604 -0.1026  0.9044  4.5228
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.59936    0.22924   2.615  0.00962 **
## z[, -1]      1.59640    0.01912  83.515  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 1.549 on 198 degrees of freedom
## Multiple R-squared:  0.9724, Adjusted R-squared:  0.9723
## F-statistic:  6975 on 1 and 198 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic is: 54.1995
##
## Critical values of Pu are:
##                   10pct   5pct    1pct
## critical values 27.8536 33.713 48.0021
```

On rejet l'hypothèse nulle de non cointégration. nos séries sont cointégrées.

```
cointest2_X = ca.po(X, demean = "constant", type = "Pz")
summary(cointest2_X)
```

```
##
## ########################################
## # Phillips and Ouliaris Unit Root Test #
## ########################################
##
## Test of type Pz
## detrending of series with constant only
##
## Response y1 :
##
## Call:
## lm(formula = y1 ~ zr)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.65872 -0.72552  0.04243  0.65663  2.67186
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.32619    0.16062   2.031   0.0436 *
## zry1         1.04180    0.04931  21.126   <2e-16 ***
## zry2        -0.08649    0.07970  -1.085   0.2792
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.066 on 196 degrees of freedom
## Multiple R-squared:  0.9868, Adjusted R-squared:  0.9867
## F-statistic:  7353 on 2 and 196 DF,  p-value: < 2.2e-16
##
##
## Response y2 :
##
## Call:
## lm(formula = y2 ~ zr)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.8421 -0.8334 -0.0027  0.7618  3.2118
##
```

```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.15740    0.17664   0.891    0.374
## zry1         0.67196    0.05423  12.391   <2e-16 ***
## zry2        -0.11693    0.08765  -1.334    0.184
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.172 on 196 degrees of freedom
## Multiple R-squared:  0.9581, Adjusted R-squared:  0.9577
## F-statistic:  2243 on 2 and 196 DF,  p-value: < 2.2e-16
##
##
##
## Value of test-statistic is: 187.7052
##
## Critical values of Pz are:
##                    10pct    5pct    1pct
## critical values 47.5877 55.2202 71.9273
```

```
resid_coint = cointest_X@res[-c(1, 200)]
pz.reg = lm(dy2 ~ resid_coint + dy1.1 + dy2.1, data = diff.dat)
summary(pz.reg)
```

```
##
## Call:
## lm(formula = dy2 ~ resid_coint + dy1.1 + dy2.1, data = diff.dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.97714 -0.85419  0.09762  0.73931  3.15746
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.09592    0.08642   1.110    0.268
## resid_coint  0.64233    0.08050   7.979 1.25e-13 ***
## dy1.1        0.02490    0.10060   0.248    0.805
## dy2.1       -0.04782    0.08926  -0.536    0.593
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.205 on 194 degrees of freedom
## Multiple R-squared:  0.4283, Adjusted R-squared:  0.4195
## F-statistic: 48.45 on 3 and 194 DF,  p-value: < 2.2e-16
```

## Exercice 2

```
library(readxl)

Dados = read_excel("~/Videos/Unicamp_IE 2019/HO:236A Times Series/Aula 12/Indices.xlsx")
Dadosindice = ts(Dados[,2:4], start = c(1998, 1), frequency = 12)

dj = Dadosindice[,1]
```

```r
nasdaq = Dadosindice[,2]
sp = Dadosindice[,3]

coint_SPDJNasdaq = ca.po(Dadosindice, demean = "constant", type = "Pz")
summary(coint_SPDJNasdaq)
```

