Indice d’Activité Économique (IBC)-Brazil

Henri Makika

May 8, 2019

Introduction

L’IBC-Brazil est utilisé comme un paramètre pour évaluer le taux de croissance de l’économie brésilienne au fil des mois. L’indicateur exerce également une grande influence sur les estimations des marchés financiers concernant le produit intérieur brut (PIB) et le taux de Selic (taux directeur de la Banque Centrale du Brésil).

L’indice d’activité économique de la Banque Centrale du Brésil (IBC-Br) est un indicateur créé pour tenter d’anticiper les résultats du produit intérieur brut (PIB) du pays, constituant un paramètre préliminaire de l’évolution de l’activité économique brésilienne. Le calcul de l’IBC-Br aide également l’autorité monétaire à définir l’objectif du taux d’intérêt de base de l’économie, le taux de Selic.

L’objectif de ce papier est de montrer comment faire une prévision selon la méthodologie proposée par Box & Jenkins (1970).

La méthodologie Box & Jenkins pour des séries temporelles

La construction du modèle de séries temporelles comprend :

1. Identification: basée sur l’analyse d’autocorrélation, critères d’autocorrélation partielle et / ou d’information;
2. Estimation: les paramètres de modèle identifiés sont estimés;
3. Vérification du modèle ajusté: à travers une analyse de résidus, il est vérifié s’il convient aux fins visées, dans le cas, pour les prévisions.

Nous importons ici les packages nécessaires pour l’analyse de nos données :

```
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)
```

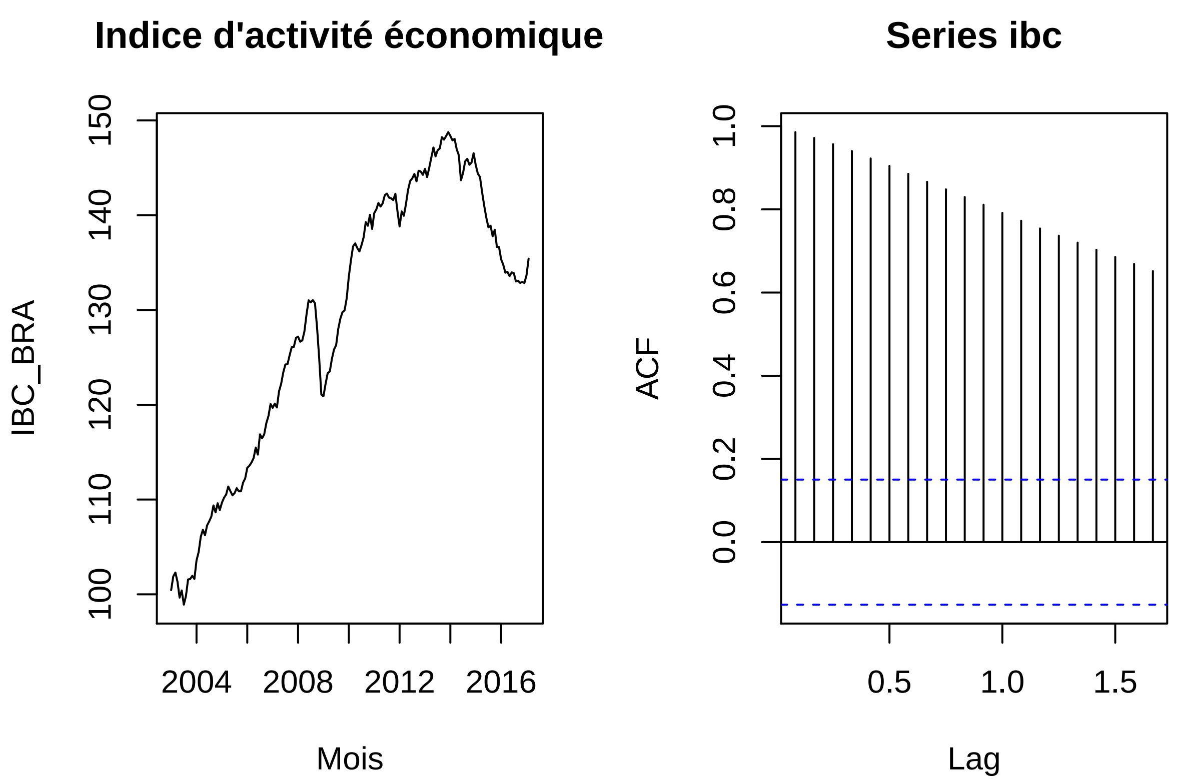
La lecture des données

```
IBC_BRA <- read_excel("~/Videos/Unicamp_IE 2019/HO:236A Times Series/Aula 8/IBC_BR.xlsx")
ibc = ts(IBC_BRA[,2], start = c(2003, 1), frequency = 12)
```

Identification du modèle d'analyse

Méthode graphique

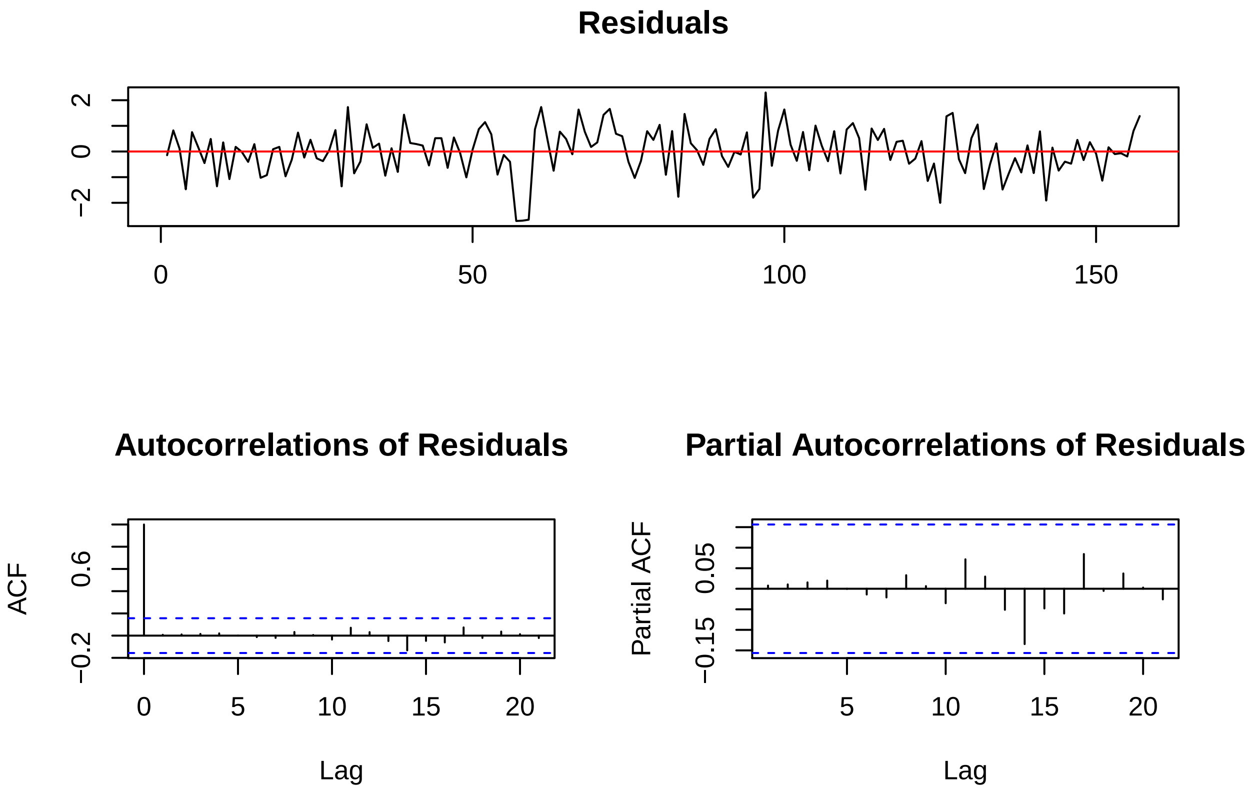
```r
par(mfrow = c(1, 2))
plot(ibc, main = "Indice d'activité économique", ylab = "IBC_BRA",
     xlab = "Mois")
acf(ibc, lag.max = 20, drop.lag.0 = TRUE)
```



