Vetor Auto-Regressivo - VAR

Henri Makika

May 21, 2019

Introdução

Modelo univariado auto-regressivo :

$$Y_t = \alpha_1 Y_{t-1} + \alpha_2 Y_{t-2} + ... + \alpha_p Y_{t-p} + \varepsilon_t$$

Sendo que :

$$E(\varepsilon_t) = 0; E(\varepsilon_t^2) = \sigma_\varepsilon^2; E(\varepsilon_t, \varepsilon_{t-j}) = 0$$

$E(\varepsilon_t, \varepsilon_{t-j}) = 0$, significa que os valores de ε_t são independentes. c'est à dire que ausência de correlação serial temporal. O VAR trata somente uma série que já está estacionário.

Modelo multivariado

1. Estudar o comportamento individual de uma série mas também as relações entre as séries;
2. Entender relações dinâmicas sob o tempo entre as séries;
3. Melhorar as previsões das séries individuais utilizando informações adicionais.

Modelo VAR irrestrito ou padrão

a. Proposto por Sims (1980);
b. Todas as variáveis são tratadas como sendo, a priori, endógenas;
c. Busca responder qual a trajetória da série, dado um choque estrutural.

VAR irrestrito: É um modelo auto-regressivo multivariado em que cada variável é expressa como função de suas defasagens e das defasagens das demais variáveis do modelo.

Modelo VAR(1) na forma padrão ou reduzida:

$$X_t = A_0 + A_1 X_{t-1} + e_t$$

em que :

$$\blacklozenge A_0 = B^{-1}\Gamma_0$$
$$\blacklozenge A_1 = B^{-1}\Gamma_1$$
$$\blacklozenge e_t = B^{-1}\varepsilon_t$$

Onde Γ_0 é o vetor de constantes n x 1; Γ_i matrizes n x n.

Packages nécessaires pour estimation du modèle VAR

```r
library(readxl)
library(urca)
library(MASS)
#library(MTS) # à installer
library(vars)
library(lmtest)
```

Données à utiliser

```r
data(Canada)
head(Canada)
```

```
## [1] 929.6105 929.8040 930.3184 931.4277 932.6620 933.5509
```

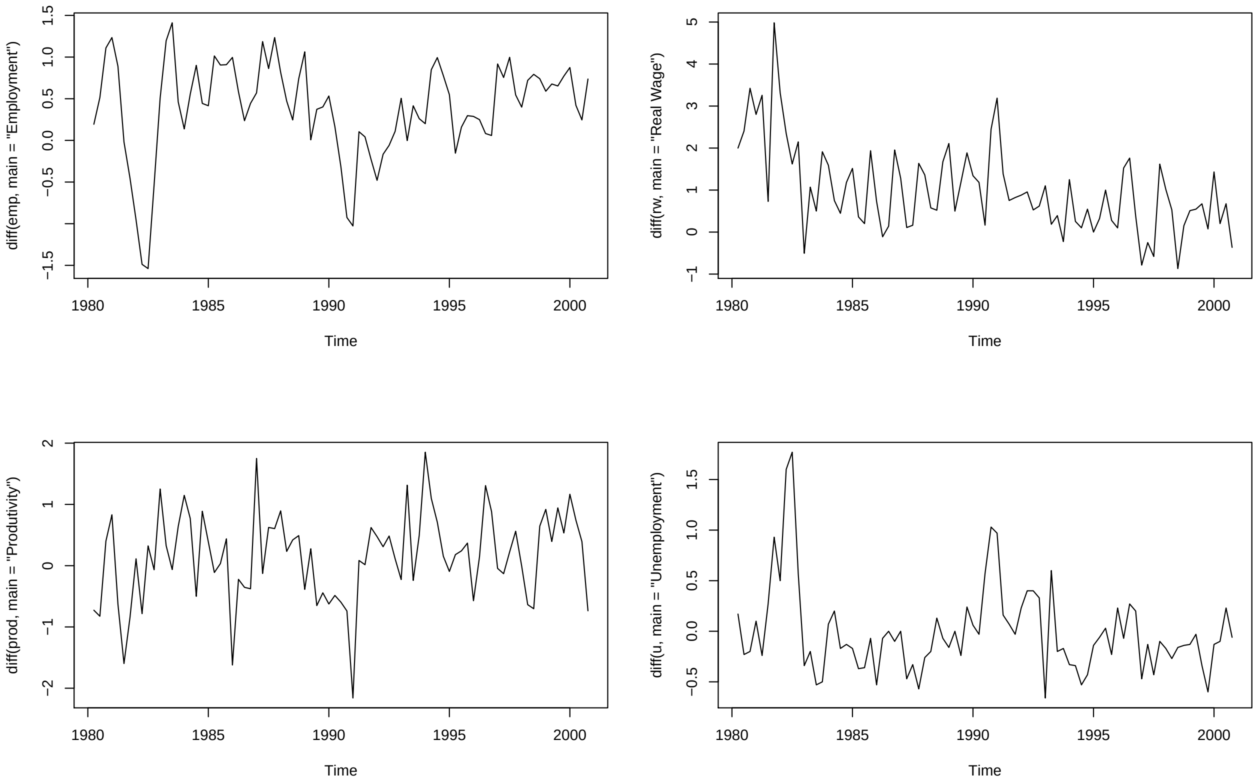
```r
View(Canada)
```

Visualisation de données dans un graphique

```r
emp = Canada[,1]
prod = Canada[,2]
rw = Canada[,3]
u = Canada[,4]
```

Les données étant non stationnaire, nous travaillons en mettant les données en différence, puisque le modèle VAR veut que les données soient stationnaire.

```r
layout(matrix(1:4, nrow = 2, ncol = 2))
plot.ts(diff(emp, main = "Employment"))
plot.ts(diff(prod, main = "Produtivity"))
plot.ts(diff(rw, main = "Real Wage"))
plot.ts(diff(u, main = "Unemployment"))
```



Identification du modèle VAR

```r
ordem = VARselect(diff(Canada), lag.max = 8, type = "const")
ordem
```

```
## $selection
## AIC(n)  HQ(n)  SC(n) FPE(n)
##      2      1      1      2
##
## $criteria
##                   1            2           3            4            5
## AIC(n) -6.270032208 -6.285458074 -6.03818873 -6.035665365 -5.808725091
## HQ(n)  -6.023272860 -5.841291247 -5.39661443 -5.196683581 -4.772335828
## SC(n)  -5.652035378 -5.173063780 -4.43139698 -3.934476142 -3.213138403
## FPE(n)  0.001893667  0.001871884  0.00241986  0.002469804  0.003191486
##                   6            7           8
## AIC(n) -5.558322503 -5.370524857 -5.33131938
## HQ(n)  -4.324525761 -3.939320637 -3.70270768
## SC(n)  -2.468338351 -1.786143241 -1.25254030
## FPE(n)  0.004286004  0.005511821  0.00626064
```