```
##
## ########################################
## # Phillips and Ouliaris Unit Root Test #
## ########################################
##
## Test of type Pz
## detrending of series with constant only
##
## Response Dow Jones :
##
## Call:
## lm(formula = `Dow Jones` ~ zr)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1564.63  -208.57    24.59   273.84   851.18
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)               1223.88272  468.90175   2.610   0.0105 *
## zrDow Jones                  0.79310    0.08236   9.629 7.04e-16 ***
## zrNASDAQ Composite Index    -0.01707    0.13056  -0.131   0.8962
## zrS&P500                     0.77315    0.82575   0.936   0.3514
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 414.2 on 99 degrees of freedom
## Multiple R-squared:  0.8176, Adjusted R-squared:  0.8121
## F-statistic:   148 on 3 and 99 DF,  p-value: < 2.2e-16
##
##
## Response NASDAQ Composite Index :
##
## Call:
## lm(formula = `NASDAQ Composite Index` ~ zr)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -718.71  -81.82   -3.46   85.63  857.14
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)               345.97878  262.31971   1.319    0.190
## zrDow Jones                -0.05429    0.04608  -1.178    0.242
## zrNASDAQ Composite Index    0.93168    0.07304  12.756   <2e-16 ***
## zrS&P500                    0.29860    0.46196   0.646    0.520
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 231.7 on 99 degrees of freedom
## Multiple R-squared:  0.9012, Adjusted R-squared:  0.8983
## F-statistic: 301.2 on 3 and 99 DF,  p-value: < 2.2e-16
##
##
## Response S&P500 :
##
## Call:
## lm(formula = `S&P500` ~ zr)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -185.044  -25.459    6.361   31.324  112.256
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)              168.795208  56.680910   2.978  0.00365 **
## zrDow Jones               -0.020215   0.009956  -2.030  0.04499 *
## zrNASDAQ Composite Index   0.006752   0.015782   0.428  0.66969
## zrS&P500                   1.017663   0.099817  10.195  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 50.07 on 99 degrees of freedom
## Multiple R-squared:  0.905,  Adjusted R-squared:  0.9021
## F-statistic: 314.4 on 3 and 99 DF,  p-value: < 2.2e-16
##
##
##
## Value of test-statistic is: 28.2857
##
## Critical values of Pz are:
##                    10pct    5pct     1pct
## critical values 80.2034 89.7619 109.4525
```

Pas cointégrées, on passe au modèle VAR.

### Modele VAR

```
Dados2 = log(Dadosindice)
```

### Ordem de modelo

```
ordem = VARselect(diff(Dados2), lag.max = 8, type = "none")
ordem
```

```
## $selection
## AIC(n)  HQ(n)  SC(n) FPE(n)
##      1      1      1      1
##
## $criteria
```

```
##                    1             2             3             4
## AIC(n) -2.045494e+01 -2.030624e+01 -2.024652e+01 -2.017231e+01
## HQ(n)  -2.035717e+01 -2.011072e+01 -1.995323e+01 -1.978125e+01
## SC(n)  -2.021299e+01 -1.982235e+01 -1.952068e+01 -1.920453e+01
## FPE(n)  1.307861e-09  1.518201e-09  1.613578e-09  1.741983e-09
##                    5             6             7             8
## AIC(n) -2.017421e+01 -2.009006e+01 -1.999490e+01 -1.996061e+01
## HQ(n)  -1.968538e+01 -1.950348e+01 -1.931054e+01 -1.917849e+01
## SC(n)  -1.896447e+01 -1.863839e+01 -1.830127e+01 -1.802504e+01
## FPE(n)  1.745520e-09  1.910008e-09  2.118335e-09  2.217096e-09
```

### Estimation

```r
modelo.var = VAR(diff(Dados2), p = 1, type = "none")
summary(modelo.var)
```