Test de Dickey Fuller Augmented (ADF)

Le test veut savoir si la série est stationnaire ou pas.

```r
df1 <- ur.df(ibc, type = "trend", lags = 12)
plot(df1)
```



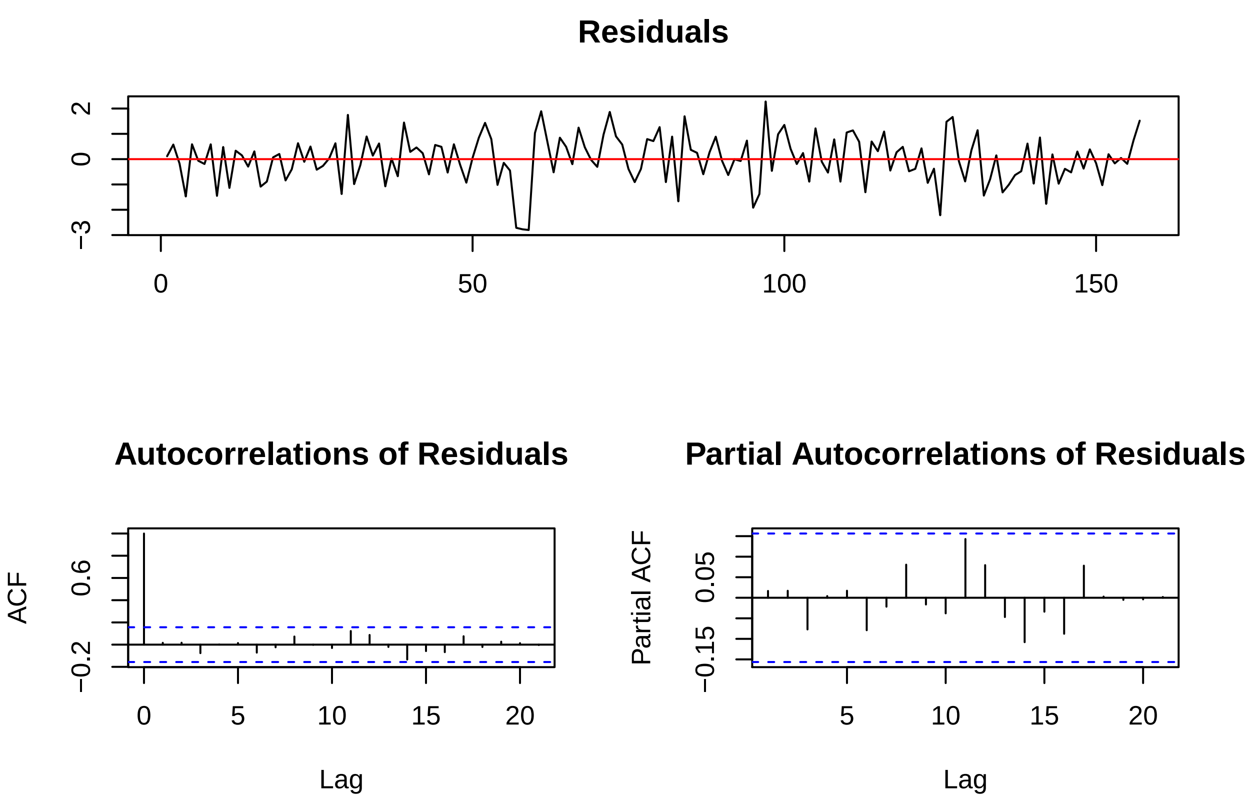

```r
summary(df1)
```

```
##
## ###############################################
## # 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
## -2.7086 -0.5231  0.1095  0.6703  2.3009
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.259859   1.572352   1.437  0.15285
## z.lag.1      -0.018371   0.015706  -1.170  0.24409
## tt            0.002479   0.005460   0.454  0.65058
## z.diff.lag1   0.222934   0.084686   2.632  0.00941 **
## z.diff.lag2   0.248075   0.087104   2.848  0.00505 **
## z.diff.lag3  -0.098343   0.089059  -1.104  0.27135
## z.diff.lag4   0.002819   0.088896   0.032  0.97474
## z.diff.lag5   0.072184   0.088038   0.820  0.41364
## z.diff.lag6  -0.070038   0.088374  -0.793  0.42938
```

```
## z.diff.lag7  -0.004662   0.086533  -0.054  0.95711
## z.diff.lag8   0.093595   0.086231   1.085  0.27959
## z.diff.lag9  -0.021833   0.086156  -0.253  0.80032
## z.diff.lag10 -0.014859   0.084412  -0.176  0.86052
## z.diff.lag11  0.088201   0.083820   1.052  0.29447
## z.diff.lag12  0.066938   0.083320   0.803  0.42309
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.961 on 142 degrees of freedom
## Multiple R-squared:  0.1792, Adjusted R-squared:  0.09825
## F-statistic: 2.214 on 14 and 142 DF,  p-value: 0.009843
##
##
## Value of test-statistic is: -1.1697 1.6764 1.9619
##
## 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
```

Au regard de nos graphiques d'autocorrelation et d'autocorrelation partielle, nous remarquons que la série est stationnaire.

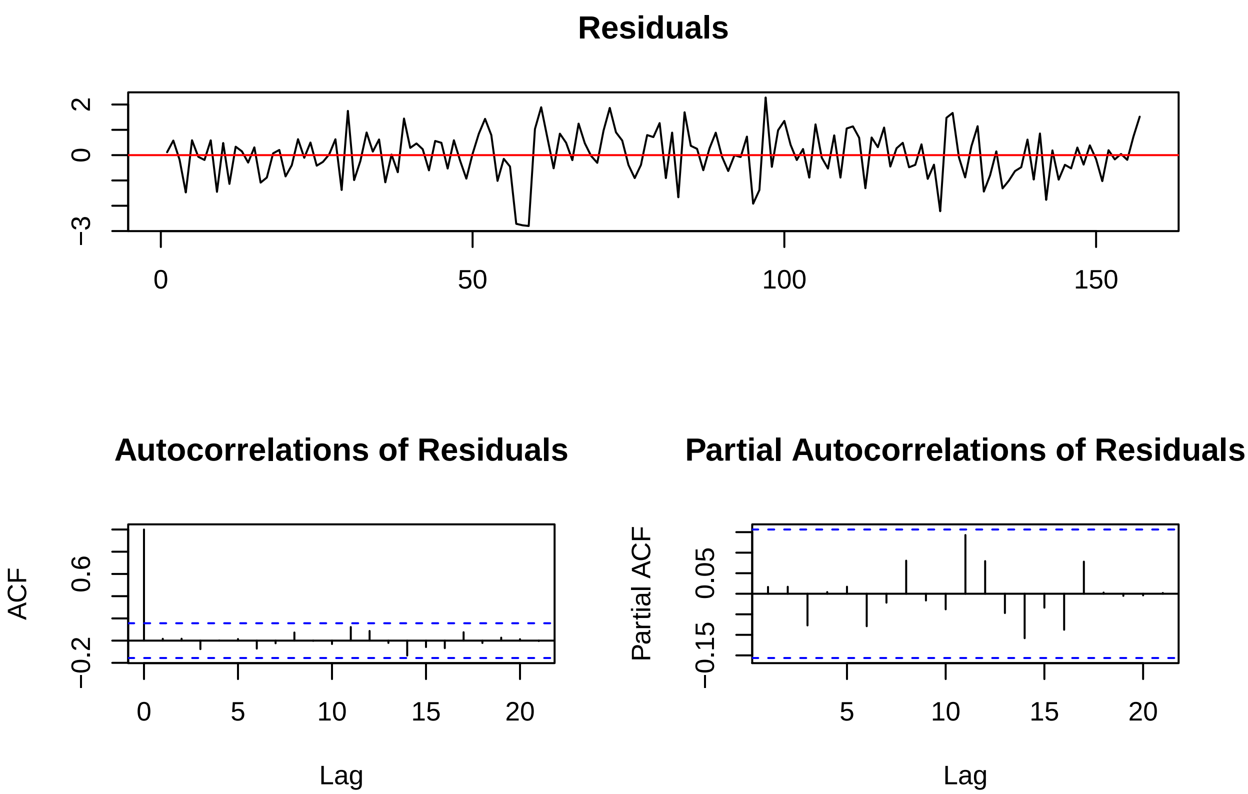
```r
df10 = ur.df(ibc, type = "trend", lags = 12, selectlags = "BIC")
plot(df10)
```



```r
summary(df10)
```

```
##
## ###############################################
## # 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
```

```r
df2 = ur.df(ibc, type = "drift", lags = 12, selectlags = "BIC")
plot(df2)
```




```r
summary(df2)
```

