Indice d’Activité Économique (IBC)-Brazil

Henri Makika

May 14, 2019

Packages importants pour cette analyse

```
library(forecast) # Para ajuste e previsão do modelo ARIMA
library(lmtest) # Por test de hipótese
library(FinTS) # Para o test de heterocedasticidade
library(urca) # Por test de Raiz Unitária
library(tseries) # Para fazer o test de normalidade
library(TSA) # Test de lag
library(readxl)
```

Lecture de données

```
ibc_bra = read_excel("~/Videos/Unicamp_IE 2019/HO:236A Times Series/Aula 8/IBC_BR.xlsx")
```

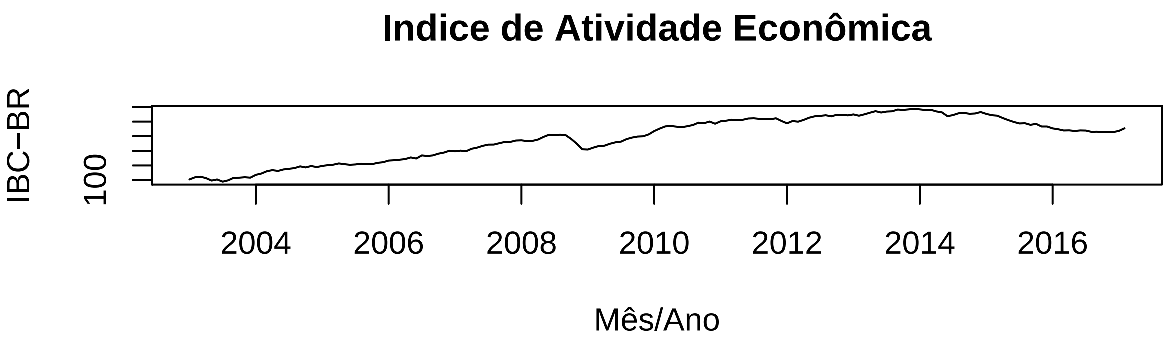
Analise de dados

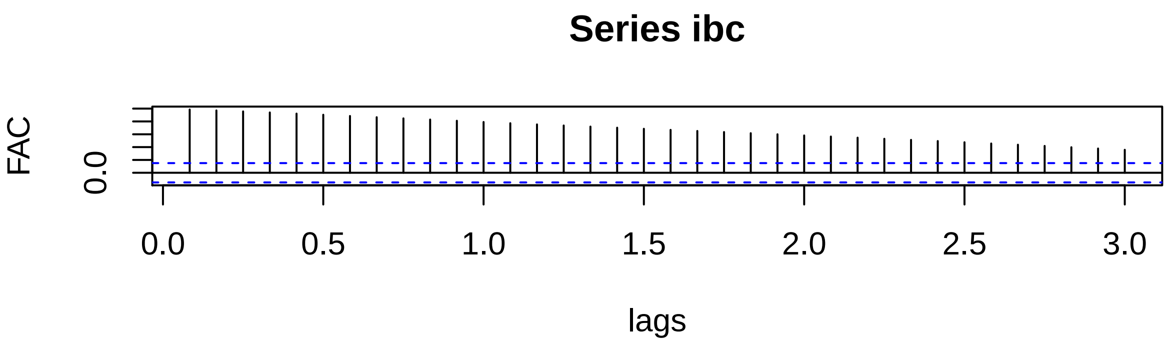
```
ibc = ts(ibc_bra[,2], start = c(2003, 1), frequency = 12)
```

Gráfico da série

```
layout(1:2) # par(mfrow = c(2, 1))

plot(ibc, main = "Indice de Atividade Econômica", xlab = "Mês/Ano",
     ylab = "IBC-BR")
acf(ibc, lag.max = 36, drop.lag.0 = TRUE, type = "correlation",
    xlab = "lags", ylab = "FAC")
```





Notre serie a une tendance, et elle est aussi non stationnaire. En 2008 et 2009 à cause de la crise financière, nous remarquons un choc négatif dans notre graphique de la série. Mais est-il possible de conclure qu'il s'agit là de quebra estruturalle ?

Test de quebra estrutural

Test de Raiz unitário

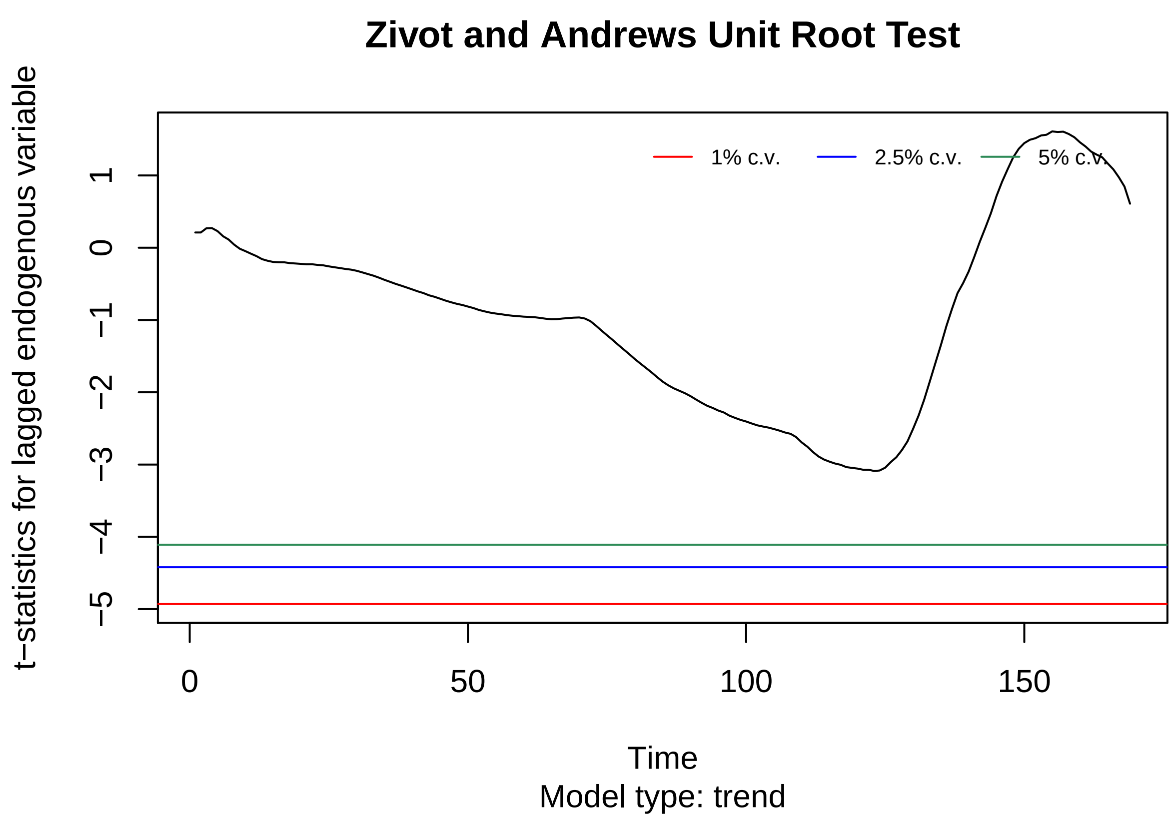
1. Test de Zivot and Andrews

```
za.ibc = ur.za(ibc, model = "trend")
summary(za.ibc)
```

```
##
## ################################
## # Zivot-Andrews Unit Root Test #
## ################################
##
##
## Call:
## lm(formula = testmat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.3965 -0.5382  0.1329  0.5963  2.2303
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)  9.47706    2.91924   3.246 0.001415 **
## y.l1         0.90885    0.02951  30.798  < 2e-16 ***
## trend        0.03462    0.01212   2.857 0.004830 **
## dt          -0.07783    0.02281  -3.412 0.000811 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9734 on 165 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.9956, Adjusted R-squared:  0.9955
## F-statistic: 1.243e+04 on 3 and 165 DF,  p-value: < 2.2e-16
##
##
## Teststatistic: -3.0888
## Critical values: 0.01= -4.93 0.05= -4.42 0.1= -4.11
##
## Potential break point at position: 123
```

```r
plot(za.ibc)
```