```
##
## VAR Estimation Results:
## =========================
## Endogenous variables: Dow.Jones, NASDAQ.Composite.Index, S.P500
## Deterministic variables: none
## Sample size: 102
## Log Likelihood: 614.697
## Roots of the characteristic polynomial:
## 0.1058 0.1058 0.05552
## Call:
## VAR(y = diff(Dados2), p = 1, type = "none")
##
##
## Estimation results for equation Dow.Jones:
## ==========================================
## Dow.Jones = Dow.Jones.l1 + NASDAQ.Composite.Index.l1 + S.P500.l1
##
##                           Estimate Std. Error t value Pr(>|t|)
## Dow.Jones.l1              -0.33957    0.28302  -1.200    0.233
## NASDAQ.Composite.Index.l1  0.05267    0.09994   0.527    0.599
## S.P500.l1                  0.25846    0.37786   0.684    0.496
##
##
## Residual standard error: 0.04478 on 99 degrees of freedom
## Multiple R-Squared: 0.0294,  Adjusted R-squared: -1.599e-05
## F-statistic: 0.9995 on 3 and 99 DF,  p-value: 0.3965
##
##
## Estimation results for equation NASDAQ.Composite.Index:
## =======================================================
## NASDAQ.Composite.Index = Dow.Jones.l1 + NASDAQ.Composite.Index.l1 + S.P500.l1
##
##                           Estimate Std. Error t value Pr(>|t|)
## Dow.Jones.l1               -0.4945     0.5579  -0.887    0.377
## NASDAQ.Composite.Index.l1   0.1435     0.1970   0.729    0.468
## S.P500.l1                   0.1831     0.7448   0.246    0.806
##
```

```