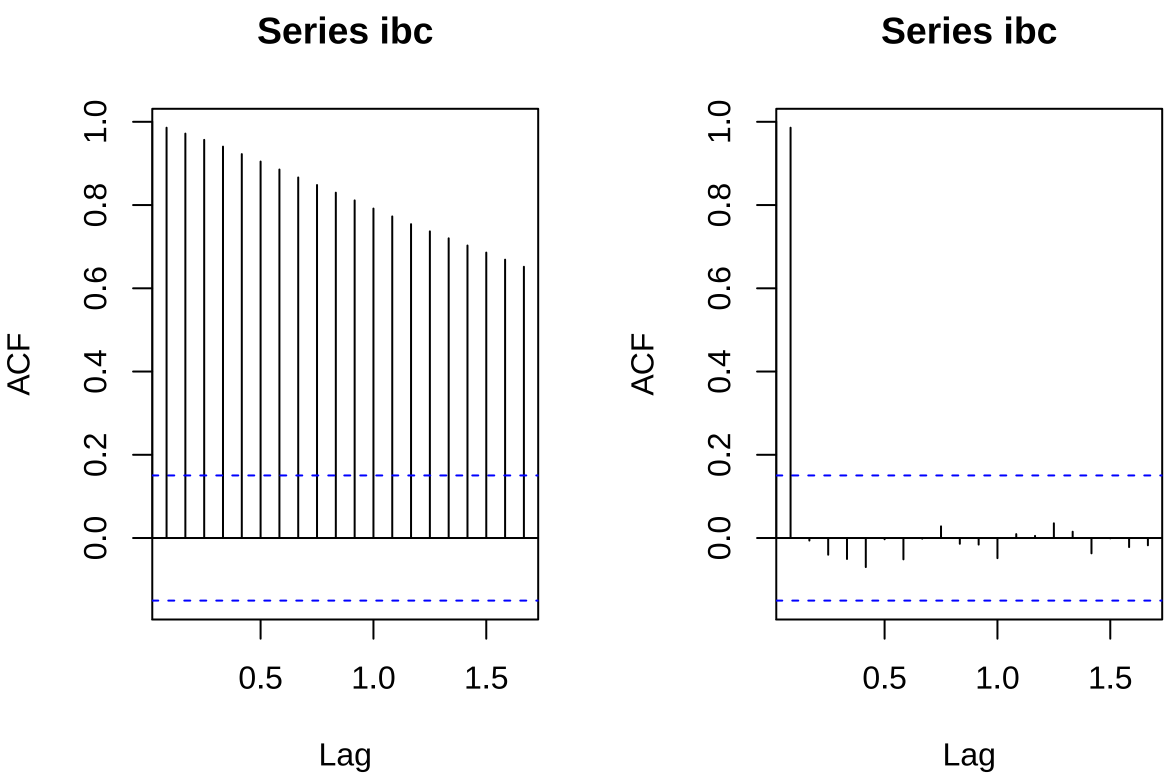
```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression drift
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + z.diff.lag)
##
## Residuals:
##      Min       1Q   Median       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
```

Ainsi, on peut donc conclure que notre série est stationnaire. On remarque aussi que toutes nos variables sont significatives au seuil de 1%, 5% et 10%.

Identification du type et de l'ordre du modèle (FACF et PACF)

```r
par(mfrow = c(1, 2))
acf(ibc, type = "correlation", lag.max = 20, drop.lag.0 = TRUE)
acf(ibc, type = "partial", lag.max = 20, drop.lag.0 = TRUE)
```



L'analyse gráphique nous montre que le modèle à estimer est AR(1).

Modèle ARIMA (1, 0, 0)

```r
model1 = Arima(ibc, order = c(1, 0, 0), method = "ML")
summary(model1)
```

```
## Series: ibc
## ARIMA(1,0,0) with non-zero mean
##
## Coefficients:
##          ar1      mean
##       0.9982  119.3511
## s.e.  0.0022   16.3932
##
## sigma^2 estimated as 1.095:  log likelihood=-250.73
## AIC=507.46   AICc=507.61   BIC=516.87
##
## Training set error measures:
##                     ME     RMSE      MAE       MPE      MAPE      MASE
## Training set 0.2155145 1.040132 0.834219 0.1762354 0.6591108 0.1596553
##                   ACF1
## Training set 0.2381553
```

Nous pouvons se tromper dans le choix du modèle pour cela nous tentons d'estimer avec ARIMA (1, 0, 1) avec degré d'intégration zéro. Nous allons enfin comparer les deux résultats et voir si notre choix a été correct.

```r
model2 <- Arima(ibc, order = c(1, 0, 1), method = "ML")
summary(model2)
```

```
## Series: ibc
## ARIMA(1,0,1) with non-zero mean
##
## Coefficients:
##          ar1     ma1      mean
##       0.9977  0.1941  119.7791
## s.e.  0.0029  0.0615   16.2366
##
## sigma^2 estimated as 1.041:  log likelihood=-246.02
## AIC=500.04   AICc=500.29   BIC=512.59
##
## Training set error measures:
##                     ME    RMSE       MAE       MPE    MAPE      MASE
## Training set 0.1846018 1.01138 0.8214544 0.1501312 0.64923 0.1572123
##                     ACF1
## Training set 0.009398223
```

La différence entre coefficients n'est pas assez considérable entre le modèle 1 et le modèle 2, nous pouvons toute fois opter pour le modèle 1.

Toute fois, nous allons faire la vérification de tests statistiques pour voir la robustesse des coefficients.