Estimation du Modèle VAR

```r
modelo = VAR(diff(Canada), type = "const", p = 2)
modelo
```

```
##
```

```
## VAR Estimation Results:
## =======================
##
## Estimated coefficients for equation e:
## ======================================
## Call:
## e = e.l1 + prod.l1 + rw.l1 + U.l1 + e.l2 + prod.l2 + rw.l2 + U.l2 + const
##
##        e.l1     prod.l1       rw.l1        U.l1        e.l2     prod.l2
##  0.92480319  0.17822020 -0.03216836  0.08640312 -0.37185180  0.02248112
##       rw.l2        U.l2       const
## -0.04652263 -0.06662069  0.22248016
##
##
## Estimated coefficients for equation prod:
## =========================================
## Call:
## prod = e.l1 + prod.l1 + rw.l1 + U.l1 + e.l2 + prod.l2 + rw.l2 + U.l2 + const
##
##        e.l1     prod.l1       rw.l1        U.l1        e.l2     prod.l2
## -0.16719423  0.21696976  0.04030841 -0.91605984 -0.51296201 -0.04934238
##       rw.l2        U.l2       const
## -0.15236513 -0.09982257  0.50562969
##
##
## Estimated coefficients for equation rw:
## =======================================
## Call:
## rw = e.l1 + prod.l1 + rw.l1 + U.l1 + e.l2 + prod.l2 + rw.l2 + U.l2 + const
##
##        e.l1     prod.l1       rw.l1        U.l1        e.l2     prod.l2
## -0.07297393 -0.19803957  0.23966978  0.51434794  0.55858997 -0.39210944
##       rw.l2        U.l2       const
##  0.11061079 -0.03004262  0.53043653
##
##
## Estimated coefficients for equation U:
## ======================================
## Call:
## U = e.l1 + prod.l1 + rw.l1 + U.l1 + e.l2 + prod.l2 + rw.l2 + U.l2 + const
##
##        e.l1     prod.l1       rw.l1        U.l1        e.l2     prod.l2
## -0.58955444 -0.15153110  0.04296745 -0.14505493  0.02900276 -0.01722080
##       rw.l2        U.l2       const
##  0.10771815 -0.24226795  0.07836532
```

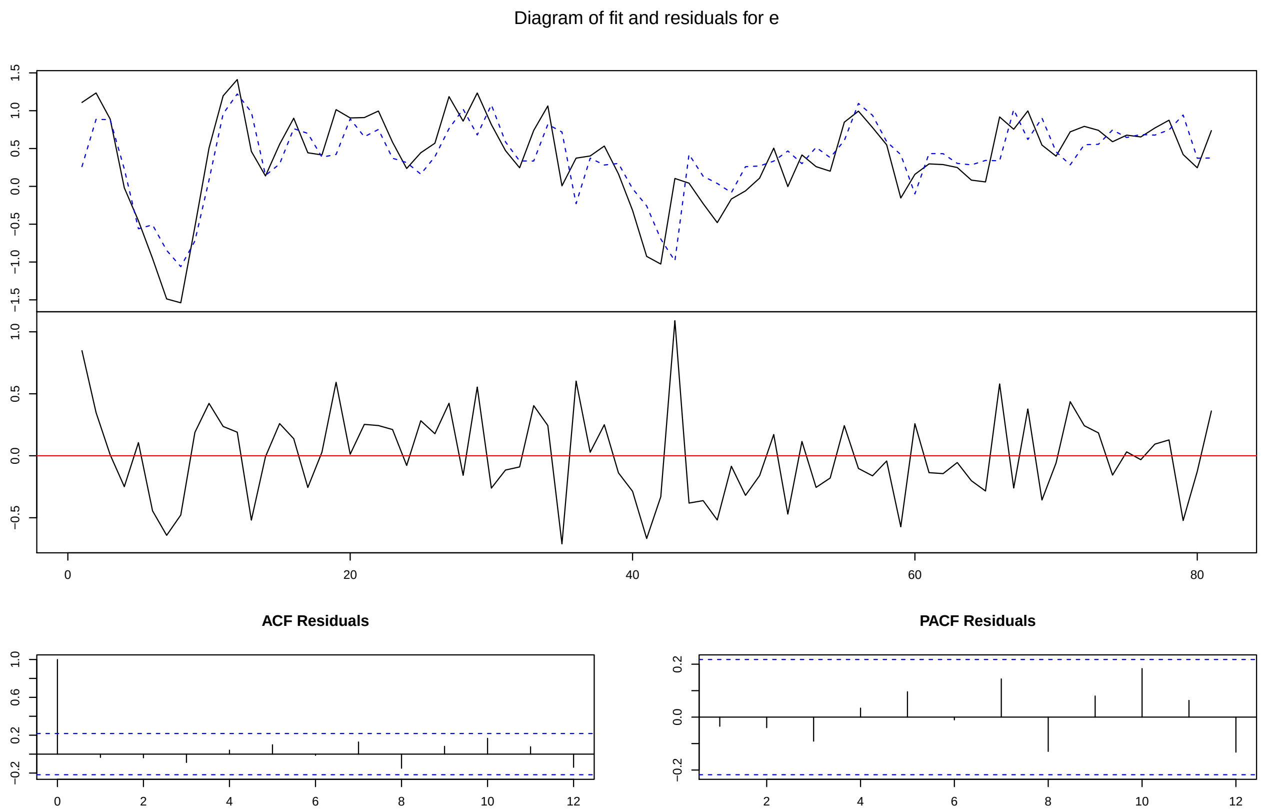
Resumo por equação

```r
summary(modelo, equation = "e")
```

```
##
## VAR Estimation Results:
## =========================
```

```
## Endogenous variables: e, prod, rw, U
## Deterministic variables: const
## Sample size: 81
## Log Likelihood: -186.088
## Roots of the characteristic polynomial:
## 0.6801 0.6801 0.5621 0.5621 0.4314 0.4314  0.33 0.3171
## Call:
## VAR(y = diff(Canada), p = 2, type = "const")
##
##
## Estimation results for equation e:
## ==================================
## e = e.l1 + prod.l1 + rw.l1 + U.l1 + e.l2 + prod.l2 + rw.l2 + U.l2 + const
##
##         Estimate Std. Error t value Pr(>|t|)
## e.l1     0.92480    0.15232   6.071  5.4e-08 ***
## prod.l1  0.17822    0.06342   2.810  0.00637 **
## rw.l1   -0.03217    0.04782  -0.673  0.50325
## U.l1     0.08640    0.19356   0.446  0.65666
## e.l2    -0.37185    0.16360  -2.273  0.02602 *
## prod.l2  0.02248    0.06497   0.346  0.73034
## rw.l2   -0.04652    0.04678  -0.995  0.32330
## U.l2    -0.06662    0.20198  -0.330  0.74248
## const    0.22248    0.09380   2.372  0.02038 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.3698 on 72 degrees of freedom
## Multiple R-Squared: 0.6476,  Adjusted R-squared: 0.6084
## F-statistic: 16.54 on 8 and 72 DF,  p-value: 1.282e-13
##
##
##
## Covariance matrix of residuals:
##              e       prod        rw          U
## e     0.136736 -0.0178671 -0.009849 -0.0738072
## prod -0.017867  0.4258636  0.059273  0.0003443
## rw   -0.009849  0.0592729  0.764699  0.0559236
## U    -0.073807  0.0003443  0.055924  0.0861512
##
## Correlation matrix of residuals:
##             e      prod       rw         U
## e     1.00000 -0.074042 -0.03046 -0.680027
## prod -0.07404  1.000000  0.10387  0.001798
## rw   -0.03046  0.103866  1.00000  0.217881
## U    -0.68003  0.001798  0.21788  1.000000
```

```r
plot(modelo, names = "e")
```