##
## Residual standard error: 0.08826 on 99 degrees of freedom
## Multiple R-Squared: 0.02432, Adjusted R-squared: -0.005251
## F-statistic: 0.8224 on 3 and 99 DF,  p-value: 0.4845
##
##
## Estimation results for equation S.P500:
## =======================================
## S.P500 = Dow.Jones.l1 + NASDAQ.Composite.Index.l1 + S.P500.l1
##
##                           Estimate Std. Error t value Pr(>|t|)
## Dow.Jones.l1              -0.36445    0.28258  -1.290    0.200
## NASDAQ.Composite.Index.l1  0.05463    0.09979   0.547    0.585
## S.P500.l1                  0.25554    0.37728   0.677    0.500
##
##
## Residual standard error: 0.04471 on 99 degrees of freedom
## Multiple R-Squared: 0.03236, Adjusted R-squared: 0.003033
## F-statistic: 1.103 on 3 and 99 DF,  p-value: 0.3515
##
##
##
## Covariance matrix of residuals:
##                         Dow.Jones NASDAQ.Composite.Index   S.P500
## Dow.Jones                0.001995               0.002554 0.001829
## NASDAQ.Composite.Index   0.002554               0.007781 0.003228
## S.P500                   0.001829               0.003228 0.001992
##
## Correlation matrix of residuals:
##                         Dow.Jones NASDAQ.Composite.Index S.P500
## Dow.Jones                  1.0000                 0.6482 0.9173
## NASDAQ.Composite.Index     0.6482                 1.0000 0.8199
## S.P500                     0.9173                 0.8199 1.0000
```

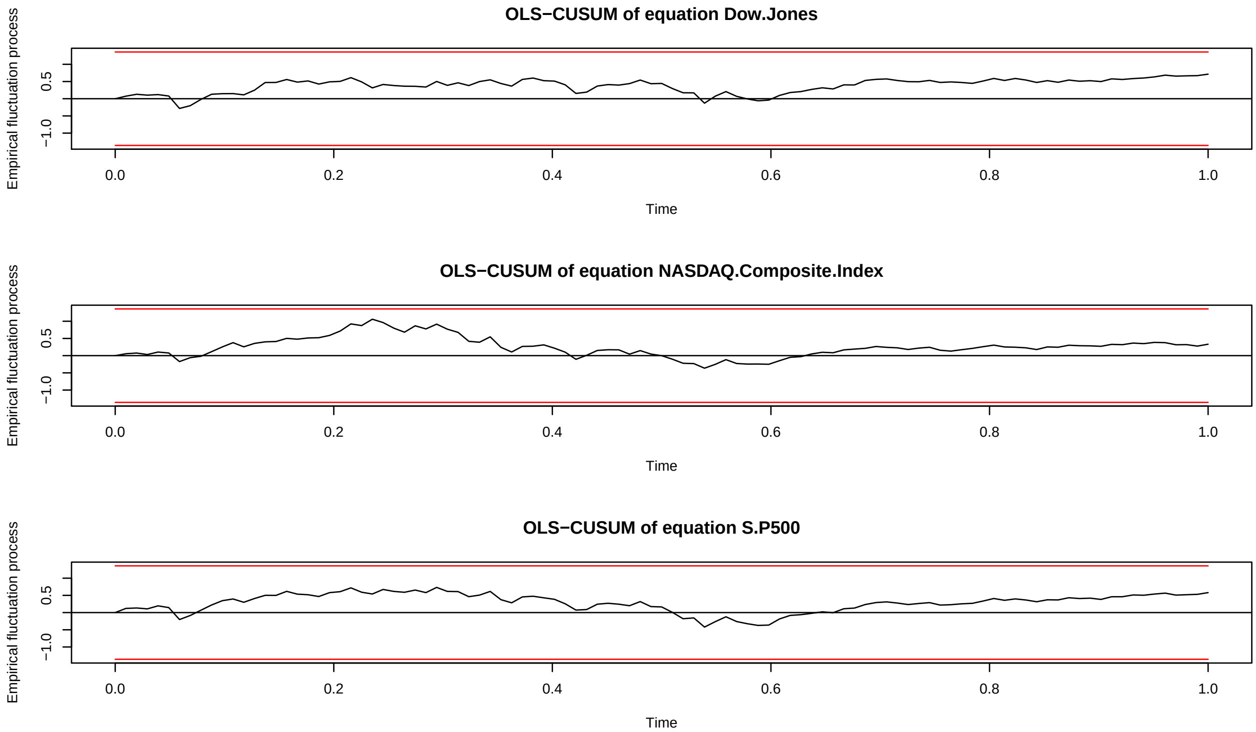
### Diagnóstico do modelo