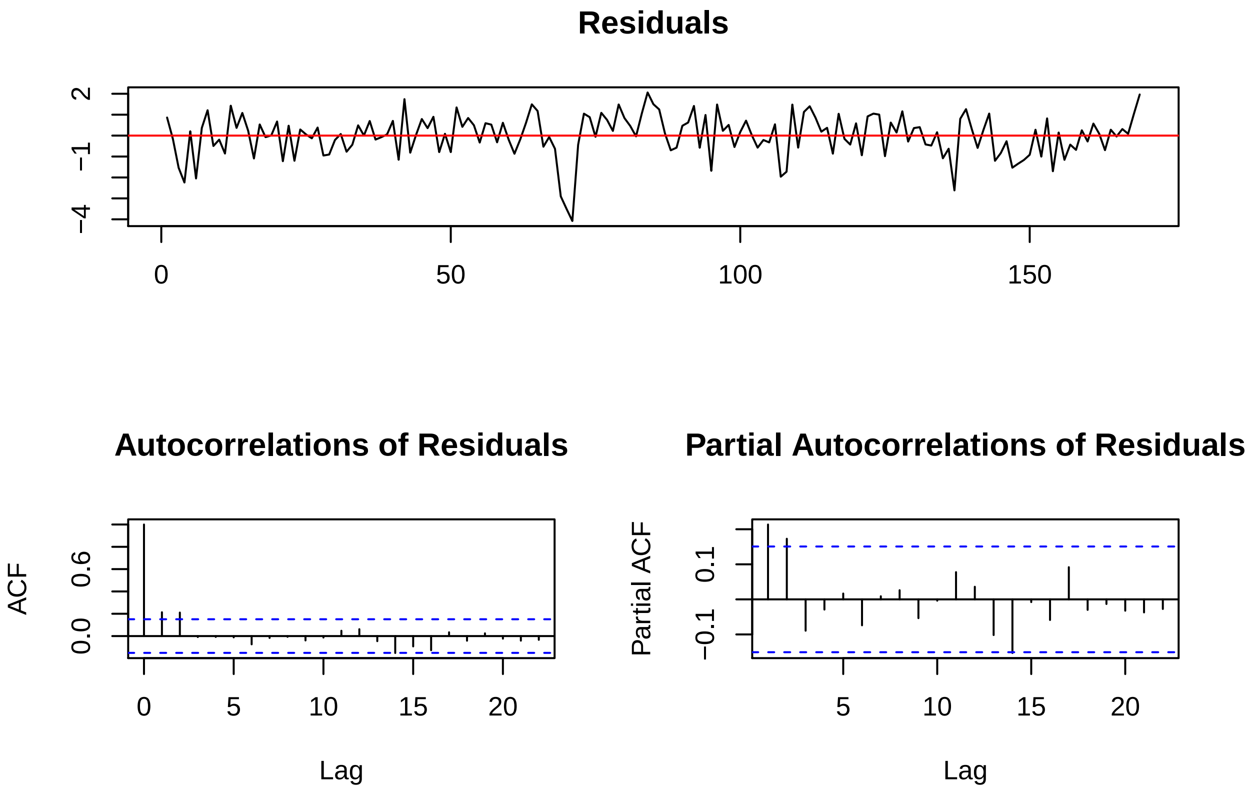


Le test de Zivot et Andrews nous montre qu'il ne s'agit donc pas une *quebra estrutural* dans notre serie considérée. C'est juste un choc aléatoire endogène. Ce test montre que une seule *quebra estrutural* mais il y a encore d'autre.

Test de Dickey-Fuller (DF and ADF)

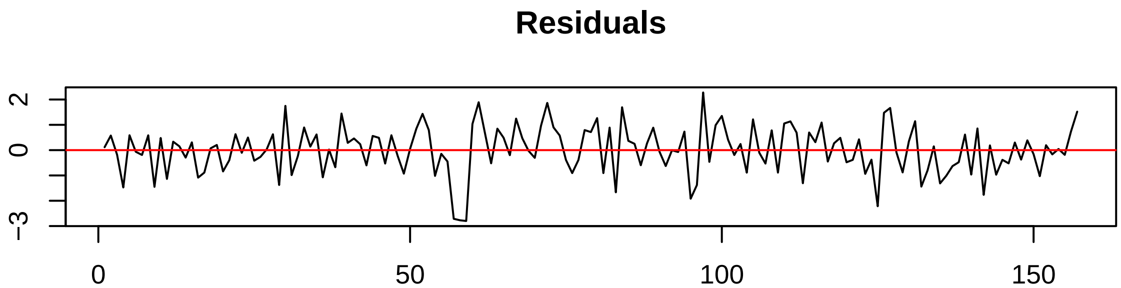
1. Modelo com intercepto e tendência (DF)

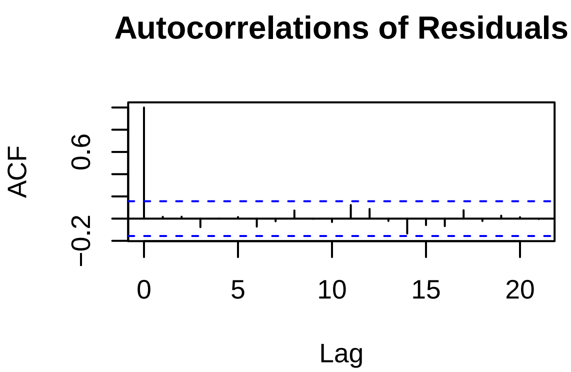
```r
df.ibc = ur.df(ibc, type = "trend", lags = 0)
plot(df.ibc)
```

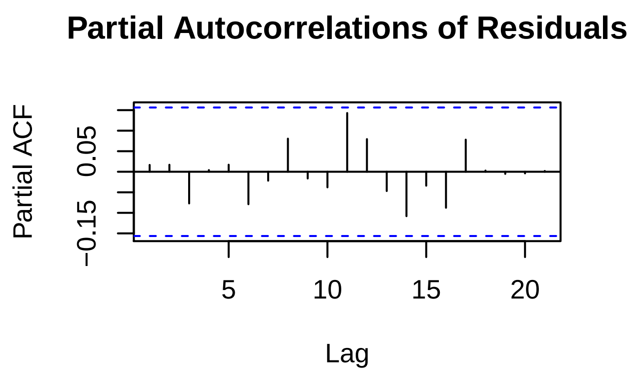


La fonction de l'autocorrélation partialle nous indique d'inclure deux retard, c'est à dire ARIMA (2, 0, 2). Bien que l'analyse graphique reste informelle mais elle nous dit quelque chose quand à la décision à prendre. Voyons avec l'analyse ADF:

```r
adf.ibc = ur.df(ibc, type = "trend", lags = 12, selectlags = "BIC")
plot(adf.ibc)
```








```r
summary(adf.ibc)
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression trend
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.79985 -0.52995  0.04467  0.61924  2.27816
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.688e+00  1.245e+00   1.355   0.1773
## z.lag.1     -1.199e-02  1.152e-02  -1.041   0.2994
## tt           1.826e-06  3.457e-03   0.001   0.9996
## z.diff.lag1  2.039e-01  8.010e-02   2.546   0.0119 *
## z.diff.lag2  2.086e-01  8.106e-02   2.573   0.0110 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9444 on 152 degrees of freedom
```

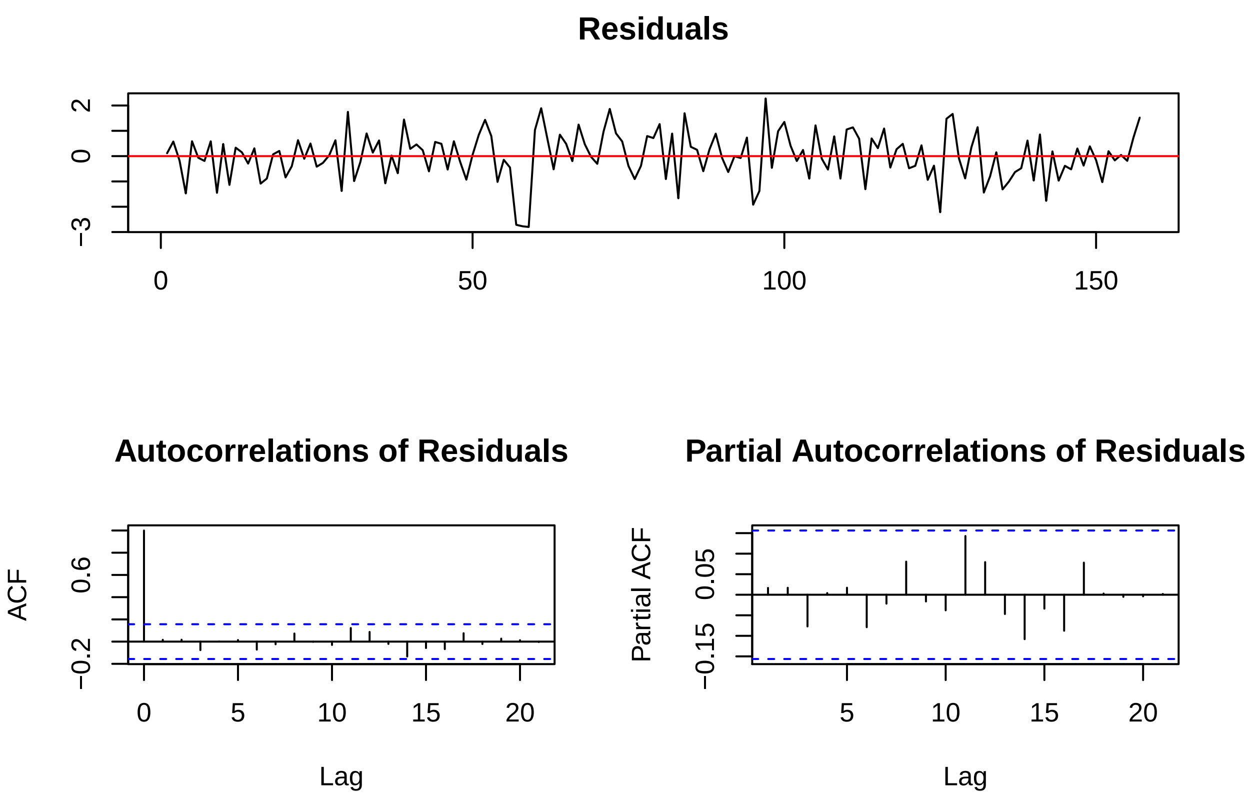
```
## Multiple R-squared:  0.1516, Adjusted R-squared:  0.1293
## F-statistic: 6.789 on 4 and 152 DF,  p-value: 4.699e-05
##
##
## Value of test-statistic is: -1.0412 2.0664 2.0424
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau3 -3.99 -3.43 -3.13
## phi2  6.22  4.75  4.07
## phi3  8.43  6.49  5.47
```

Contrairement au test DF, le test ADF nous indique qu'il faut inclure 0 retard pour MA et 1 retard pour AR, c'est à dire AR(1).