Raíses

```r
roots(modelo, modulus = F)
```

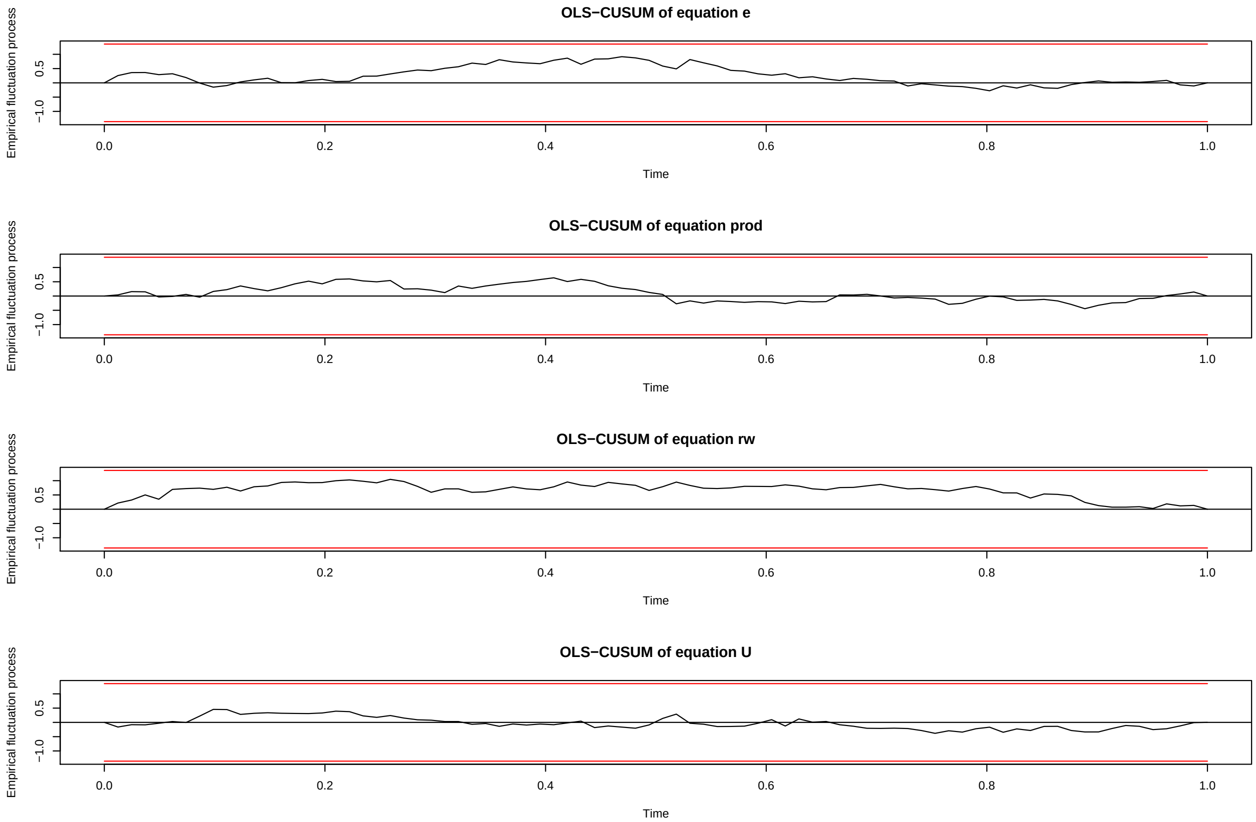
```
## [1]  0.6306181+0.2546095i  0.6306181-0.2546095i  0.1286084+0.5472355i
## [4]  0.1286084-0.5472355i -0.1474887+0.4053708i -0.1474887-0.4053708i
## [7]  0.3300337+0.0000000i -0.3171215+0.0000000i
```

```r
roots(modelo)
```

```
## [1] 0.6800773 0.6800773 0.5621449 0.5621449 0.4313680 0.4313680 0.3300337
## [8] 0.3171215
```

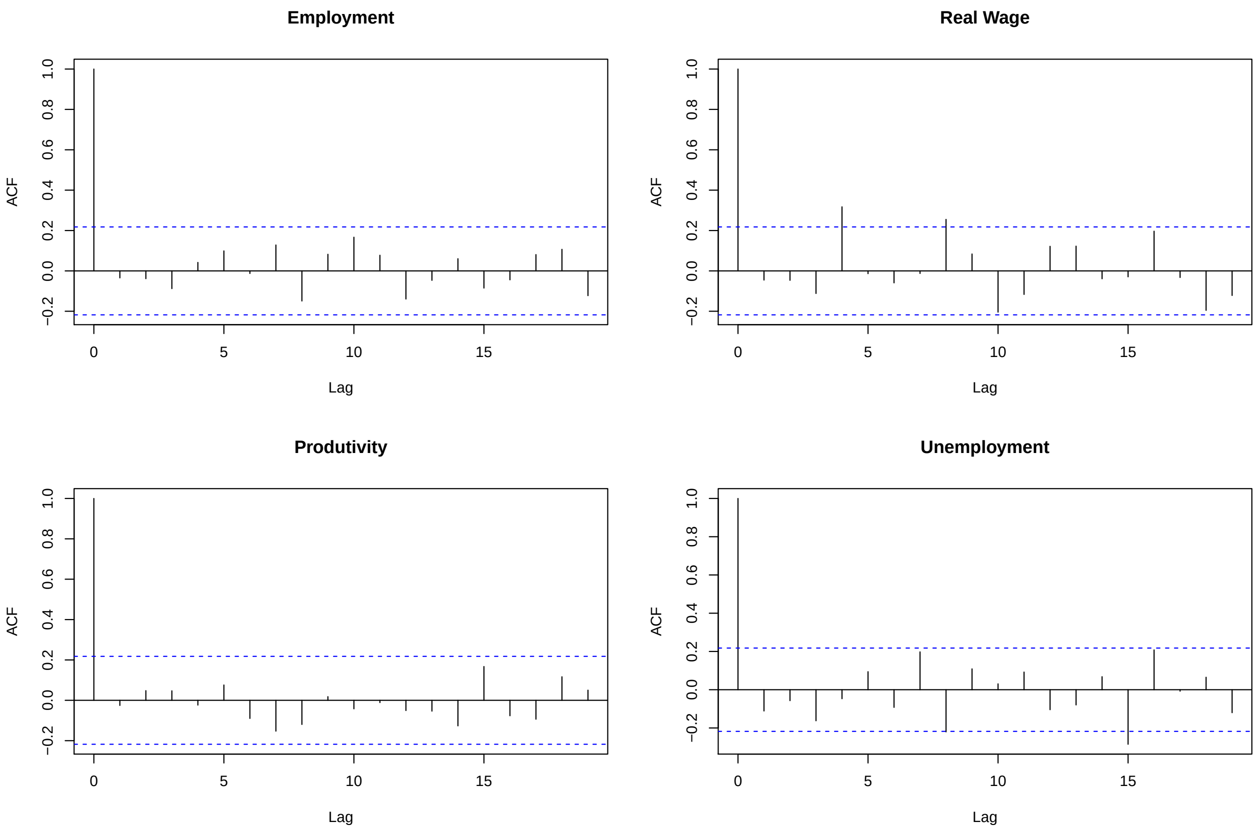
Estabilidade do modelo

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



Autocorrelação dos resíduos

```
layout(matrix(1:4, nrow = 2, ncol = 2))
acf(residuals(modelo)[,1], main = "Employment")
acf(residuals(modelo)[,2], main = "Produtivity")
acf(residuals(modelo)[,3], main = "Real Wage")
acf(residuals(modelo)[,4], main = "Unemployment")
```