```r
roots(modelo.var)
```

```
## [1] 0.10575835 0.10575835 0.05551779
```

### Estabilidade dos parâmetros

```r
modelo.est = stability(modelo.var, type = "OLS-CUSUM")
plot(modelo.est)
```




## Test de correlação serial

```r
modelo.pt.asy = serial.test(modelo.var, lags.pt = 4, type = "PT.asymptotic")
modelo.pt.asy
```

```
##
##  Portmanteau Test (asymptotic)
##
## data:  Residuals of VAR object modelo.var
## Chi-squared = 27.275, df = 27, p-value = 0.4491
```

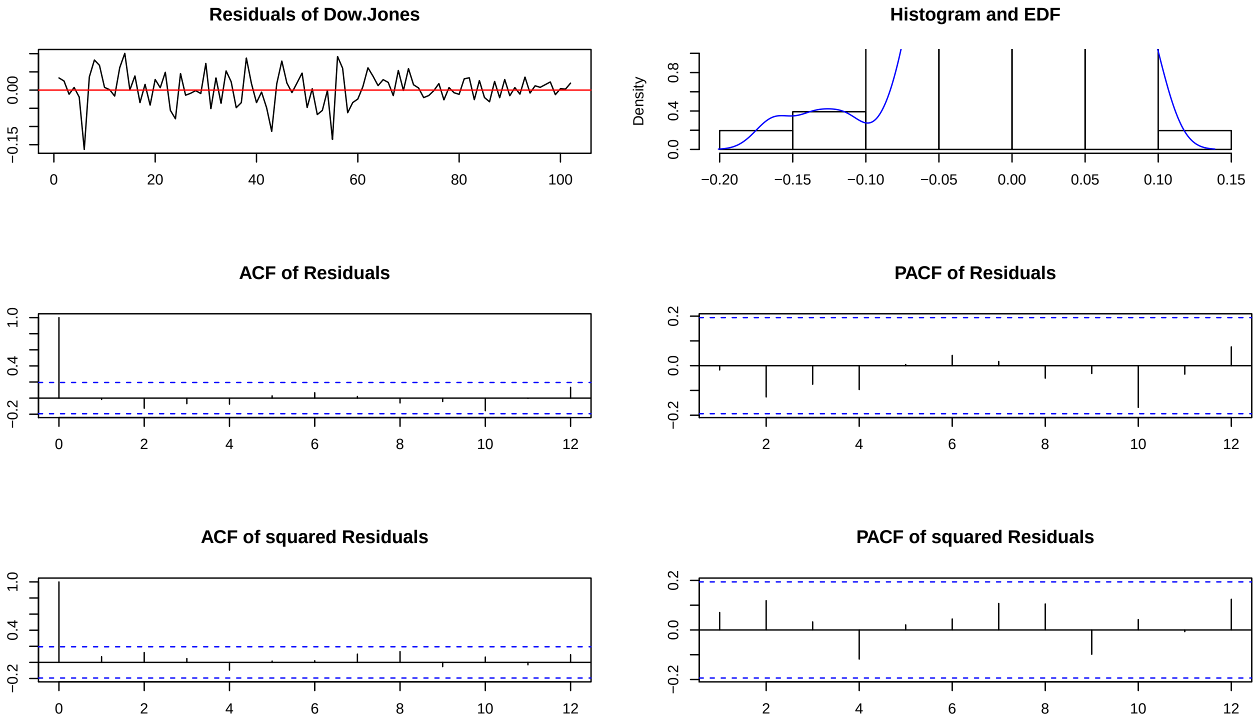
```r
modelo.pt.adj = serial.test(modelo.var, lags.pt = 8, type = "PT.adjusted")
modelo.pt.adj
```

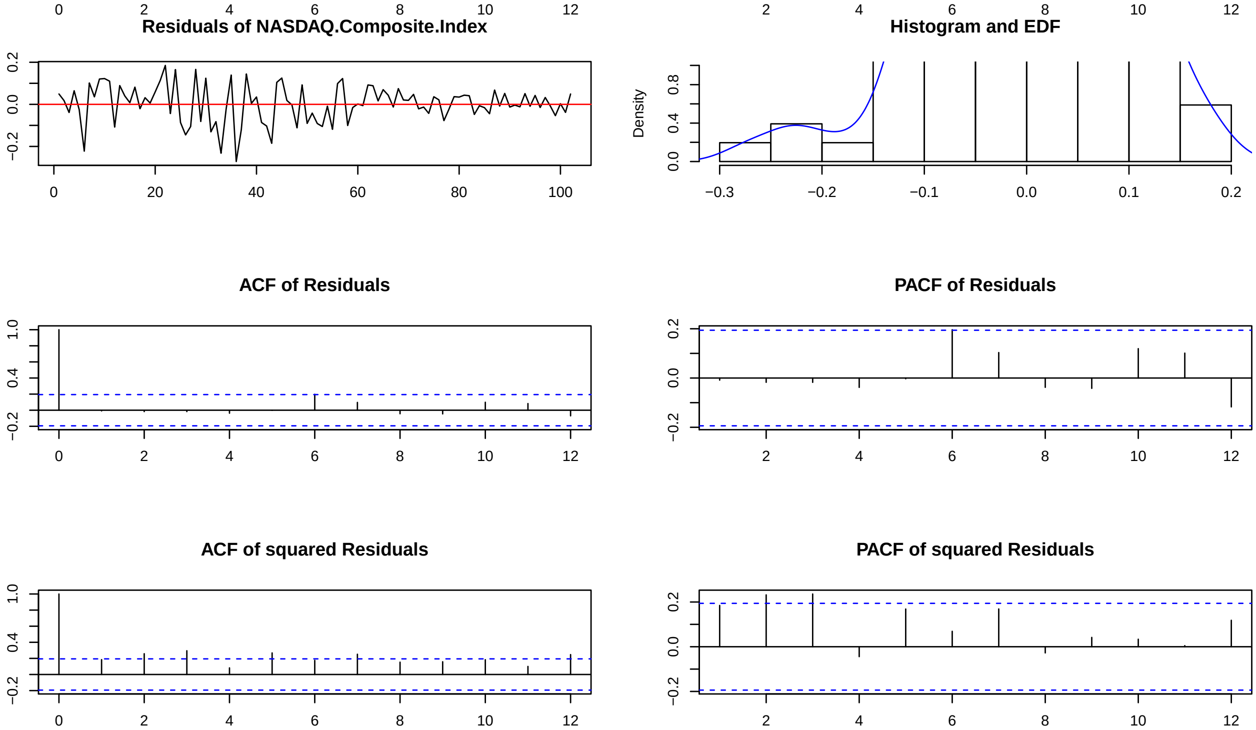
```
##
##  Portmanteau Test (adjusted)
##
## data:  Residuals of VAR object modelo.var
## Chi-squared = 67.017, df = 63, p-value = 0.341
```

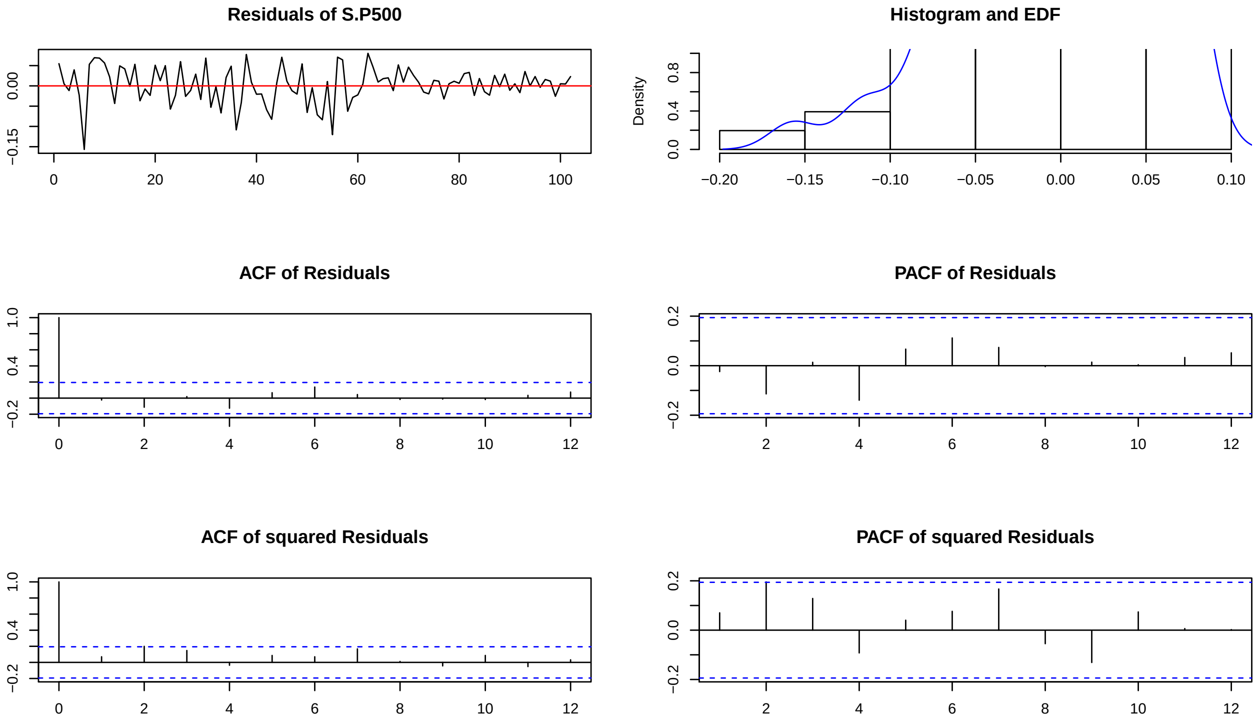
```r
modelo.bg = serial.test(modelo.var, lags.bg = 4, type = "BG")
modelo.bg
```