Significance statistique

```r
fit1 = coeftest(model1)
print(fit1)
```

```
##
## z test of coefficients:
##
##             Estimate Std. Error  z value  Pr(>|z|)
## ar1       9.9823e-01 2.2118e-03 451.3153 < 2.2e-16 ***
## intercept 1.1935e+02 1.6393e+01   7.2805 3.325e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Les coefficients du modèle 1 sont statistiquement significatif. Voyons alors ceux du modèle 2.

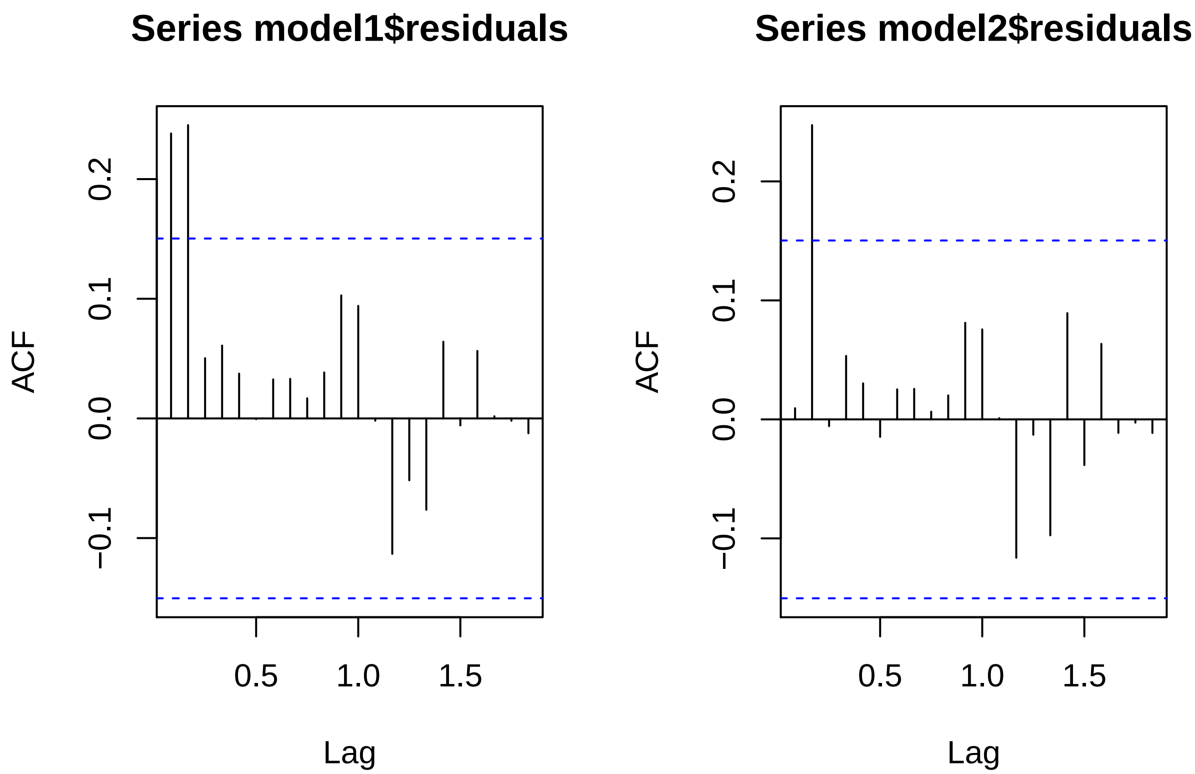
```r
fit2 <- coeftest(model2)
print(fit2)
```

```
##
## z test of coefficients:
##
##             Estimate Std. Error  z value  Pr(>|z|)
## ar1         0.997661   0.002857 349.1968 < 2.2e-16 ***
## ma1         0.194064   0.061485   3.1563  0.001598 **
## intercept 119.779051  16.236648   7.3771 1.618e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Notre modèle ARIMA (1, 0, 1) est aussi statistiquement significatif. En effet, tous les coefficients du modèle sont statistiquements significatifs. Nous passons à present aux tests d'autocorrelation des erreurs ou résiduels.

Test d'autocorrelation résiduelle

```r
par(mfrow = c(1, 2))
acf(model1$residuals, drop.lag.0 = TRUE)
acf(model2$residuals, drop.lag.0 = TRUE)
```



Test de Box-Pierce (1970) ou test de Ljung-Box (1978)

La statistique Q de Box-Pierce (1970) ou celle modifiée par Ljung-Box (1978) vérifie si les k premiers coefficients d'autocorrélations sont statistiquement égales à zéro:

```
Box.test(model1$residuals, lag = 4, type = "Box-Pierce", fitdf = 1)
```

```
##
##  Box-Pierce test
##
## data:  model1$residuals
## X-squared = 20.918, df = 3, p-value = 0.0001095
```

```
Box.test(model1$residuals, lag = 8, type = "Box-Pierce", fitdf = 1)
```

```
##
##  Box-Pierce test
##
## data:  model1$residuals
## X-squared = 21.523, df = 7, p-value = 0.003069
```

```
Box.test(model1$residuals, lag = 12, type = "Box-Pierce", fitdf = 1)
```

```
##
##  Box-Pierce test
##
## data:  model1$residuals
## X-squared = 25.123, df = 11, p-value = 0.008745
```

Le test de Box & Pierce nous dit que, pour le modèle 1, les coefficients ne sont pas pas statistiquement égales à zéro. Dans ce cas nous rejetons l'hypothèse nulle.

Test de Ljung & Box

```
Box.test(model1$residuals, lag = 4, type = "Ljung-Box", fitdf = 1)
```

```
##
##  Box-Ljung test
##
## data:  model1$residuals
## X-squared = 21.368, df = 3, p-value = 8.829e-05
```

```
Box.test(model1$residuals, lag = 8, type = "Ljung-Box", fitdf = 1)
```

```
##
##  Box-Ljung test
##
## data:  model1$residuals
## X-squared = 22.004, df = 7, p-value = 0.002536
```

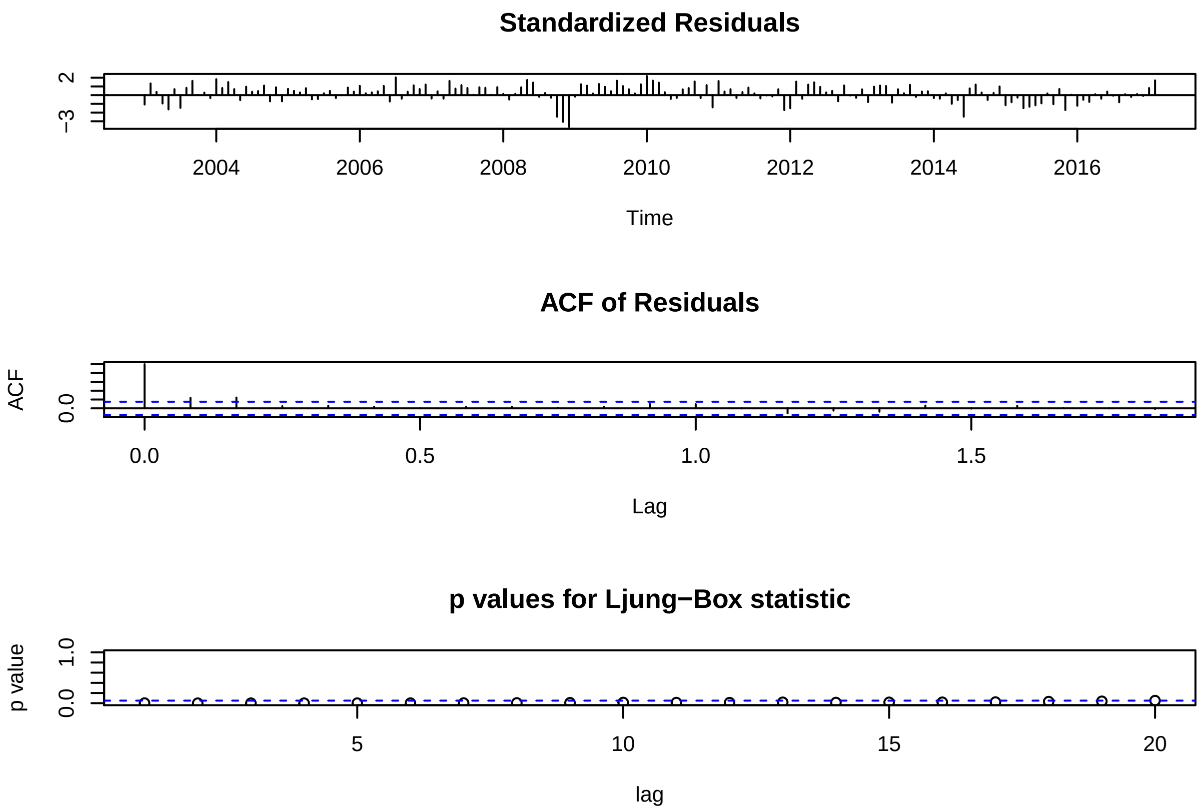
```
Box.test(model1$residuals, lag = 12, type = "Ljung-Box", fitdf = 1)
```

```
##
##  Box-Ljung test
##
## data:  model1$residuals
## X-squared = 25.907, df = 11, p-value = 0.006699
```

Même interprétation pour le test de Ljung & Box.