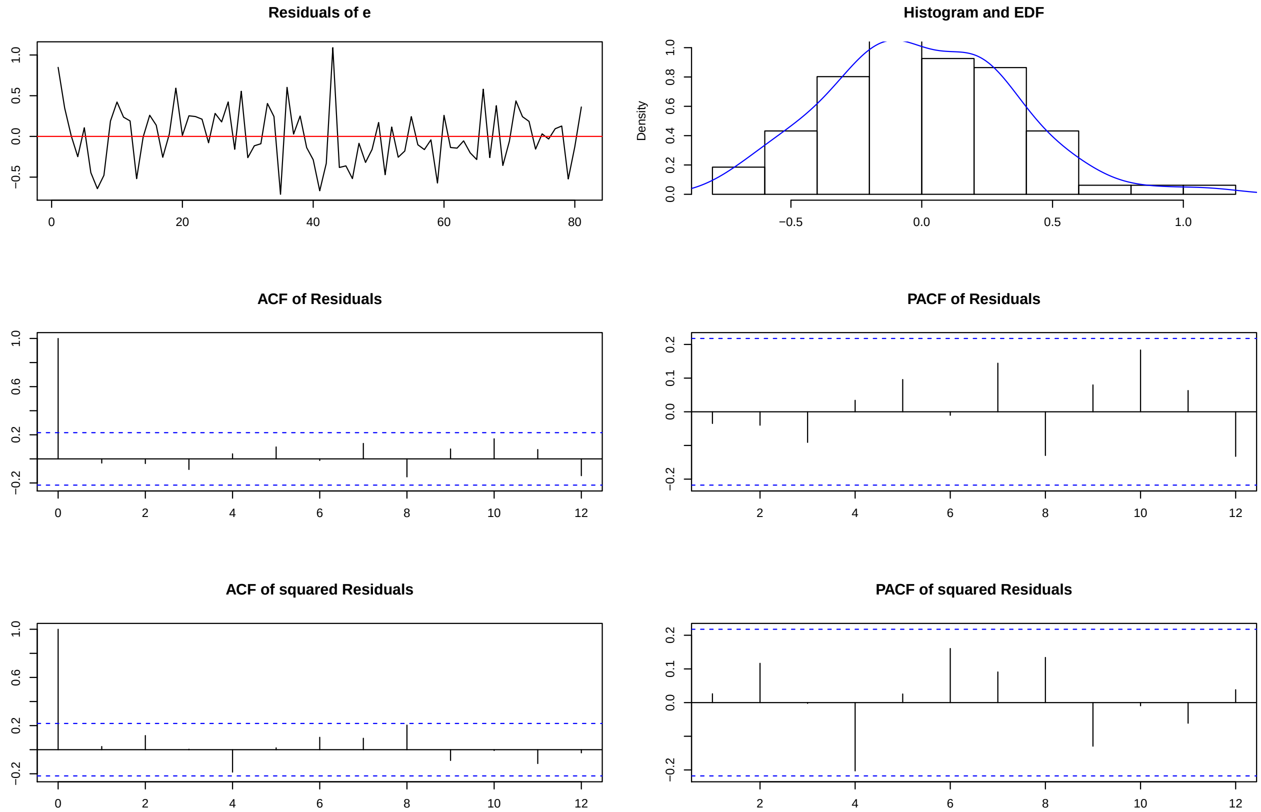


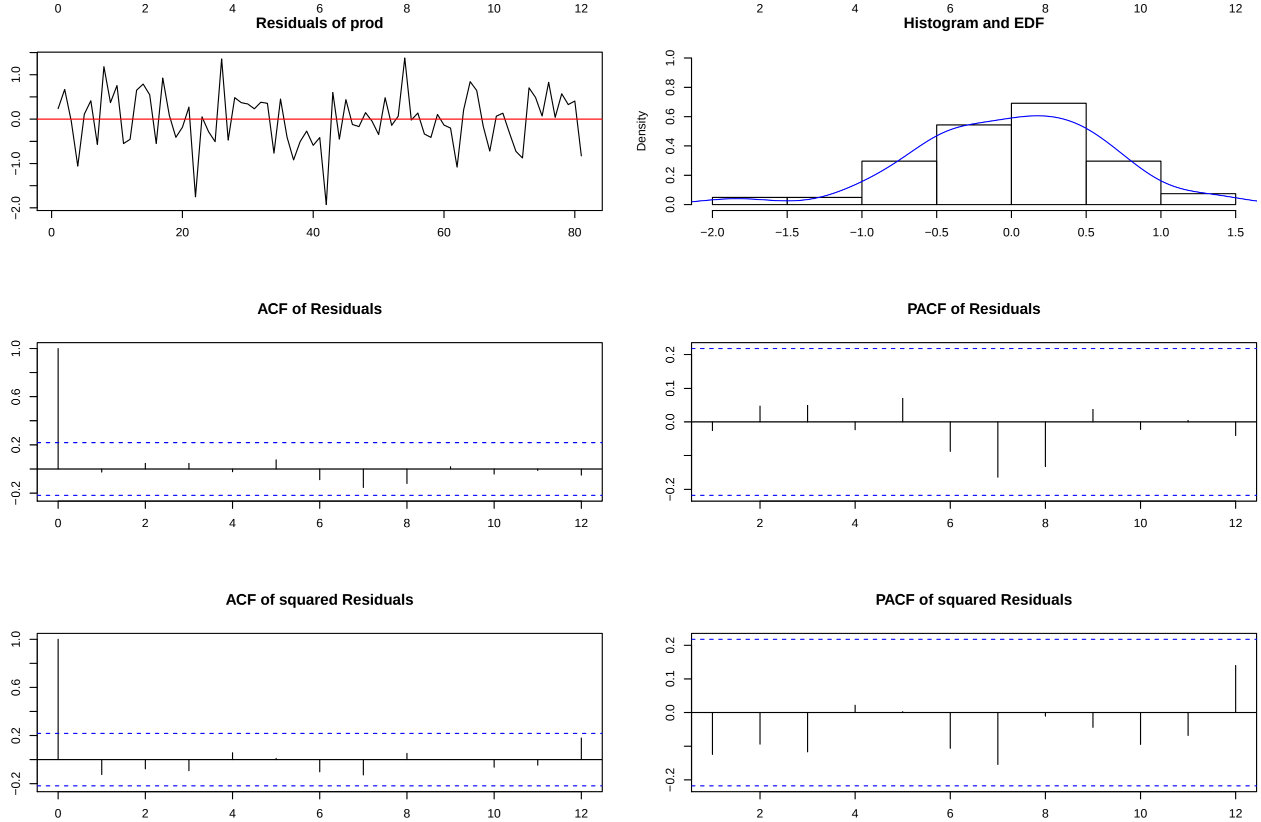
Test de Portemanteau

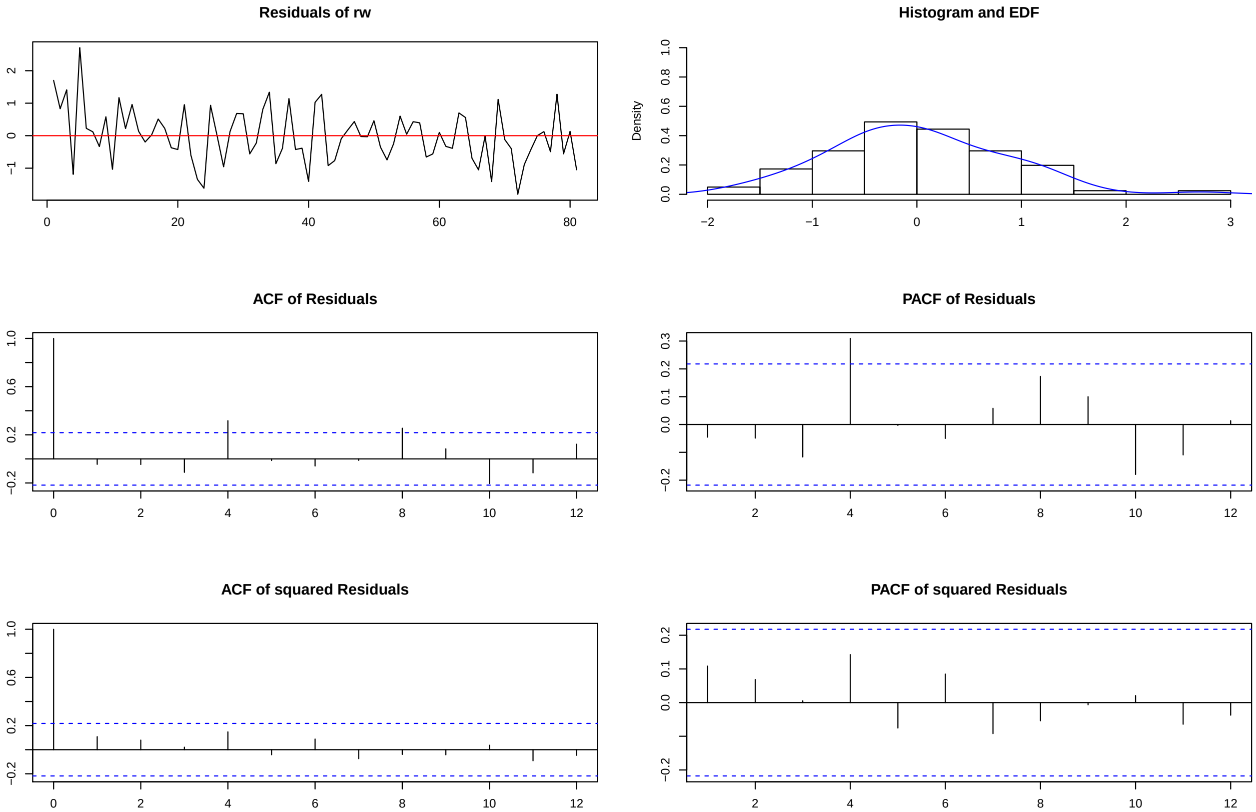
```
model.pt.asy = serial.test(modelo, lags.pt = 12, type = "PT.asymptotic")
model.pt.asy
```

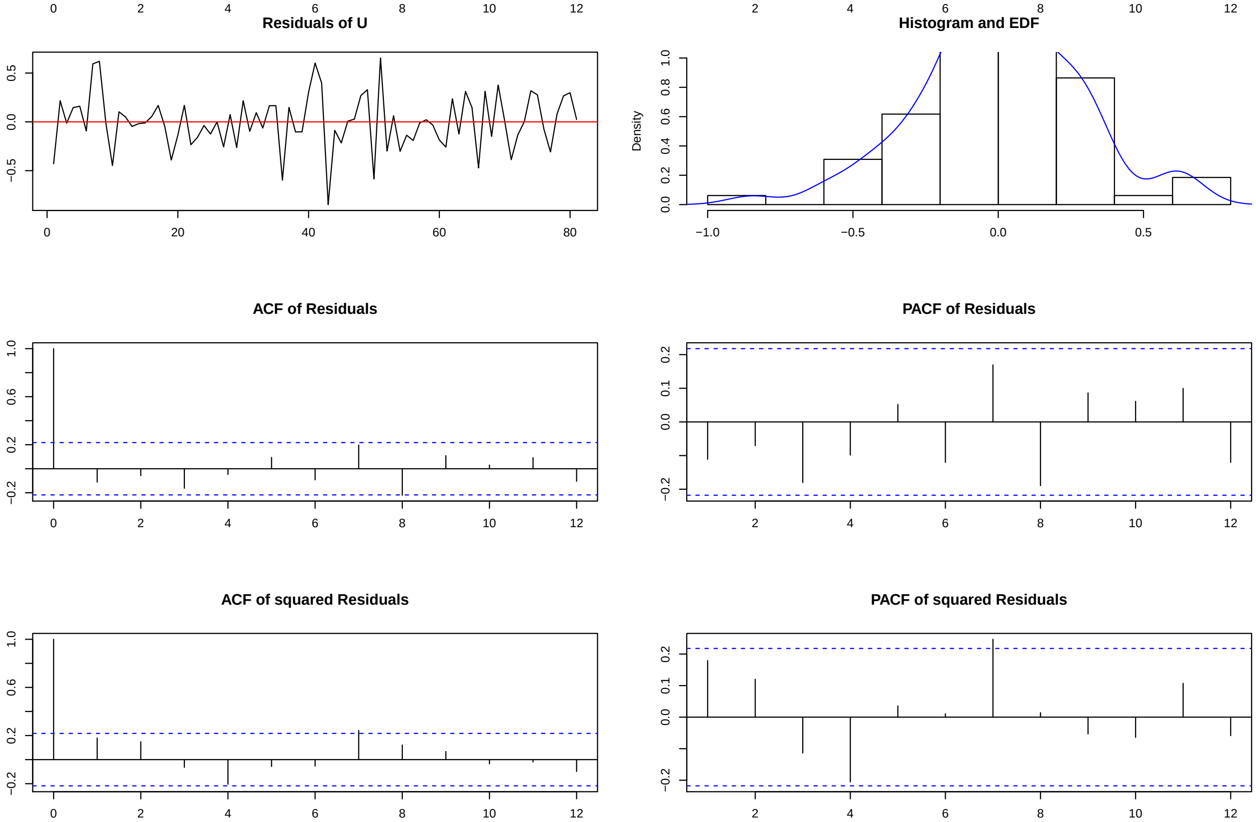
```
##
##  Portmanteau Test (asymptotic)
##
## data:  Residuals of VAR object modelo
## Chi-squared = 123.2, df = 160, p-value = 0.9862
```

```
plot(model.pt.asy)
```


Residuals of e
Histogram and EDF
Density
ACF of Residuals
PACF of Residuals
ACF of squared Residuals
PACF of squared Residuals


Residuals of prod
Histogram and EDF
Density
ACF of Residuals
PACF of Residuals
ACF of squared Residuals
PACF of squared Residuals


Residuals of rw
Histogram and EDF
Density
ACF of Residuals
PACF of Residuals
ACF of squared Residuals
PACF of squared Residuals