```
##
##  Breusch-Godfrey LM test
##
## data:  Residuals of VAR object modelo.var
## Chi-squared = 35.307, df = 36, p-value = 0.5014
```

```r
plot(modelo.bg)
```


Residuals of Dow.Jones
Histogram and EDF
Density
ACF of Residuals
PACF of Residuals
ACF of squared Residuals
PACF of squared Residuals


Residuals of NASDAQ.Composite.Index
Histogram and EDF
Density
ACF of Residuals
PACF of Residuals
ACF of squared Residuals
PACF of squared Residuals




## Test de Normalidade

```
modelo.norm = normality.test(modelo.var, multivariate.only = F)
modelo.norm
```

```
## $Dow.Jones
##
##  JB-Test (univariate)
##
## data:  Residual of Dow.Jones equation
## Chi-squared = 24.062, df = 2, p-value = 5.955e-06
##
##
## $NASDAQ.Composite.Index
##
##  JB-Test (univariate)
##
## data:  Residual of NASDAQ.Composite.Index equation
## Chi-squared = 5.6465, df = 2, p-value = 0.05941
##
##
## $S.P500
##
##  JB-Test (univariate)
##
## data:  Residual of S.P500 equation
## Chi-squared = 14.514, df = 2, p-value = 0.0007052
##
##
## $JB
```

```
##
##  JB-Test (multivariate)
##
## data:  Residuals of VAR object modelo.var
## Chi-squared = 89.947, df = 6, p-value < 2.2e-16
##
##
## $Skewness
##
##  Skewness only (multivariate)
##
## data:  Residuals of VAR object modelo.var
## Chi-squared = 13.006, df = 3, p-value = 0.004623
##
##
## $Kurtosis
##
##  Kurtosis only (multivariate)
##
## data:  Residuals of VAR object modelo.var
## Chi-squared = 76.941, df = 3, p-value < 2.2e-16
```

### Test de causalidade

```
grangertest(diff(log(nasdaq)) ~ diff(log(dj)), order = 1)
```

```
## Granger causality test
##
## Model 1: diff(log(nasdaq)) ~ Lags(diff(log(nasdaq)), 1:1) + Lags(diff(log(dj)), 1:1)
## Model 2: diff(log(nasdaq)) ~ Lags(diff(log(nasdaq)), 1:1)
##   Res.Df Df      F Pr(>F)
## 1     99
## 2    100 -1 2.1784 0.1431
```

```
grangertest(diff(log(dj)) ~ diff(log(nasdaq)), order = 1)
```

```
## Granger causality test
##
## Model 1: diff(log(dj)) ~ Lags(diff(log(dj)), 1:1) + Lags(diff(log(nasdaq)), 1:1)
## Model 2: diff(log(dj)) ~ Lags(diff(log(dj)), 1:1)
##   Res.Df Df      F Pr(>F)
## 1     99
## 2    100 -1 2.4655 0.1196
```

```
grangertest(diff(log(nasdaq)) ~ diff(log(sp)), order = 1)
```

```
## Granger causality test
##
## Model 1: diff(log(nasdaq)) ~ Lags(diff(log(nasdaq)), 1:1) + Lags(diff(log(sp)), 1:1)
## Model 2: diff(log(nasdaq)) ~ Lags(diff(log(nasdaq)), 1:1)
##   Res.Df Df      F Pr(>F)
## 1     99
## 2    100 -1 1.4179 0.2366
```

```r
grangertest(diff(log(sp)) ~ diff(log(nasdaq)), order = 1)
```

```
## Granger causality test
##
## Model 1: diff(log(sp)) ~ Lags(diff(log(sp)), 1:1) + Lags(diff(log(nasdaq)), 1:1)
## Model 2: diff(log(sp)) ~ Lags(diff(log(sp)), 1:1)
##   Res.Df Df      F Pr(>F)
## 1     99
## 2    100 -1 1.6623 0.2003
```

```r
grangertest(diff(log(dj)) ~ diff(log(sp)), order = 1)
```

```
## Granger causality test
##
## Model 1: diff(log(dj)) ~ Lags(diff(log(dj)), 1:1) + Lags(diff(log(sp)), 1:1)
## Model 2: diff(log(dj)) ~ Lags(diff(log(dj)), 1:1)
##   Res.Df Df      F Pr(>F)
## 1     99
## 2    100 -1 2.6688 0.1055
```

```r
grangertest(diff(log(sp)) ~ diff(log(dj)), order = 1)
```

```
## Granger causality test
##
## Model 1: diff(log(sp)) ~ Lags(diff(log(sp)), 1:1) + Lags(diff(log(dj)), 1:1)
## Model 2: diff(log(sp)) ~ Lags(diff(log(sp)), 1:1)
##   Res.Df Df      F  Pr(>F)
## 1     99
## 2    100 -1 3.1129 0.08076 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```