Tau3 = -3.43 (critical values for test statistics) est lié à la valeur -1.0412 (value of test-statistics) qui à l'interieur de l'intervalle (région critique) non critique. Alors dans ce cas on ne rejet pas l'hypothèse nulle au seuil de 5%.

2. Modelo com intercepto e sem tendência (Test ADF)

```r
adf.ibc2 = ur.df(ibc, type = "drift", lags = 12, selectlags = "BIC")
plot(adf.ibc2)
```




```r
summary(adf.ibc2)
```

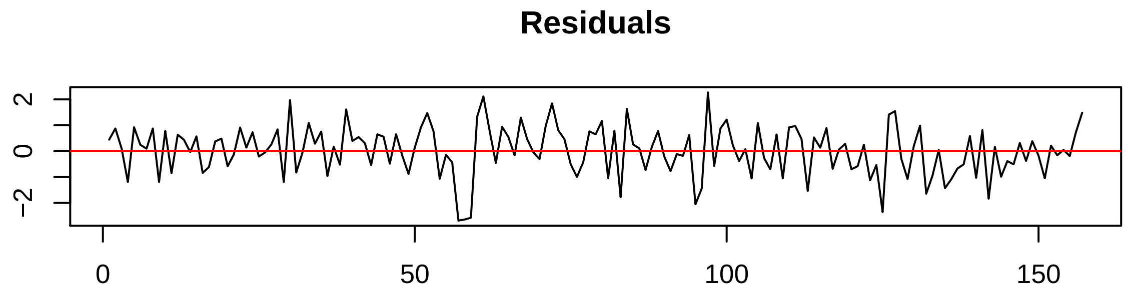
```
##
```

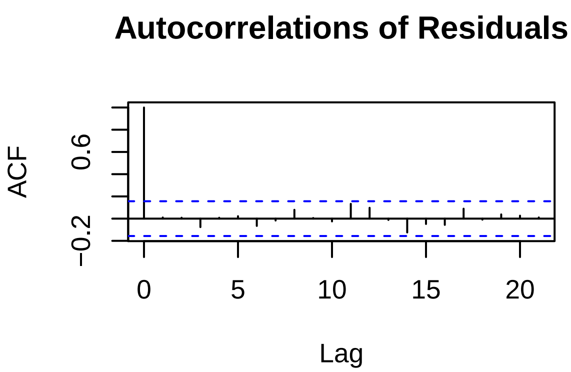
```
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression drift
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + z.diff.lag)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.79992 -0.52997  0.04479  0.61924  2.27811
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.68747    0.78035   2.162  0.03214 *
## z.lag.1     -0.01198    0.00591  -2.028  0.04432 *
## z.diff.lag1  0.20393    0.07798   2.615  0.00981 **
## z.diff.lag2  0.20856    0.07767   2.685  0.00805 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9413 on 153 degrees of freedom
## Multiple R-squared:  0.1516, Adjusted R-squared:  0.1349
## F-statistic: 9.112 on 3 and 153 DF,  p-value: 1.382e-05
##
##
## Value of test-statistic is: -2.0277 3.12
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau2 -3.46 -2.88 -2.57
## phi1  6.52  4.63  3.81
```

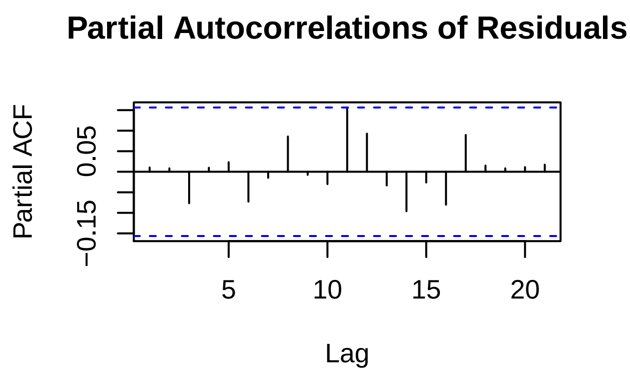
-2.88 n'appartient pas à la région critique, alors on ne rejet pas l'hypothèse nulle.

3. Modelo sem intercepto sem tendência

```r
adf.ibc3 = ur.df(ibc, type = "none", lags = 12, selectlags = "BIC")
plot(adf.ibc3)
```








```r
summary(adf.ibc3)
```

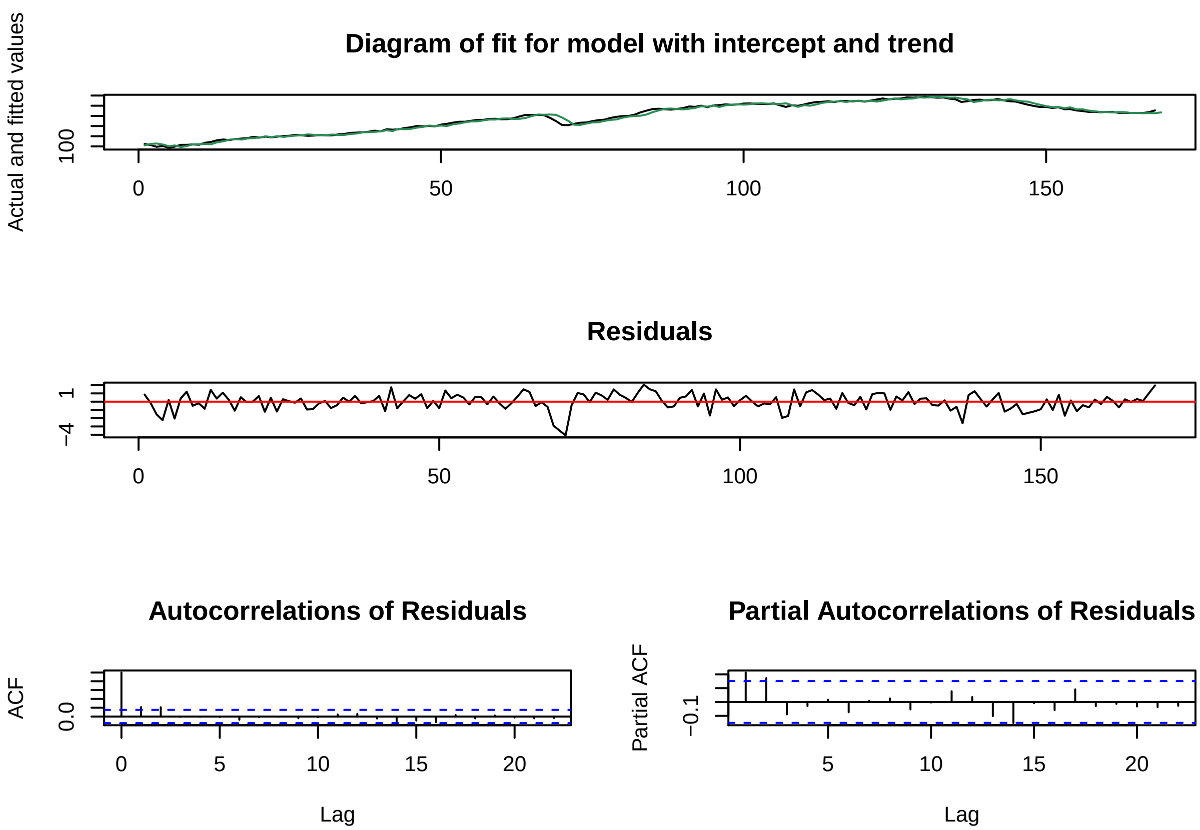
```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression none
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.6848 -0.5803  0.1087  0.7306  2.2741
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## z.lag.1     0.0007331  0.0005932   1.236  0.21838
## z.diff.lag1 0.2287486  0.0780450   2.931  0.00389 **
## z.diff.lag2 0.2269508  0.0781214   2.905  0.00421 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9524 on 154 degrees of freedom
## Multiple R-squared:  0.1596, Adjusted R-squared:  0.1433
## F-statistic: 9.751 on 3 and 154 DF,  p-value: 6.272e-06
```

```
##
##
## Value of test-statistic is: 1.2359
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau1 -2.58 -1.95 -1.62
```

1.23 n'appartient pas à la région cretique d'où on rejet l'hypothèse nulle. Et on conlut que la serie IBC-Brazil n'est pas stationnaire.