Analyse gráphique du test (p-valor)

```
tsdiag(model1, gof.lag = 20)
```



Test de Ljung & Box pour le modèle 2

```r
Box.test(model2$residuals, lag = 4, type = "Ljung-Box", fitdf = 2)
```

```
##
##  Box-Ljung test
##
## data:  model2$residuals
## X-squared = 11.167, df = 2, p-value = 0.00376
```

```r
Box.test(model2$residuals, lag = 8, type = "Ljung-Box", fitdf = 2)
```

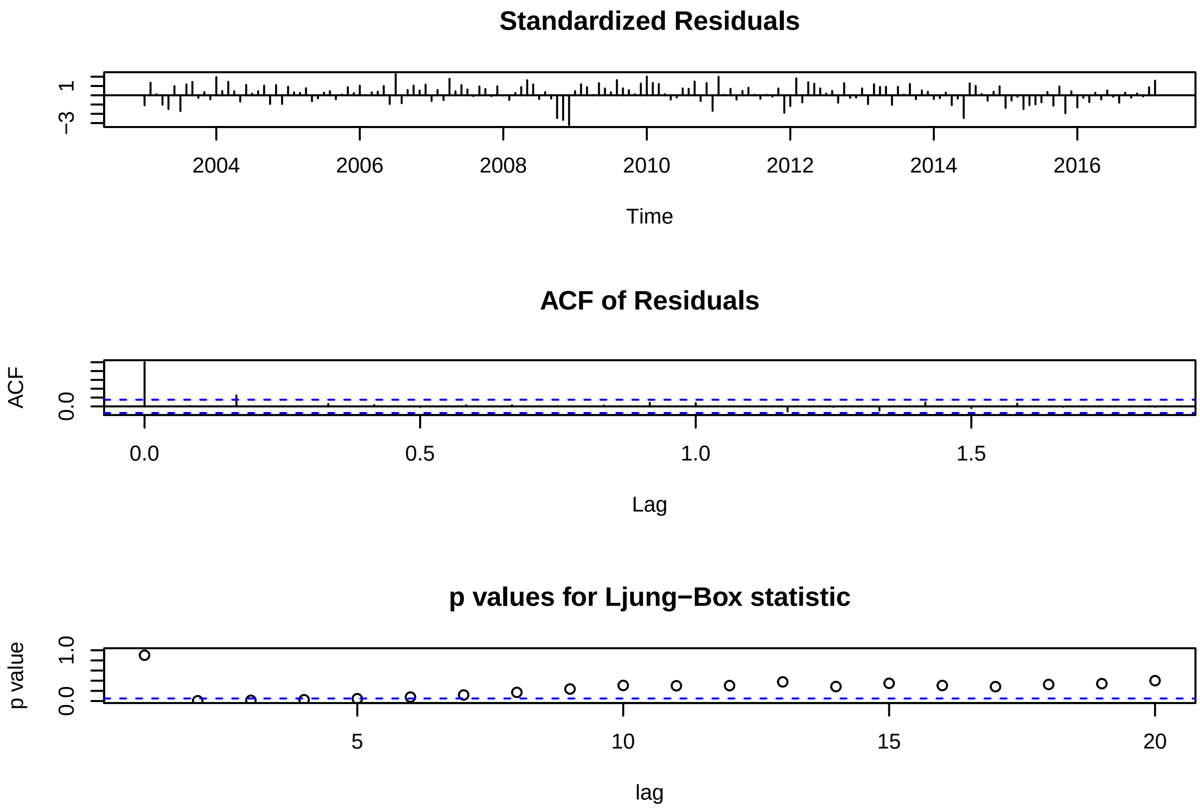
```
##
##  Box-Ljung test
##
## data:  model2$residuals
## X-squared = 11.6, df = 6, p-value = 0.0715
```

```r
Box.test(model2$residuals, lag = 12, type = "Ljung-Box", fitdf = 2)
```

```
##
##  Box-Ljung test
##
## data:  model2$residuals
## X-squared = 13.954, df = 10, p-value = 0.1751
```

Avec lag = 12, les coefficients du modèle sont statistiquement significatif, au regard du p-valor qui est supérieur au seuil de 5%. Voyons l'analyse graphique du modèle :

```
tsdiag(model2, gof.lag = 20)
```



Test de normalité des résidus

Nous utilisons ici deux tests : Jarque-Bera & Shapiro-Wilk (1965).

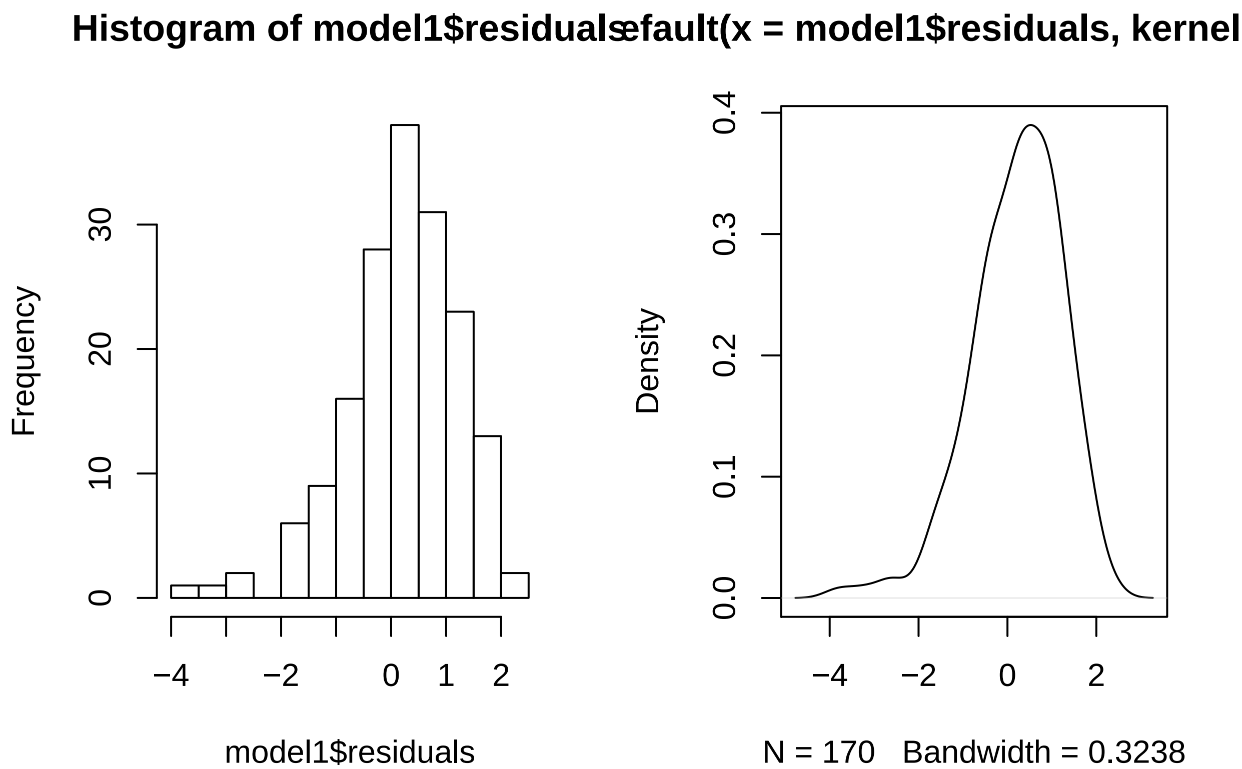
1. Test de Jarque-Bera: Ce test est logiquement utilisé pour les grands échantillons.