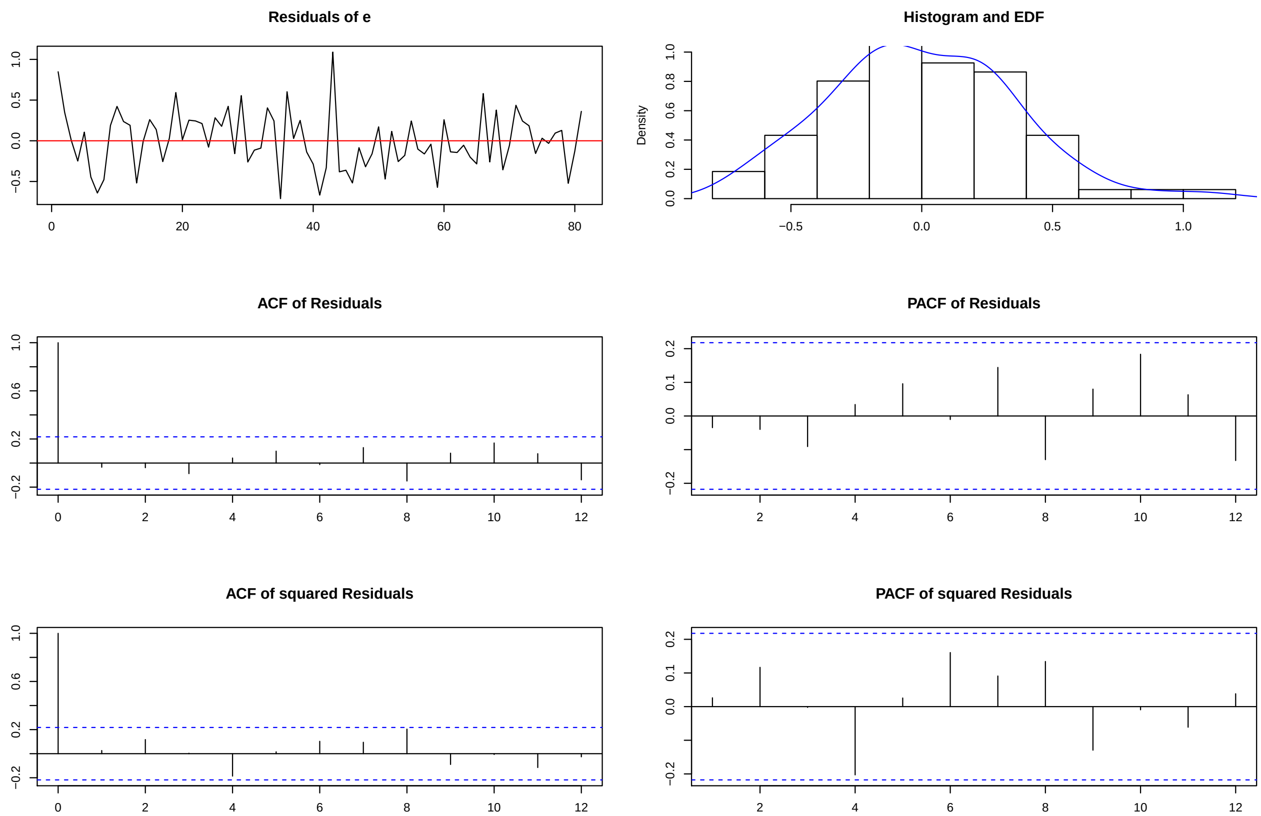

Residuals of U
Histogram and EDF
Density
ACF of Residuals
PACF of Residuals
ACF of squared Residuals
PACF of squared Residuals