Test de Philip-Perron

```r
pp.ibc = ur.pp(ibc, type = "Z-tau", model = "trend", lags = "short")
plot(pp.ibc)
```




```r
summary(pp.ibc)
```

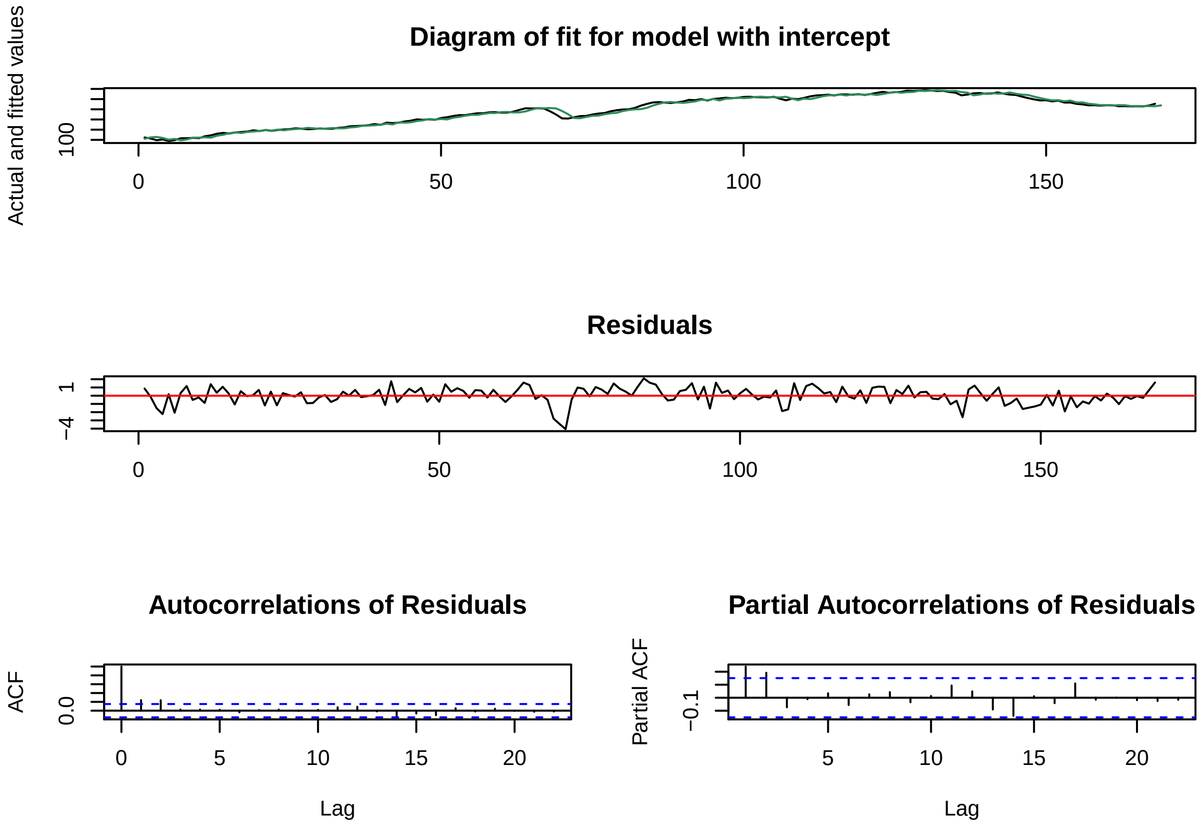
```
##
## ##################################
## # Phillips-Perron Unit Root Test #
## ##################################
##
## Test regression with intercept and trend
```

```
##
##
## Call:
## lm(formula = y ~ y.l1 + trend)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.0806 -0.5761  0.0895  0.6748  2.0617
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.096595   1.451231  -0.067    0.947
## y.l1         1.002380   0.011275  88.902   <2e-16 ***
## trend       -0.005184   0.003384  -1.532    0.128
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.004 on 166 degrees of freedom
## Multiple R-squared:  0.9953, Adjusted R-squared:  0.9952
## F-statistic: 1.751e+04 on 2 and 166 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic, type: Z-tau  is: -0.2652
##
##            aux. Z statistics
## Z-tau-mu             -0.4851
## Z-tau-beta           -0.7973
##
## Critical values for Z statistics:
##                     1pct      5pct     10pct
## critical values -4.014887 -3.437124 -3.142473
```

La formule utilisée ici est la suivante :

$$\Delta Y_t = \mu + \beta t + \tau Y_{t-1} + \varepsilon_t$$

```r
pp.ibc1 = ur.pp(ibc, type = "Z-tau", model = "constant", lags = "short")
plot(pp.ibc1)
```




```r
summary(pp.ibc1)
```

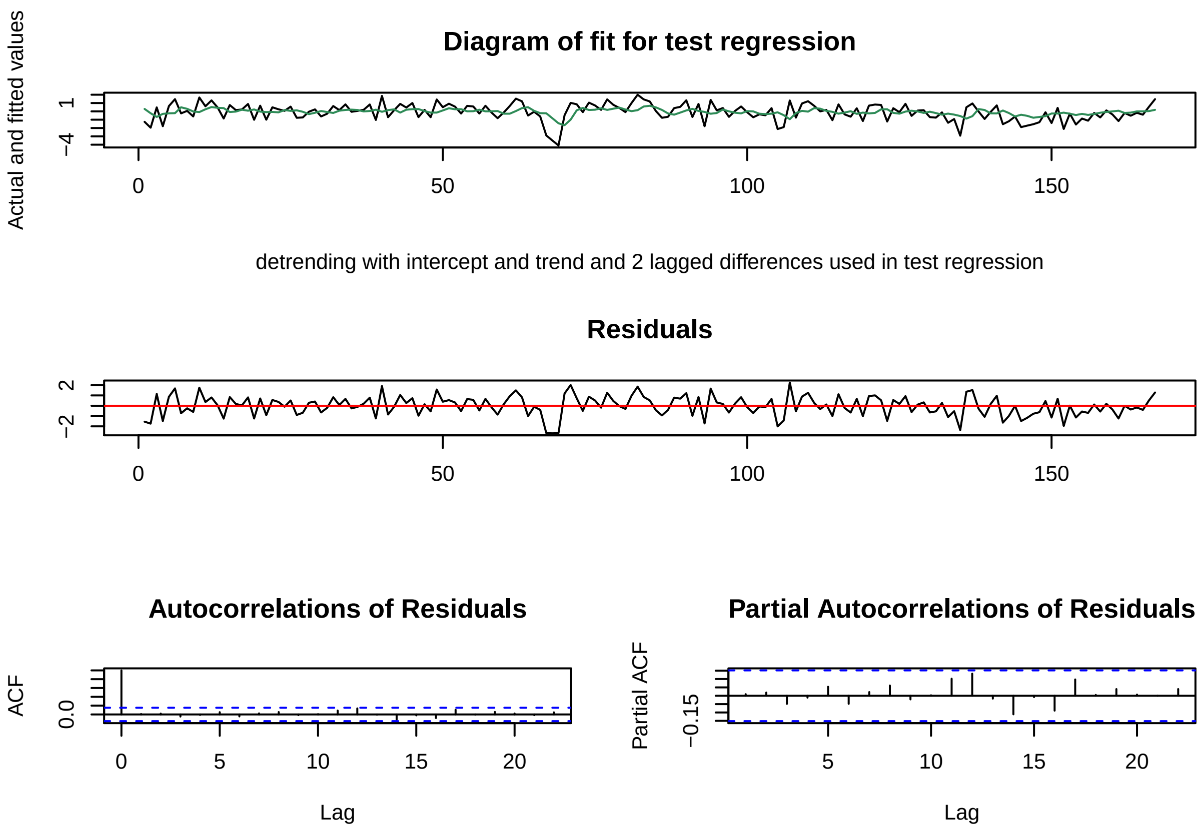
```
##
## ###################################
## # Phillips-Perron Unit Root Test #
## ###################################
##
## Test regression with intercept
##
##
## Call:
## lm(formula = y ~ y.l1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.0657 -0.5257  0.0730  0.6995  2.1062
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.864593   0.685798   2.719  0.00724 **
## y.l1        0.987116   0.005296 186.386  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.008 on 167 degrees of freedom
## Multiple R-squared:  0.9952, Adjusted R-squared:  0.9952
## F-statistic: 3.474e+04 on 1 and 167 DF,  p-value: < 2.2e-16
```

```
##
##
## Value of test-statistic, type: Z-tau  is: -2.1069
##
##          aux. Z statistics
## Z-tau-mu            2.3248
##
## Critical values for Z statistics:
##                      1pct      5pct     10pct
## critical values -3.470021 -2.878594 -2.575766
```

Test de DF-GLS (ERS-Elliot, Rotenberg e)

1. Modelo avec tendance

```r
ers.ibc = ur.ers(ibc, type = "DF-GLS", model = "trend", lag.max = 2)
plot(ers.ibc)
```




```r
summary(ers.ibc)
```