Le test de Jarque-Bera (JB) analyse si les moments de la série estimée (dans ce cas, les résidus) sont les mêmes que la normale. Dans cette hypothèse, l'asymétrie est égale à zéro et le kurtosis est égal à 3.

2. Test Shapiro-Wilk (1965): Peut être utilisé pour des échantillons de n'importe quelle taille.

Le test est basé sur le calcul de la statistique W qui vérifie si un échantillon aléatoire de taille n provient d'une distribution normale.

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



Nous doutons de la normalité de la série. Voyons avec les tests de Jarque-Bera et de Shapiro-Wilk.

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

```
##
##  Jarque Bera Test
##
## data:  model1$residuals
## X-squared = 29.988, df = 2, p-value = 3.077e-07
```

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

```
##
##  Shapiro-Wilk normality test
##
## data:  model1$residuals
## W = 0.96401, p-value = 0.0002213
```

L'analyse gráphique, le test de Jarque-Bera et celui de Shapiro-Wilk nous disent que les résidus ne suivent pas une loi normale.

Test d'hétéroscédasticité conditionnelle ou Homocédasticité (ARCH-LM)

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

```
##
##  ARCH LM-test; Null hypothesis: no ARCH effects
##
## data:  model1$residuals
## Chi-squared = 28.108, df = 4, p-value = 1.186e-05
```

```r
ArchTest(model1$residuals, lags = 8)
```

```
##
##  ARCH LM-test; Null hypothesis: no ARCH effects
##
## data:  model1$residuals
## Chi-squared = 30.994, df = 8, p-value = 0.0001408
```

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

```
##
##  ARCH LM-test; Null hypothesis: no ARCH effects
##
## data:  model1$residuals
## Chi-squared = 31.462, df = 12, p-value = 0.001674
```

La statistique du modèle n'est pas significative pour lag $= 4 = 8 = 12$. Nous passons au modèle 2.

```r
ArchTest(model2$residuals, lag = 4)
```

```
##
##  ARCH LM-test; Null hypothesis: no ARCH effects
##
## data:  model2$residuals
## Chi-squared = 22.995, df = 4, p-value = 0.0001269
```

```r
ArchTest(model2$residuals, lag = 8)
```

```
##
##  ARCH LM-test; Null hypothesis: no ARCH effects
##
## data:  model2$residuals
## Chi-squared = 25.953, df = 8, p-value = 0.00107
```

```r
ArchTest(model2$residuals, lag = 12)
```

```
##
##  ARCH LM-test; Null hypothesis: no ARCH effects
##
## data:  model2$residuals
## Chi-squared = 25.722, df = 12, p-value = 0.01175
```

Même interprétation pour le modèle 2. Notre série n'est toujours pas bonne pour faire la prévision, ainsi, tous les résultats des différents tests ne nous satisfont pas pour prévoir la série. Quand à cela, nous passons à la trasformation de Box-Cox avant de faire la prévision de la série.

Transformation de Box & Cox

Les Raisons pour transformer les données: stabiliser la variance et rendre additif l'effet saisonnier.

Dans le cas des séries économiques et financières, il peut être nécessaire d'appliquer à la série d'origine une transformation non linéaire, telle que la transformation logarithmique ou celle proposée par Box-Cox (1964).

```r
lambda <- BoxCox.lambda(ibc)
print(lambda)
```

```
## [1] 1.061736
```

Nous commençons avec le modèle ARIMA (1, 0, 0) puis on va terminer avec ARIMA (1, 0, 1).

Modèle ARIMA (1, 0, 0) ou AR(1)

```r
model3 <- Arima(ibc, order = c(1, 0, 0), method = "ML", lambda = lambda)
summary(model3)
```

```
## Series: ibc
## ARIMA(1,0,0) with non-zero mean
## Box Cox transformation: lambda= 1.061736
##
## Coefficients:
##          ar1      mean
##       0.9982  150.4454
## s.e.  0.0022   21.9467
##
## sigma^2 estimated as 1.992:  log likelihood=-301.6
## AIC=609.21   AICc=609.35   BIC=618.62
##
## Training set error measures:
##                     ME     RMSE       MAE       MPE      MAPE      MASE
## Training set 0.2149139 1.040195 0.8342397 0.1757054 0.6591507 0.1596592
##                   ACF1
## Training set 0.2378351
```

Modèle ARIMA (1, 0, 1) ou ARMA (1,1)

```r
model4 <- Arima(ibc, order = c(1, 0, 1), method = "ML", lambda = lambda)
summary(model4)
```

```
## Series: ibc
## ARIMA(1,0,1) with non-zero mean
## Box Cox transformation: lambda= 1.061736
##
## Coefficients:
##          ar1     ma1     mean
##       0.9976  0.1941  150.860
```

```
## s.e.  0.0029  0.0615   21.709
##
## sigma^2 estimated as 1.895:  log likelihood=-296.89
## AIC=601.79   AICc=602.03   BIC=614.33
##
## Training set error measures:
##                     ME     RMSE       MAE       MPE      MAPE      MASE
## Training set 0.1843196 1.011452 0.8215037 0.1498475 0.6492867 0.1572218
##                     ACF1
## Training set 0.009142207
```

Nous allons à présent vérifier si lequel des deux modèles est meilleur pour faire la prévision. Pour ainsi, nous allons vérifier les coefficients de chaque modèle s'ils sont significatifs ou non après la transformation de Box_Cox.

```r
fit3 <- coeftest(model3)
print(fit3)
```