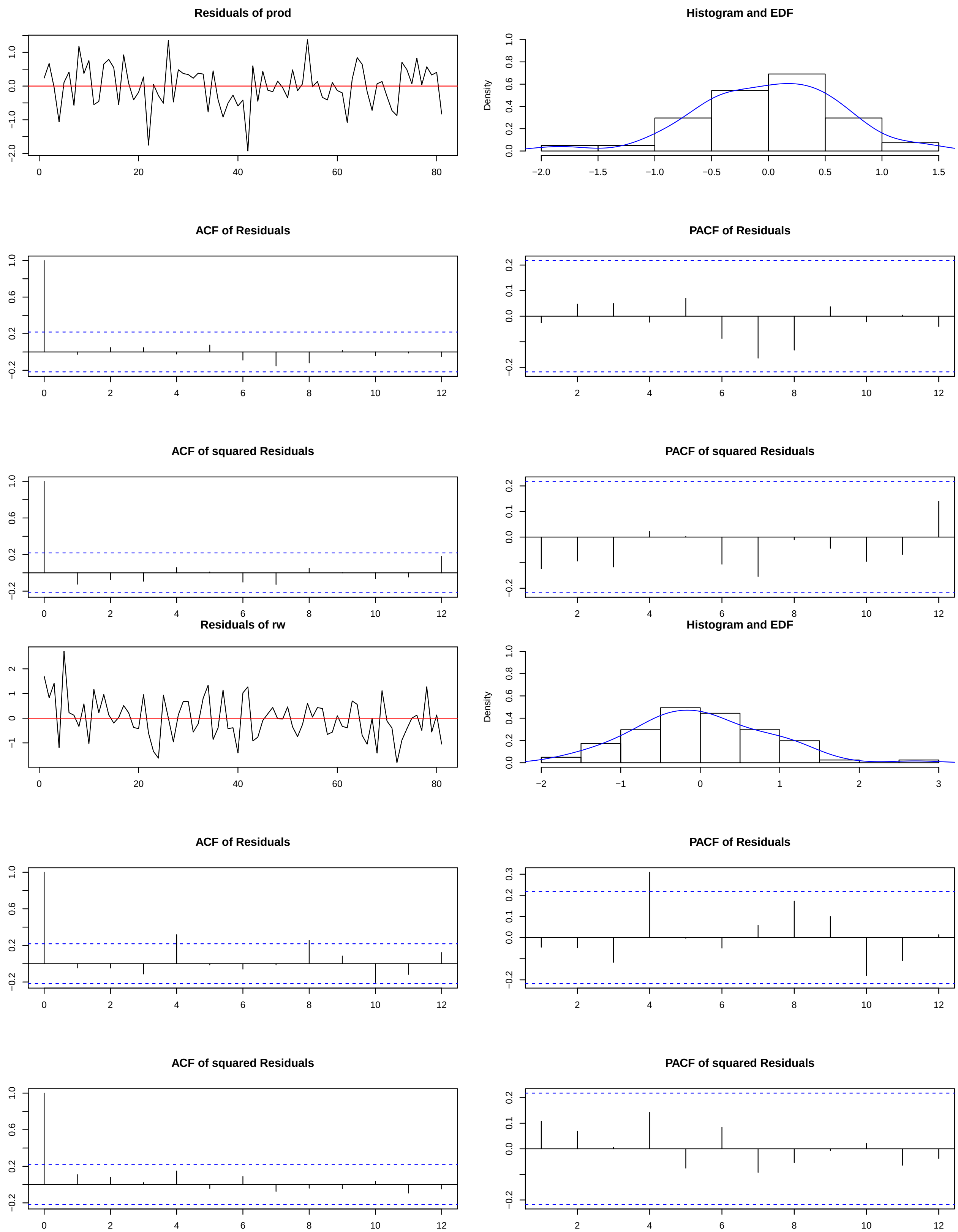
Test Ljung-Box

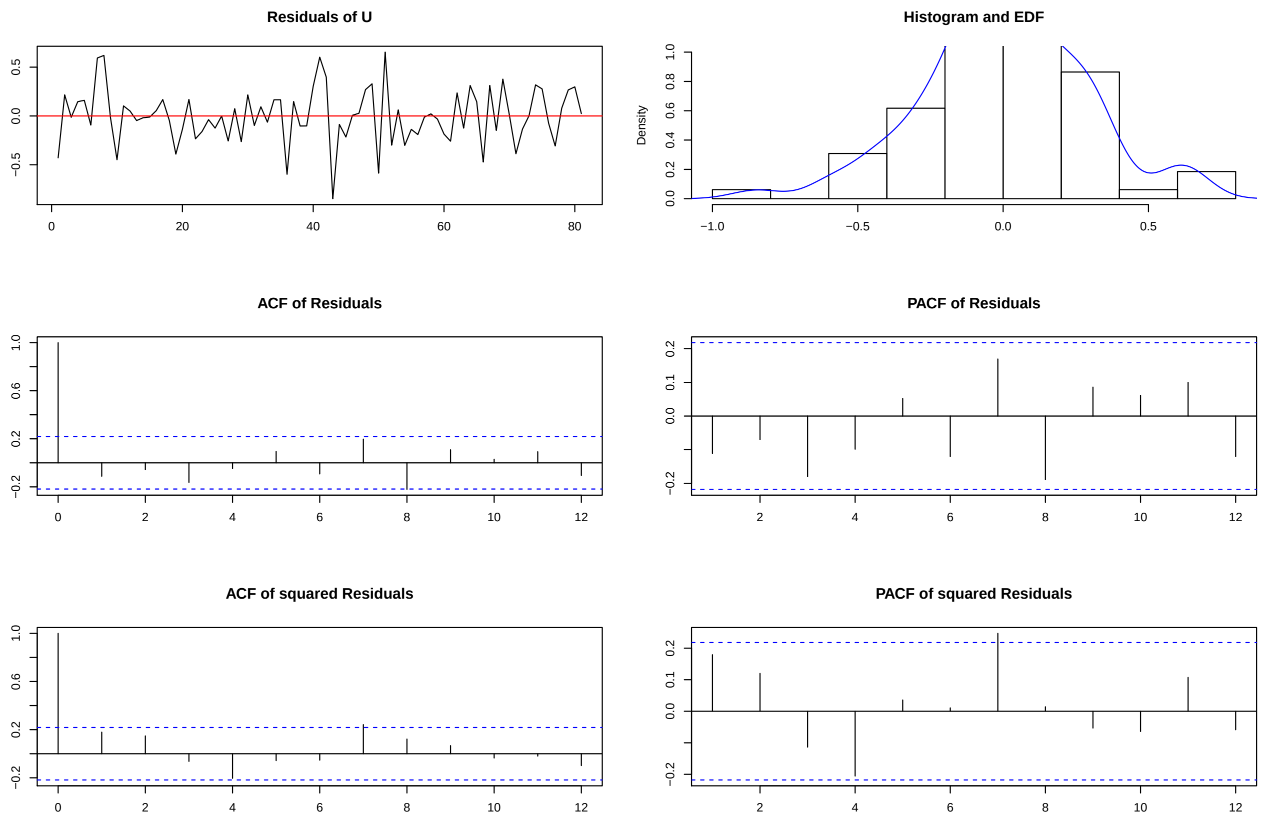
```r
model.pt.adj = serial.test(modelo, lags.pt = 12, type = "PT.adjusted")
model.pt.adj
```

```
##
##  Portmanteau Test (adjusted)
##
## data:  Residuals of VAR object modelo
## Chi-squared = 134.8, df = 160, p-value = 0.9268
```

```r
plot(model.pt.adj)
```




Residuals of prod
Histogram and EDF
Density
ACF of Residuals
PACF of Residuals
ACF of squared Residuals
PACF of squared Residuals
Residuals of rw
Histogram and EDF
Density
ACF of Residuals
PACF of Residuals
ACF of squared Residuals
PACF of squared Residuals



Test LM

```
modelo.BG = serial.test(modelo, lags.bg = 12, type = "BG")
modelo.BG
```

```
##
##  Breusch-Godfrey LM test
##
## data:  Residuals of VAR object modelo
## Chi-squared = 198.86, df = 192, p-value = 0.352
```

Test de normalidad

```
model.norm = normality.test(modelo, multivariate.only = FALSE)
model.norm
```

```
## $e
##
##  JB-Test (univariate)
##
## data:  Residual of e equation
## Chi-squared = 1.5858, df = 2, p-value = 0.4525
##
##
## $prod
##
```

```
##  JB-Test (univariate)
##
## data:  Residual of prod equation
## Chi-squared = 3.4993, df = 2, p-value = 0.1738
##
##
## $rw
##
##  JB-Test (univariate)
##
## data:  Residual of rw equation
## Chi-squared = 2.0587, df = 2, p-value = 0.3572
##
##
## $U
##
##  JB-Test (univariate)
##
## data:  Residual of U equation
## Chi-squared = 1.3656, df = 2, p-value = 0.5052
##
##
## $JB
##
##  JB-Test (multivariate)
##
## data:  Residuals of VAR object modelo
## Chi-squared = 9.8861, df = 8, p-value = 0.2731
##
##
## $Skewness
##
##  Skewness only (multivariate)
##
## data:  Residuals of VAR object modelo
## Chi-squared = 7.6138, df = 4, p-value = 0.1068
##
##
## $Kurtosis
##
##  Kurtosis only (multivariate)
##
## data:  Residuals of VAR object modelo
## Chi-squared = 2.2723, df = 4, p-value = 0.6858
```

Mudança de ordenação

```
Canada2 = Canada[,c(3, 1, 2, 4)]
View(Canada2)

modelo.alt = VAR(diff(Canada2), p = 2, type = "const")
```