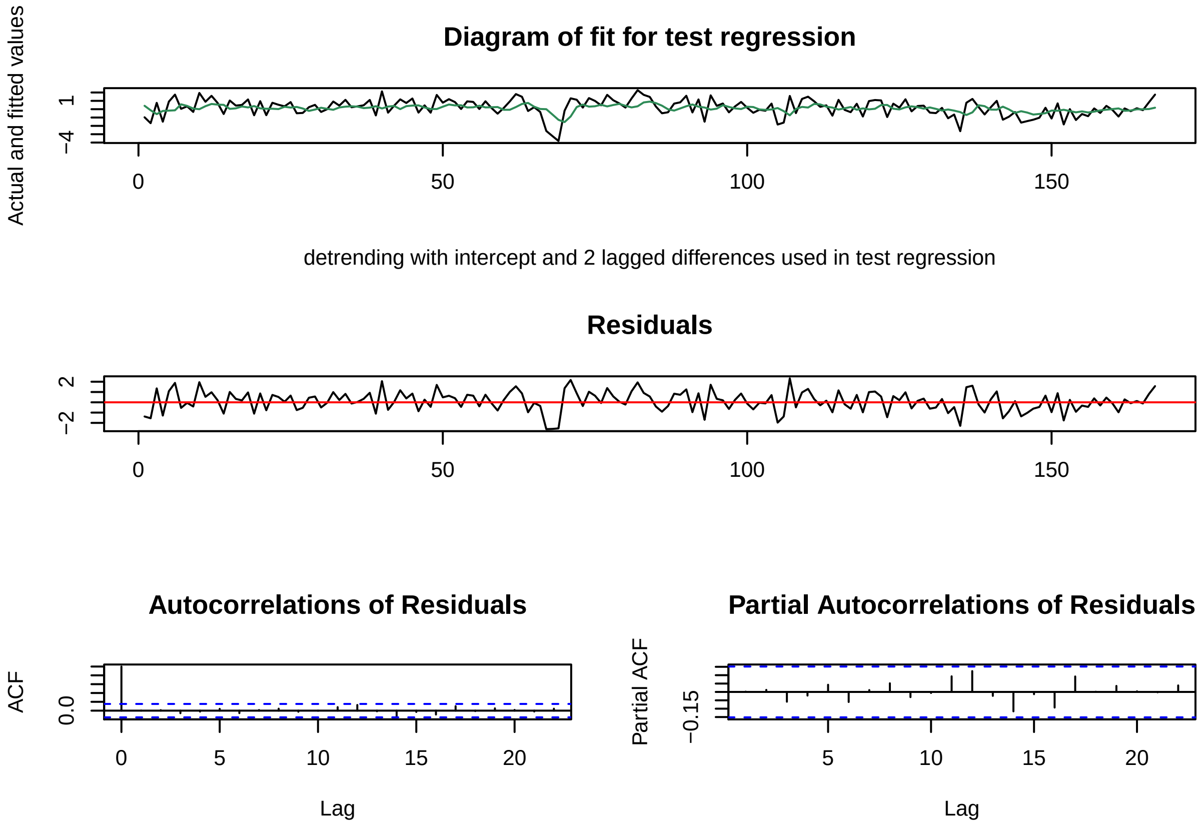
```
##
## ###############################################
## # Elliot, Rothenberg and Stock Unit Root Test #
## ###############################################
##
```

```
## Test of type DF-GLS
## detrending of series with intercept and trend
##
##
## Call:
## lm(formula = dfgls.form, data = data.dfgls)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.67313 -0.65150  0.04426  0.68331  2.25021
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## yd.lag       -0.009623   0.010406  -0.925  0.35643
## yd.diff.lag1  0.218350   0.077138   2.831  0.00523 **
## yd.diff.lag2  0.221198   0.077526   2.853  0.00489 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9732 on 164 degrees of freedom
## Multiple R-squared:  0.114,  Adjusted R-squared:  0.09775
## F-statistic: 7.031 on 3 and 164 DF,  p-value: 0.0001779
##
##
## Value of test-statistic is: -0.9248
##
## Critical values of DF-GLS are:
##                  1pct  5pct 10pct
## critical values -3.46 -2.93 -2.64
```

Ce test applique aussi la correction des données. Non rejeter l'hypothèse nulle, il y a aussi présence de racine unitaire.

2. Modelo avec constante

```
ers.ibc1 = ur.ers(ibc, type = "DF-GLS", model = "constant", lag.max = 2)
plot(ers.ibc1)
```




```
summary(ers.ibc1)
```

```
##
## ###############################################
## # Elliot, Rothenberg and Stock Unit Root Test #
## ###############################################
##
## Test of type DF-GLS
## detrending of series with intercept
##
##
## Call:
## lm(formula = dfgls.form, data = data.dfgls)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.6089 -0.5188  0.1970  0.8272  2.3281
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## yd.lag       0.0008308  0.0029882   0.278  0.78133
## yd.diff.lag1 0.2216597  0.0768289   2.885  0.00444 **
## yd.diff.lag2 0.2211461  0.0765616   2.888  0.00439 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9814 on 164 degrees of freedom
```

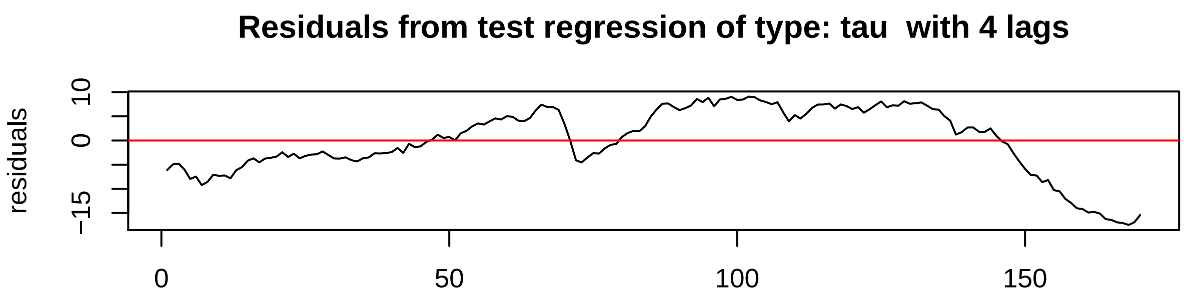
```
## Multiple R-squared:  0.1261, Adjusted R-squared:  0.1101
## F-statistic: 7.885 on 3 and 164 DF,  p-value: 6.033e-05
##
##
## Value of test-statistic is: 0.278
##
## Critical values of DF-GLS are:
##                  1pct  5pct 10pct
## critical values -2.58 -1.94 -1.62
```

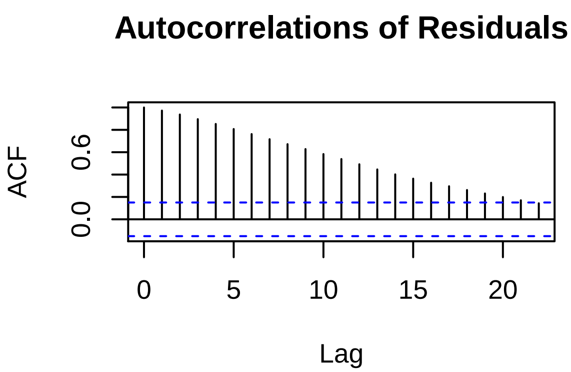
Il ya présence de RU.

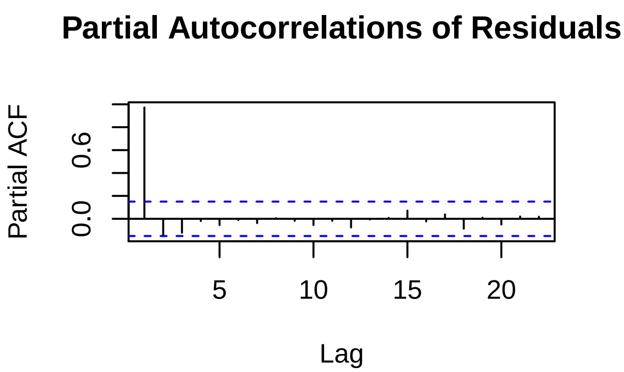
Test de KPSS

1. Modèle avec tendance

```
kpss.ibc = ur.kpss(ibc, type = "tau", lags = "short")
plot(kpss.ibc)
```








```
summary(kpss.ibc)
```

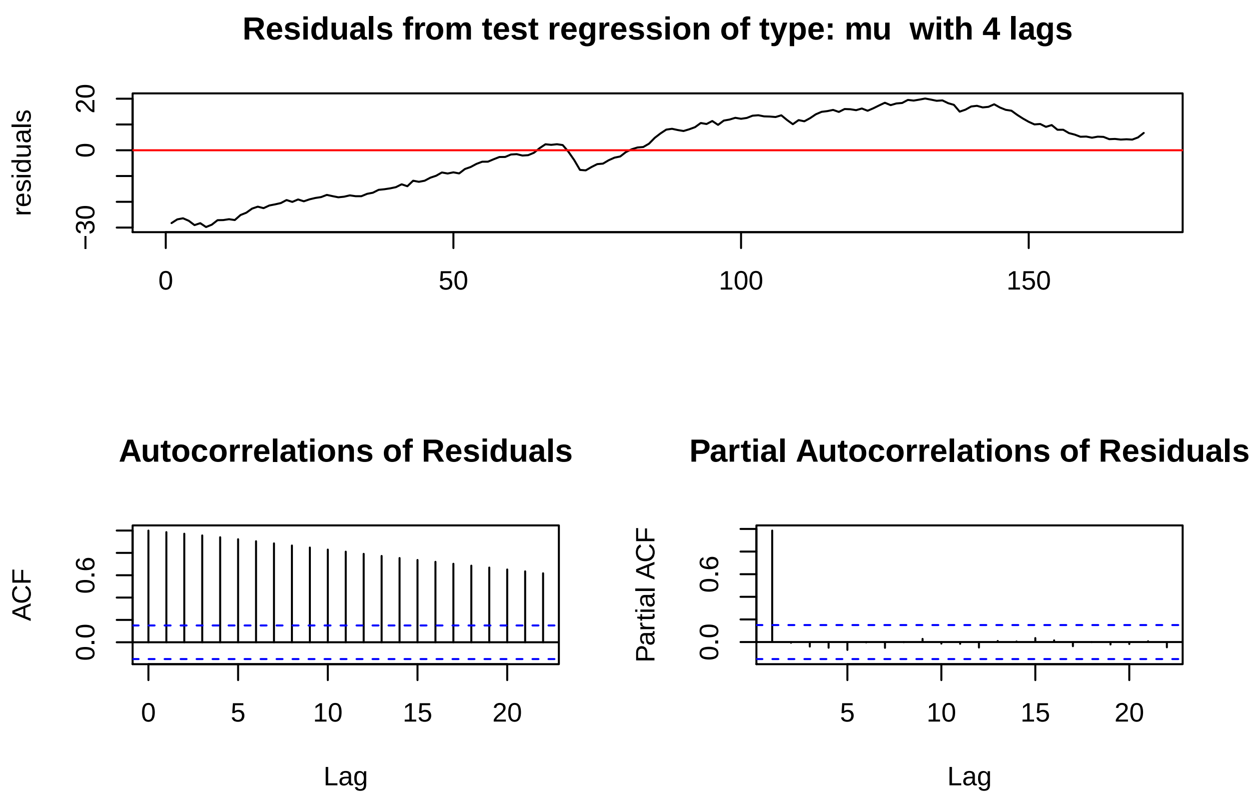
```
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
```

```
## Test is of type: tau with 4 lags.
##
## Value of test-statistic is: 0.6313
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.119 0.146  0.176 0.216
```

Les résidus ne sont pas aléatoire, le graphique de correlation non plus. Alors on rejet l'hypothèse nulle. La série n'est pas stationnaire.