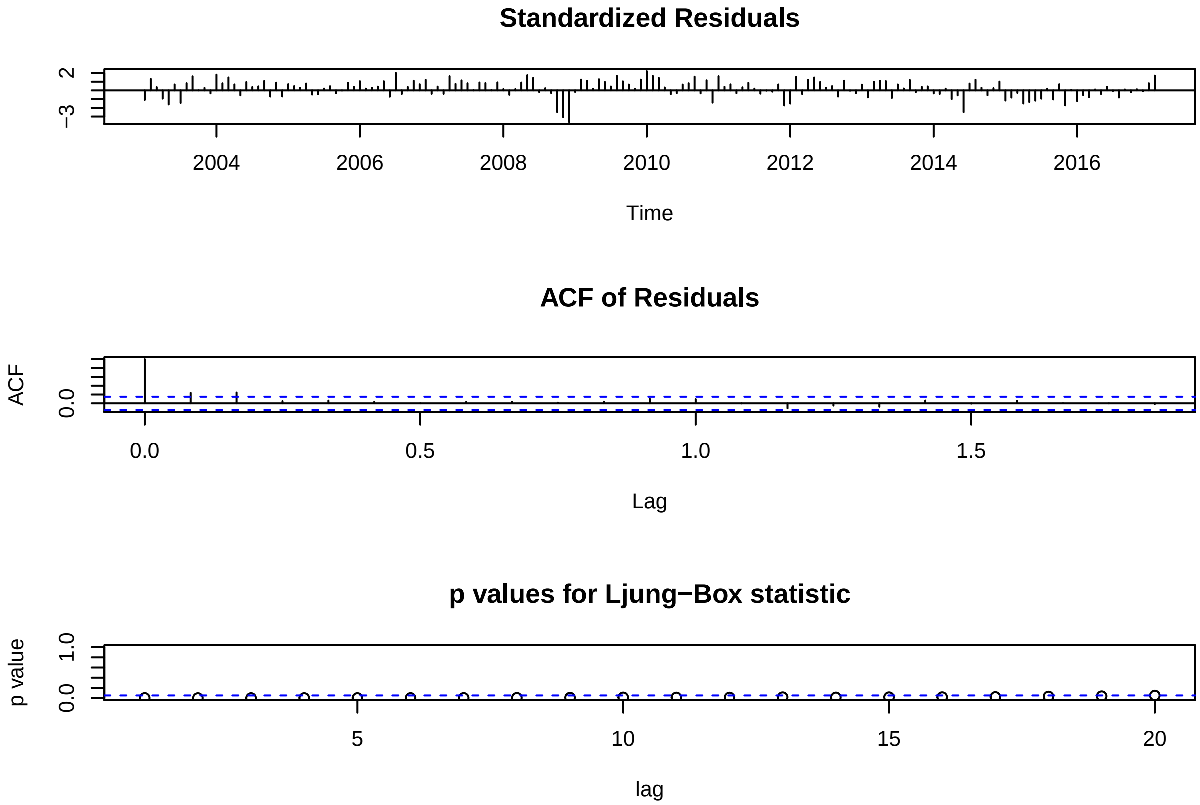
```
##
## z test of coefficients:
##
##             Estimate Std. Error z value  Pr(>|z|)
## ar1       9.9821e-01 2.2311e-03 447.401 < 2.2e-16 ***
## intercept 1.5045e+02 2.1947e+01   6.855 7.129e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
fit4 <- coeftest(model4)
print(fit4)
```

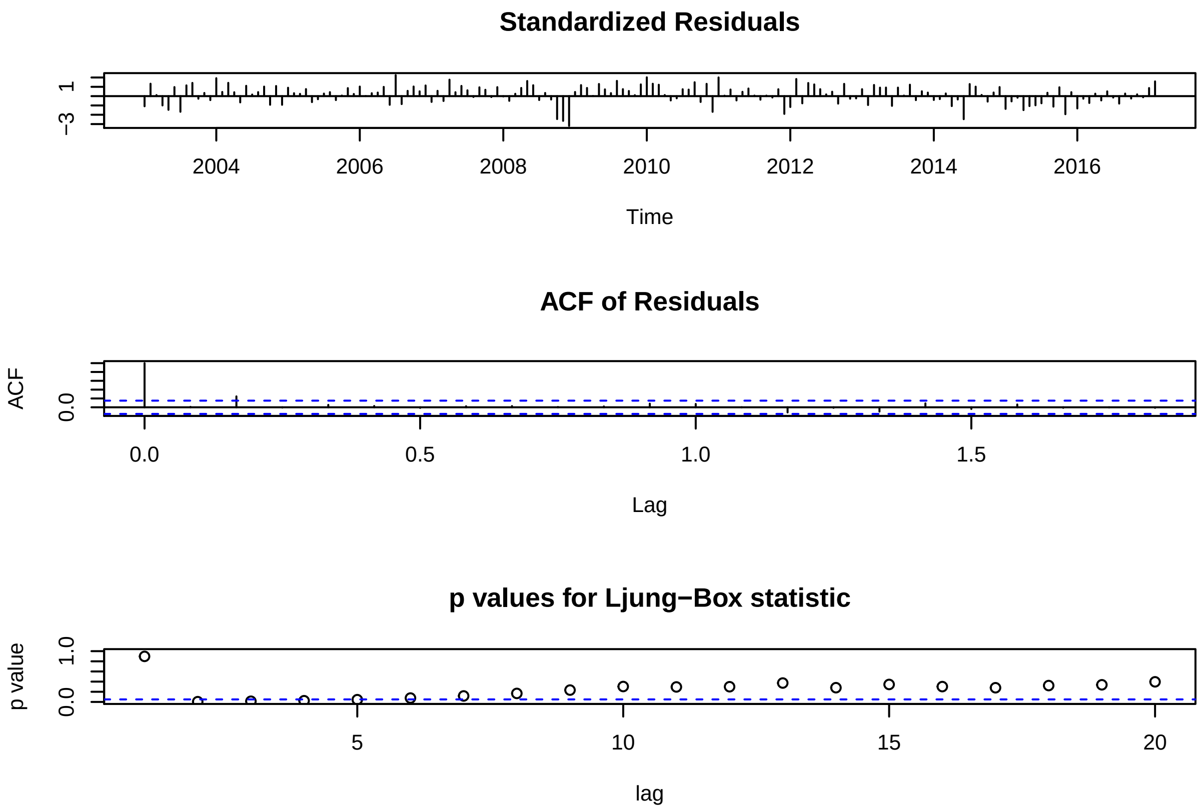
```
##
## z test of coefficients:
##
##             Estimate Std. Error  z value  Pr(>|z|)
## ar1       9.9763e-01 2.8824e-03 346.1140 < 2.2e-16 ***
## ma1       1.9409e-01 6.1466e-02   3.1577   0.00159 **
## intercept 1.5086e+02 2.1709e+01   6.9492 3.674e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Les coefficients de nos deux modèles sont statistiquement significatif. Vérifions aussi à l'aide du graphique de p-valor de Ljung-Box.