Test de normalidade apos mudança

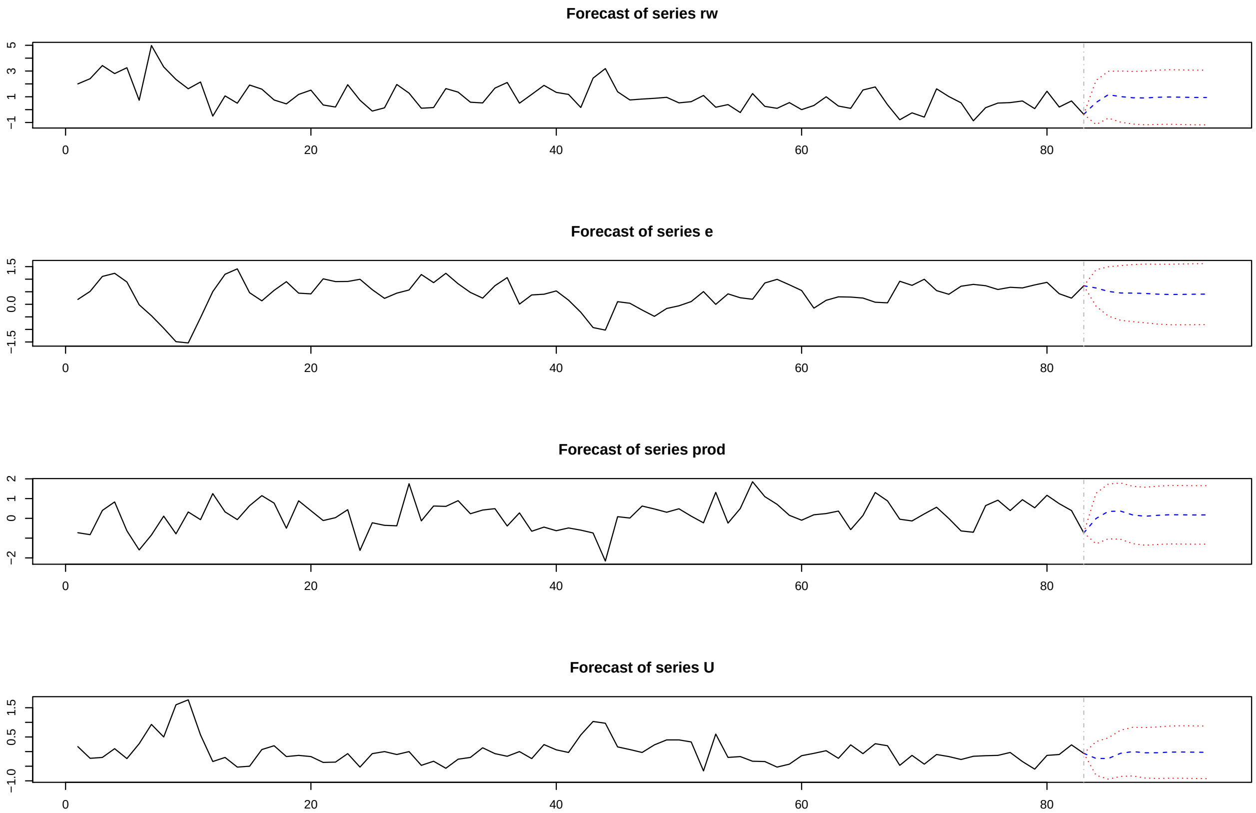
```r
model.norm2 = normality.test(modelo.alt, multivariate.only = FALSE)
model.norm2
```

```
## $rw
##
##  JB-Test (univariate)
##
## data:  Residual of rw equation
## Chi-squared = 2.0587, df = 2, p-value = 0.3572
##
##
## $e
##
##  JB-Test (univariate)
##
## data:  Residual of e equation
## Chi-squared = 1.5858, df = 2, p-value = 0.4525
##
##
## $prod
##
##  JB-Test (univariate)
##
## data:  Residual of prod equation
## Chi-squared = 3.4993, df = 2, p-value = 0.1738
##
##
## $U
##
##  JB-Test (univariate)
##
## data:  Residual of U equation
## Chi-squared = 1.3656, df = 2, p-value = 0.5052
##
##
## $JB
##
##  JB-Test (multivariate)
##
## data:  Residuals of VAR object modelo.alt
## Chi-squared = 10.17, df = 8, p-value = 0.2533
##
##
## $Skewness
##
##  Skewness only (multivariate)
##
## data:  Residuals of VAR object modelo.alt
## Chi-squared = 6.8369, df = 4, p-value = 0.1448
##
##
## $Kurtosis
```

```
##
##  Kurtosis only (multivariate)
##
## data:  Residuals of VAR object modelo.alt
## Chi-squared = 3.333, df = 4, p-value = 0.5037
```

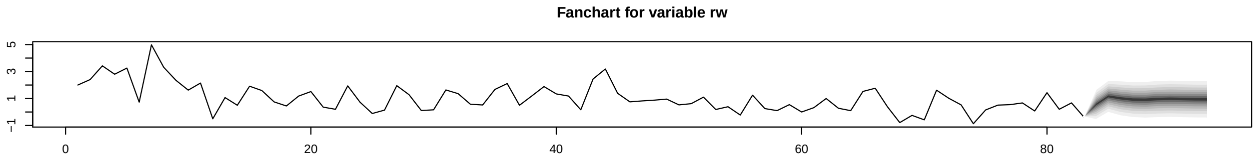
Previsão

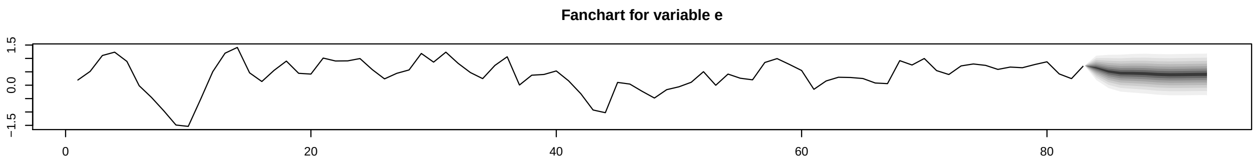
```r
modelo.forec = predict(modelo.alt, n.ahead = 10, ci = 0.95)
plot(modelo.forec)
```

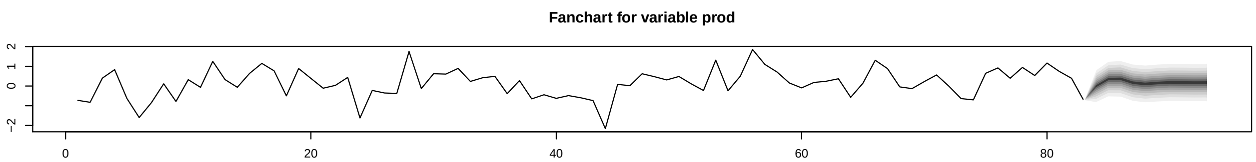


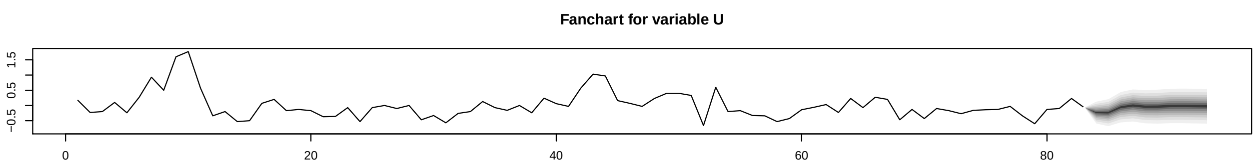
Graphique de prévision là dessus.

```r
fanchart(modelo.forec)
```









Test de Causalidade de Granger

```
grangertest(diff(e) ~ diff(rw), order = 8, data = Canada2)
```

```
## Granger causality test
##
## Model 1: diff(e) ~ Lags(diff(e), 1:8) + Lags(diff(rw), 1:8)
## Model 2: diff(e) ~ Lags(diff(e), 1:8)
##   Res.Df Df     F Pr(>F)
## 1     58
## 2     66 -8 0.721 0.6722
```

Le salaire au sens de granger ne cause pas la variation de l'emploi. Ou ne contribue pas pour prévoir l'emploi.

```
model.causal = causality(modelo.alt, cause = c("rw", "prod", "e"))
model.causal
```

```
## $Granger
##
##  Granger causality H0: rw e prod do not Granger-cause U
##
## data:  VAR object modelo.alt
## F-Test = 8.0995, df1 = 6, df2 = 288, p-value = 4.244e-08
##
##
## $Instant
##
##  H0: No instantaneous causality between: rw e prod and U
```

```
##
## data:  VAR object modelo.alt
## Chi-squared = 27.22, df = 3, p-value = 5.295e-06
```

On rejet l'hypothèse nulle portante causa.

```
model.causal1 = causality(modelo.alt, cause = c("rw", "prod", "e"),
                         vcov. = vcovHC(modelo.alt))
model.causal1
```

```
## $Granger
##
##  Granger causality H0: rw e prod do not Granger-cause U
##
## data:  VAR object modelo.alt
## F-Test = 3.7549, df1 = 6, df2 = 288, p-value = 0.001294
##
##
## $Instant
##
##  H0: No instantaneous causality between: rw e prod and U
##
## data:  VAR object modelo.alt
## Chi-squared = 27.22, df = 3, p-value = 5.295e-06
```