2. Modèle avec constante

```r
kpss.ibc1 = ur.kpss(ibc, type = "mu", lags = "short")
plot(kpss.ibc1)
```




```r
summary(kpss.ibc1)
```

```
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 4 lags.
##
## Value of test-statistic is: 2.9676
##
```

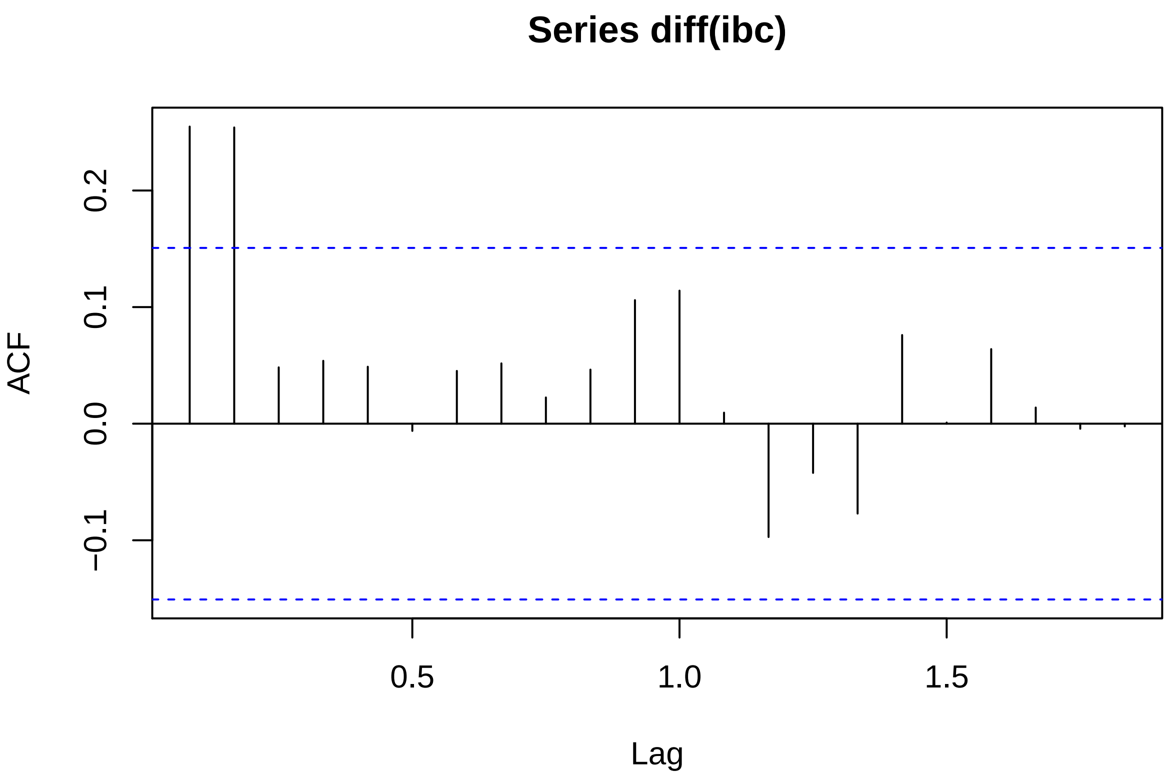
```
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

La serie est non stationnaire. Donc, On conclut que IBC-Brazil n'est pas stationnaire.

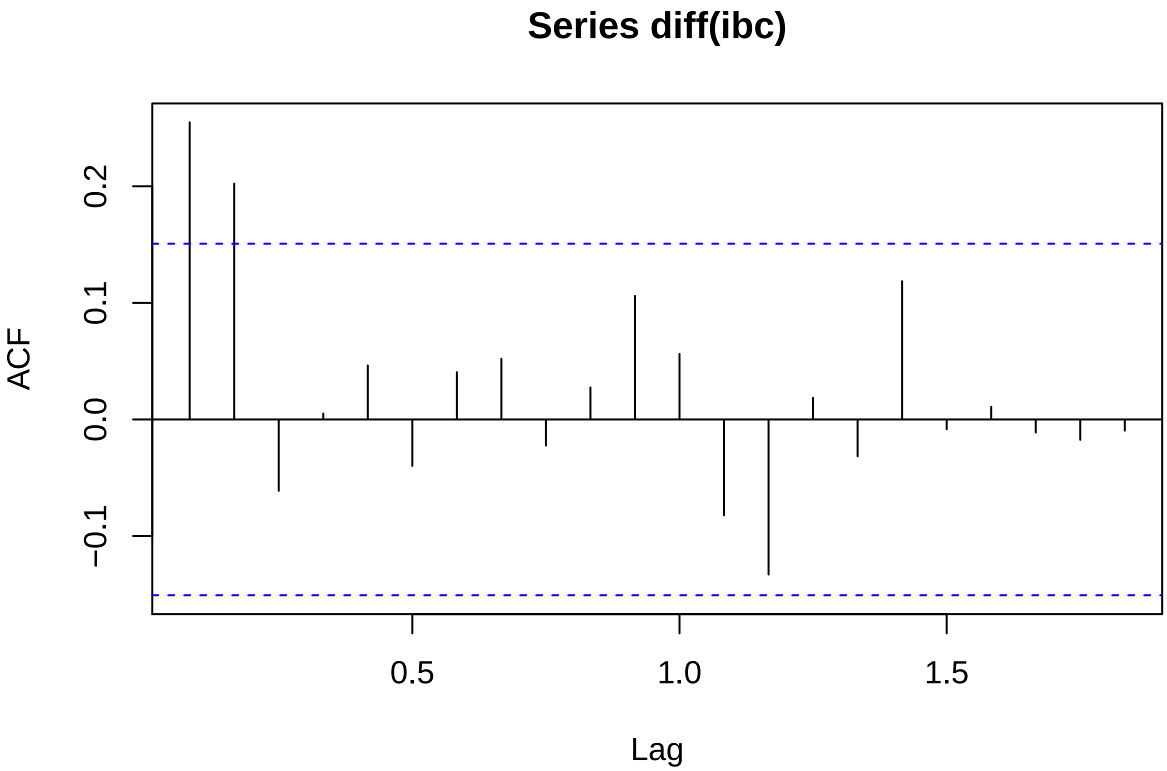
Ajuste ARIMA

Dado que IBC é não estacionário, devemos usar primeira diferença. Où seja :

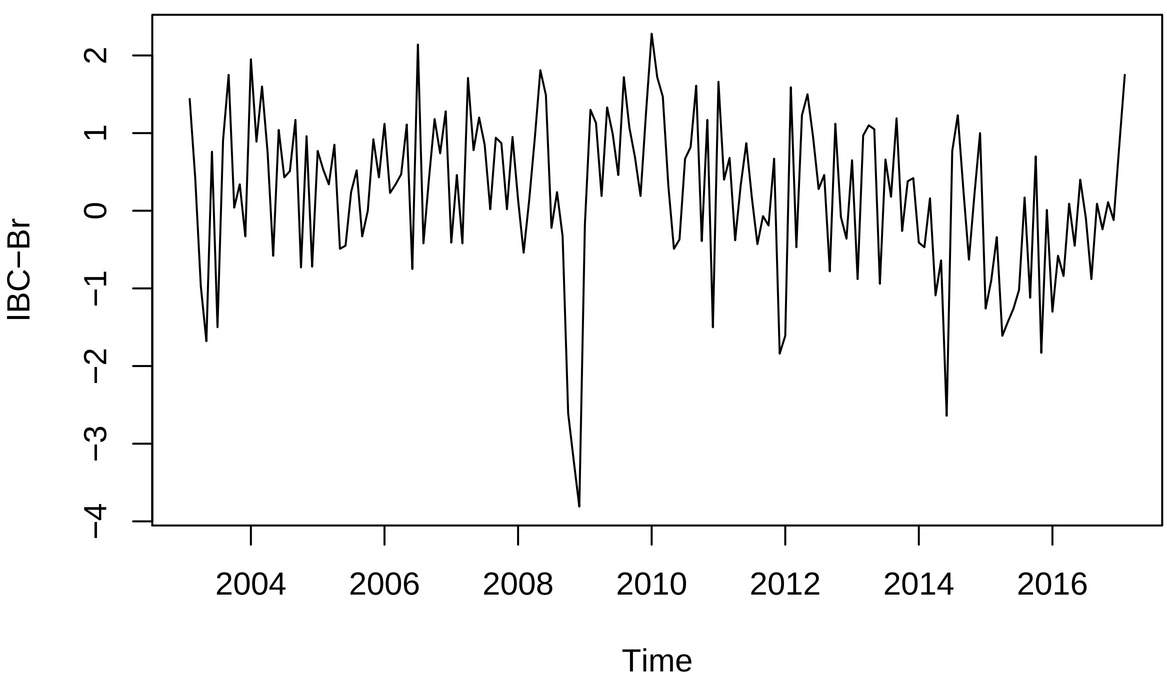
```r
layout(1:1)
acf(diff(ibc), drop.lag.0 = TRUE, type = "correlation")
```




```r
acf(diff(ibc), type = "partial")
```




```r
plot(diff(ibc))
```



À la foçon dont les deux graphique se présente, il est difficile de savoir si quel type de modèle ajusté. Ainsi, nous décidons de faire plusieurs modèles, puis nous allons décidé ce qu'il faut considéré.

```r
model1 = Arima(ibc, order = c(2, 1, 0))
model2 = Arima(ibc, order = c(0, 1, 2))
```

```r
model3 = Arima(ibc, order = c(2, 1, 2))
model4 = Arima(ibc, order = c(1, 1, 1))
```

Vérification de coefficients des modèles :

```r
coeftest(model1)
```

```
##
## z test of coefficients:
##
##     Estimate Std. Error z value Pr(>|z|)
## ar1 0.225136   0.075525  2.9809 0.002874 **
## ar2 0.222606   0.075579  2.9453 0.003226 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
coeftest(model2)
```

```
##
## z test of coefficients:
##
##     Estimate Std. Error z value  Pr(>|z|)
## ma1 0.242699   0.076486  3.1731 0.0015082 **
## ma2 0.273400   0.071910  3.8020 0.0001436 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
coeftest(model3)
```

```
##
## z test of coefficients:
##
##     Estimate Std. Error z value Pr(>|z|)
## ar1 0.046811   0.296662  0.1578   0.8746
## ar2 0.205888   0.256967  0.8012   0.4230
## ma1 0.193481   0.301166  0.6424   0.5206
## ma2 0.076802   0.248710  0.3088   0.7575
```

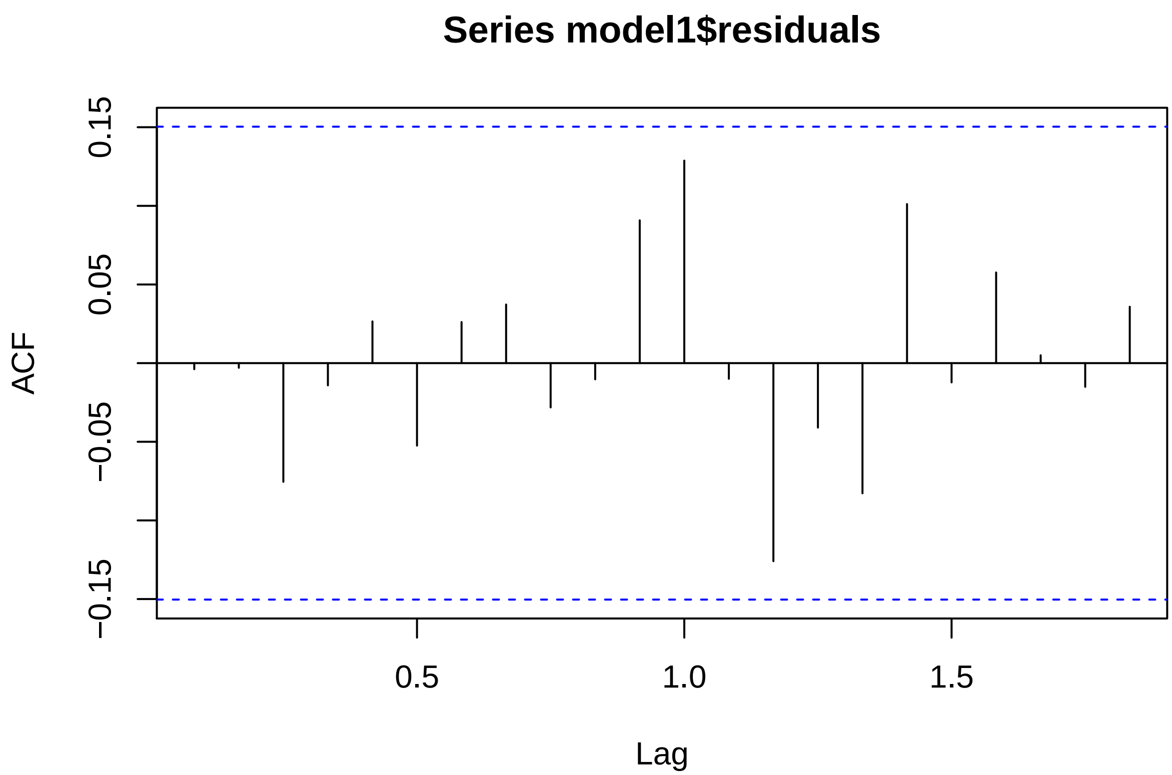
```r
coeftest(model4)
```

```
##
## z test of coefficients:
##
##     Estimate Std. Error z value  Pr(>|z|)
## ar1  0.71425    0.14092  5.0686 4.008e-07 ***
## ma1 -0.46409    0.17877 -2.5960   0.00943 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Modèle 1 les coefficients sont statistiquement différents de zéro. De même pour le modèle 2. Mais le modèle 3 non. Et le modèle 4 les coefficients sont aussi significatifs.

Pour faire le choix d'un modèle on le fait avec le critère AIC E BIC le plus petit. Mais aussi on peut voir l'ecart type de chaque modèle. les classés dans un tableau si possible. Ou on peut voir aussi les résidus de chaque modèles :

```
acf(model1$residuals, drop.lag.0 = TRUE)
```




```
Box.test(model1$residuals, lag = 14, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  model1$residuals
## X-squared = 9.7627, df = 14, p-value = 0.7793
```

P-value = 0.77 est supérieur au seuil de 5%, on rejet l'hypothèse nulle.

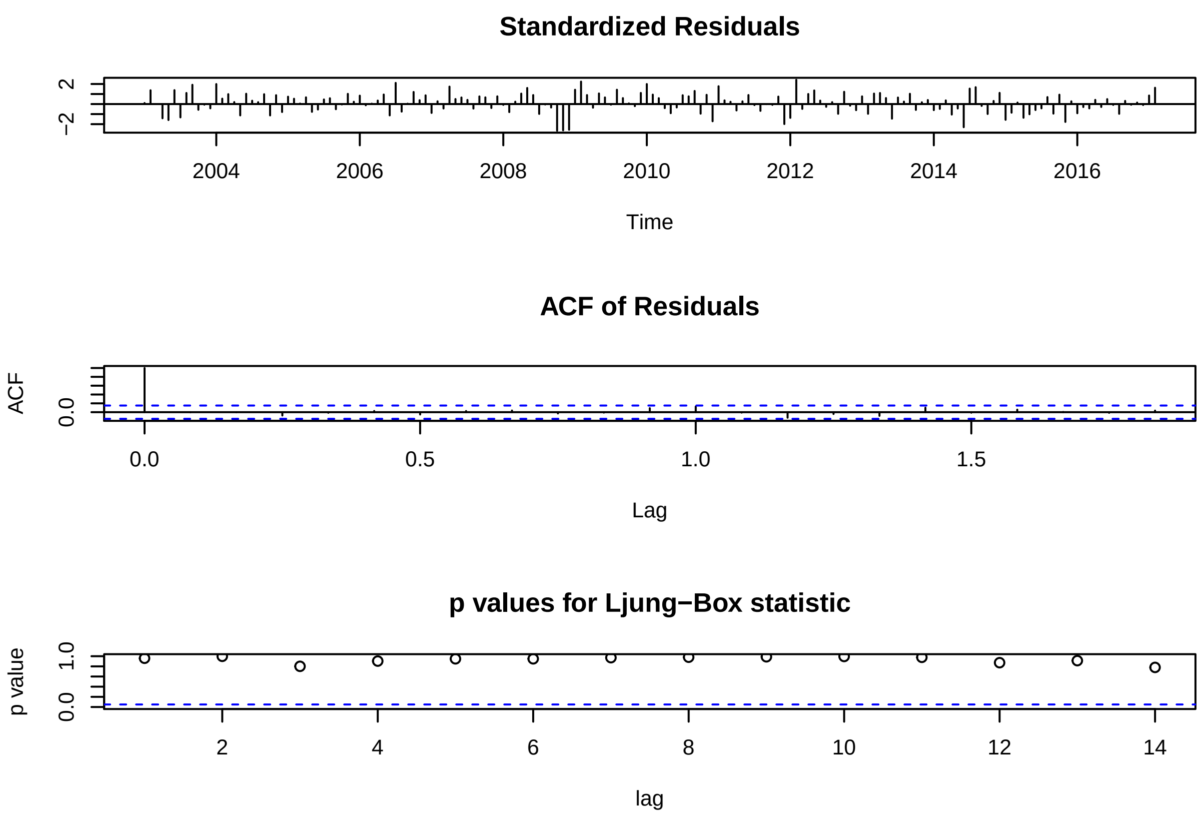
```
for (i in 1:14) {
  b = Box.test(model1$residuals, i, type = "Ljung-Box")$p.value
  print(b)
  }
```

```
## [1] 0.9594742
## [1] 0.997932
## [1] 0.8008211
## [1] 0.9041306
## [1] 0.9484891
## [1] 0.948724
```

```
## [1] 0.971239
## [1] 0.9802081
## [1] 0.9884856
## [1] 0.9946727
## [1] 0.9777693
## [1] 0.8722864
## [1] 0.912603
## [1] 0.7793202
```

Toutes les valeurs de p-value nous dit de rejeter l'hypothèse nulle. On peut aussi voir le graphique de Ljung-Box pour le p-value.

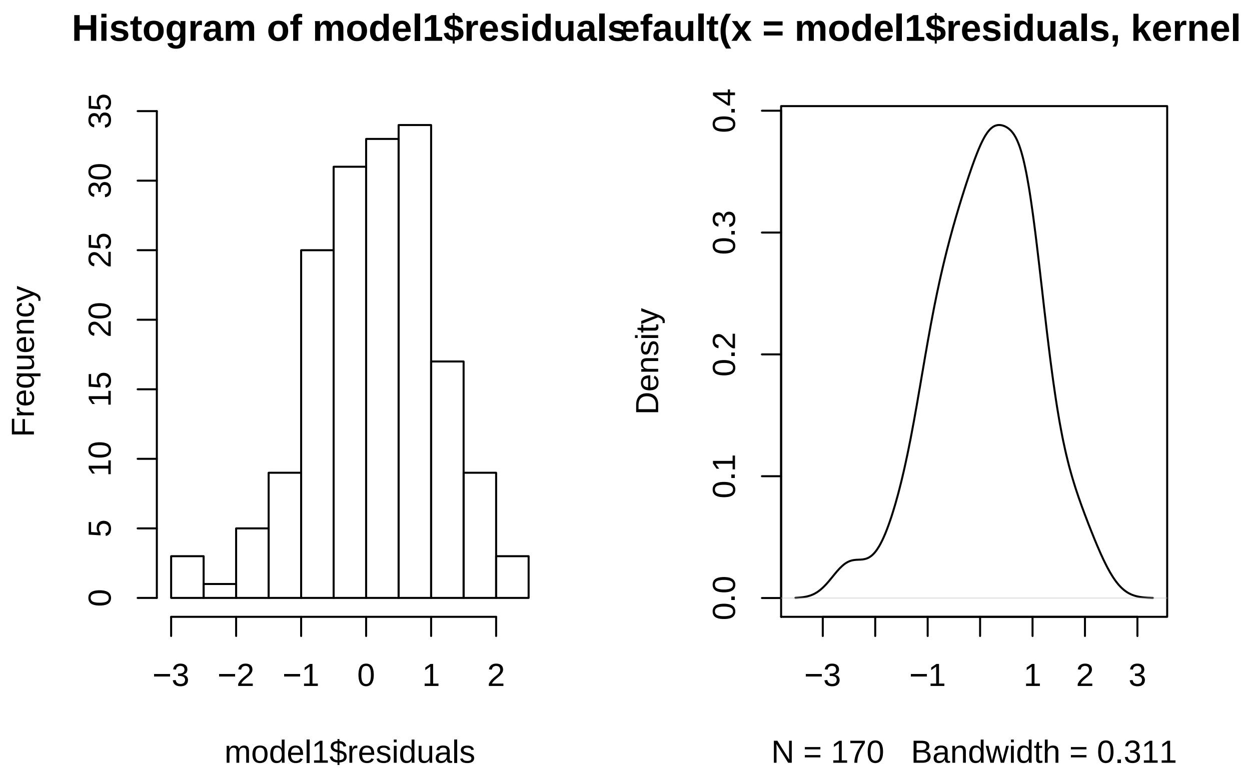
```r
tsdiag(model1, gof.lag = 14)
```



Les p-values tout au long du graphique sont très hautes, on rejet H0.

Test de nomalité

```r
par(mfrow = c(1:2))
hist(model1$residuals)
plot(density(model1$residuals, kernel = "gaussian"))
```




```r
shapiro.test(model1$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  model1$residuals
## W = 0.98963, p-value = 0.2504
```

```r
jarque.bera.test(model1$residuals)
```

```
##
##  Jarque Bera Test
##
## data:  model1$residuals
## X-squared = 2.8529, df = 2, p-value = 0.2402
```

Les tests de Shapiro et Jarque-Bera nous montre qu'il faut rejeter l'hypothèse nulle.

Test de heterocedasticidade

```r
ArchTest(model1$residuals, lags = 4)
```

```
##
##  ARCH LM-test; Null hypothesis: no ARCH effects
##
## data:  model1$residuals
## Chi-squared = 21.505, df = 4, p-value = 0.0002514
```

```r
ArchTest(model1$residuals, lags = 12)
```

```
##
##  ARCH LM-test; Null hypothesis: no ARCH effects
##
## data:  model1$residuals
## Chi-squared = 22.565, df = 12, p-value = 0.03166
```

Notre serie a un problème de l'hétérocédasticité.

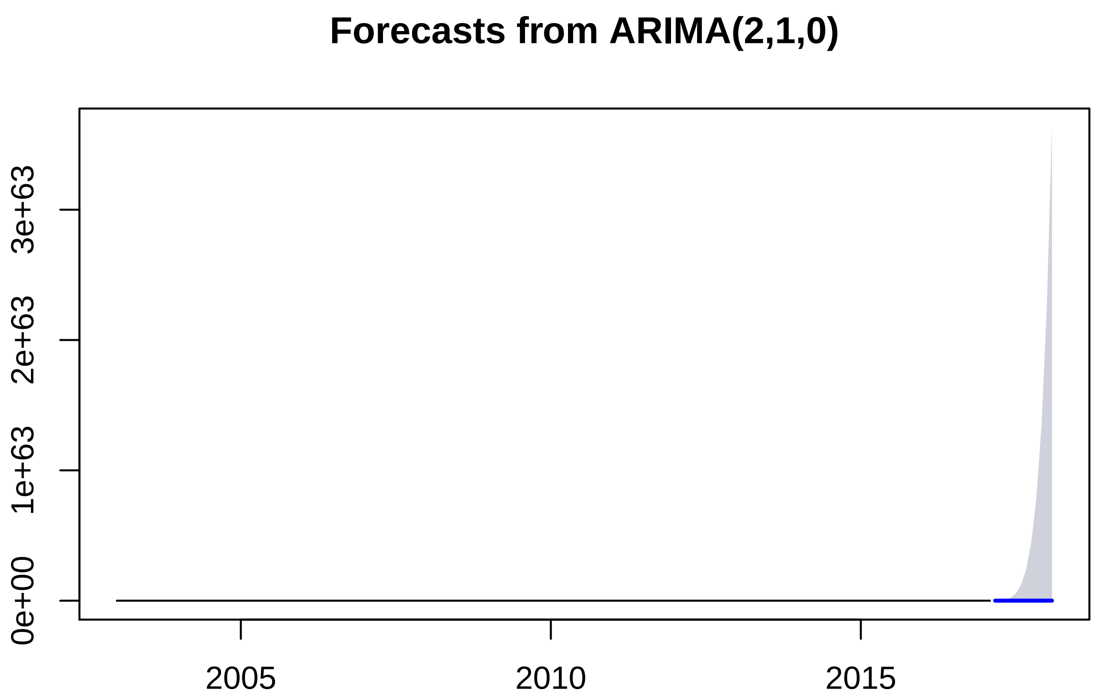
Prvision

```r
layout(1:1)
prev = forecast(model1, h = 12, level = c(0.90), lambda = 0, biasadj = FALSE)
print(prev)
```

```
##          Point Forecast        Lo 90        Hi 90
## Mar 2017   1.157462e+59 2.315313e+58 5.786340e+59
## Apr 2017   1.946607e+59 1.527613e+58 2.480523e+60
## May 2017   2.489196e+59 7.472235e+57 8.292157e+60
## Jun 2017   2.953638e+59 3.776231e+57 2.310234e+61
## Jul 2017   3.242309e+59 1.892780e+57 5.554037e+61
## Aug 2017   3.439625e+59 9.842380e+56 1.202048e+62
## Sep 2017   3.558789e+59 5.287043e+56 2.395474e+62
## Oct 2017   3.633653e+59 2.945193e+56 4.483047e+62
## Nov 2017   3.678507e+59 1.695671e+56 7.979977e+62
## Dec 2017   3.705816e+59 1.006476e+56 1.364470e+63
## Jan 2018   3.722143e+59 6.138408e+55 2.256994e+63
## Feb 2018   3.731969e+59 3.835535e+55 3.631199e+63
```

Graphique de prevision

```r
plot(prev)
```



Avaliar as previsões

```
erro1 = matrix(NA, nrow = length(window(ibc, start = 2016)), ncol = 1)
errop = matrix(NA, nrow = length(window(ibc, start = 2016)), ncol = 1)

real = matrix(NA, nrow = length(window(ibc, start = 2016)), ncol = 1)
previsto = matrix(NA, nrow = length(window(ibc, start = 2016)), ncol = 1)

y = BoxCox(ibc, lambda = 0)
```