```r
tsdiag(model3, gof.lag = 20)
```

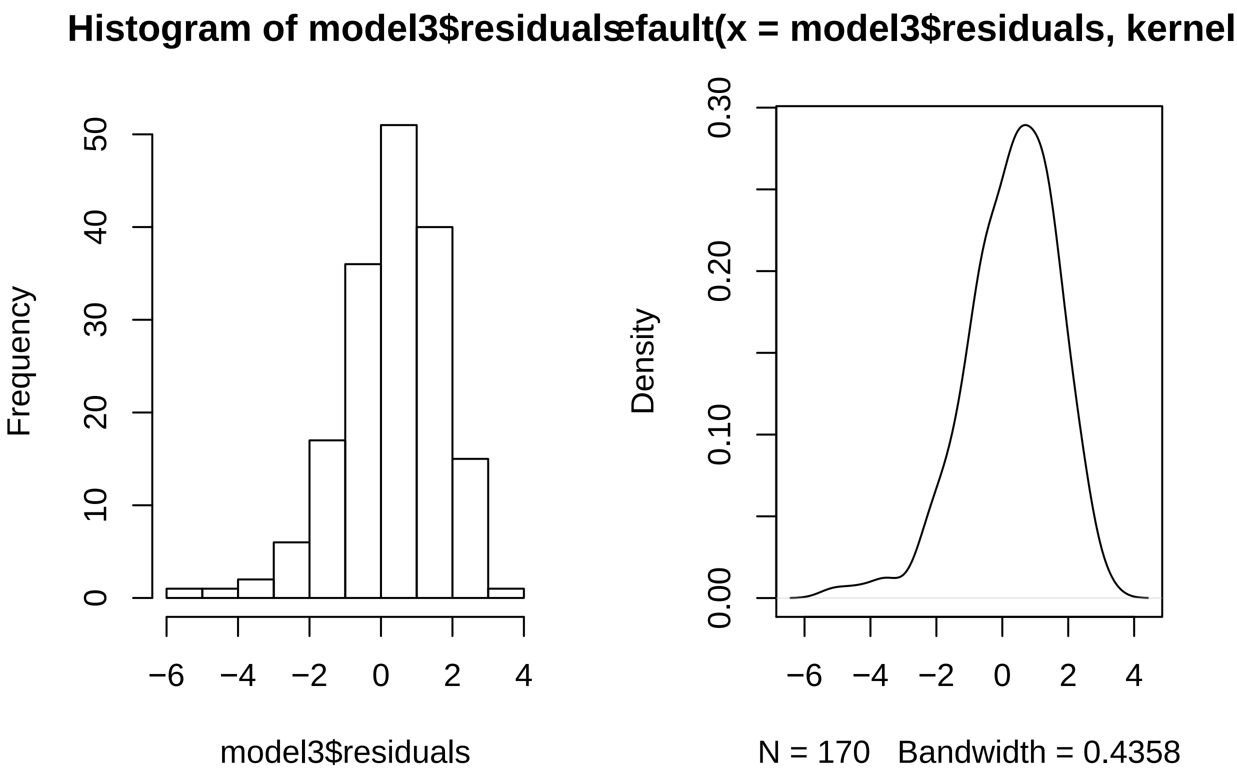



```r
tsdiag(model4, gof.lag = 20)
```



Il y a une amélioration après la transformation de Box & Cox. Nous pouvons encore vérifier à l'aide de test de normalité des résidus.

```
par(mfrow = c(1, 2))
hist(model3$residuals)
plot(density(model3$residuals, kernel = "gaussian"))
```



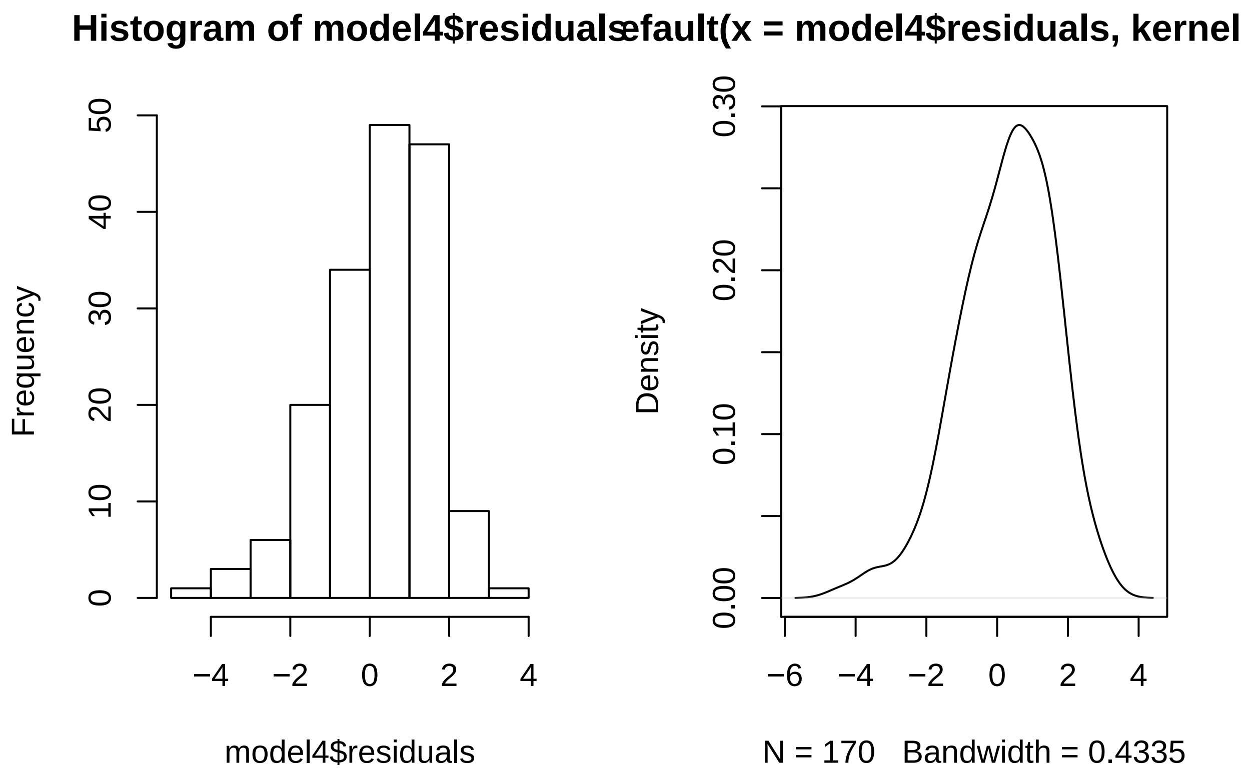

```
jarque.bera.test(model3$residuals)
```

```
##
##  Jarque Bera Test
##
## data:  model3$residuals
## X-squared = 29.87, df = 2, p-value = 3.265e-07
```

```
shapiro.test(model3$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  model3$residuals
## W = 0.96385, p-value = 0.000213
```

```
par(mfrow = c(1, 2))
hist(model4$residuals)
plot(density(model4$residuals, kernel = "gaussian"))
```



Nous remarquons que après transformation notre série s'est améliorée.

```
jarque.bera.test(model4$residuals)
```

```
##
##  Jarque Bera Test
##
## data:  model4$residuals
## X-squared = 12.516, df = 2, p-value = 0.001915
```

```
shapiro.test(model4$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  model4$residuals
## W = 0.97605, p-value = 0.004885
```

Test d'hétéroscédasticité conditionnelle

Modèle 3

```
ArchTest(model3$residuals, lag = 4)
```

```
##
```

```
##  ARCH LM-test; Null hypothesis: no ARCH effects
##
## data:  model3$residuals
## Chi-squared = 28.27, df = 4, p-value = 1.099e-05
```

```r
ArchTest(model3$residuals, lag = 8)
```

```
##
##  ARCH LM-test; Null hypothesis: no ARCH effects
##
## data:  model3$residuals
## Chi-squared = 31.068, df = 8, p-value = 0.0001367
```

```r
ArchTest(model3$residuals, lag = 12)
```

```
##
##  ARCH LM-test; Null hypothesis: no ARCH effects
##
## data:  model3$residuals
## Chi-squared = 31.468, df = 12, p-value = 0.001671
```

Modèle 4

```r
ArchTest(model4$residuals, lag = 4)
```

```
##
##  ARCH LM-test; Null hypothesis: no ARCH effects
##
## data:  model4$residuals
## Chi-squared = 23.064, df = 4, p-value = 0.0001229
```

```r
ArchTest(model4$residuals, lag = 8)
```

```
##
##  ARCH LM-test; Null hypothesis: no ARCH effects
##
## data:  model4$residuals
## Chi-squared = 25.969, df = 8, p-value = 0.001063
```

```r
ArchTest(model4$residuals, lag = 12)
```

```
##
##  ARCH LM-test; Null hypothesis: no ARCH effects
##
## data:  model4$residuals
## Chi-squared = 25.697, df = 12, p-value = 0.01184
```

Avant de passer à la décision sur le modèle à estimer, nous devons tout d'abord effectuer un choix du modèle. Pour cela, les critères AIC & BIC nous permettrons de le réaliser. Il s'agit d'effectuer AIC et BIC le plus bas possible entre les deux modèle c'est à dire le modèle 3 ou 4 que nous avons transformés.

```r
AIC(model3, model4)
```

```
##        df      AIC
## model3  3 609.2094
## model4  4 601.7859
```

```r
BIC(model3, model4)
```

```
##        df      BIC
## model3  3 618.6167
## model4  4 614.3291
```

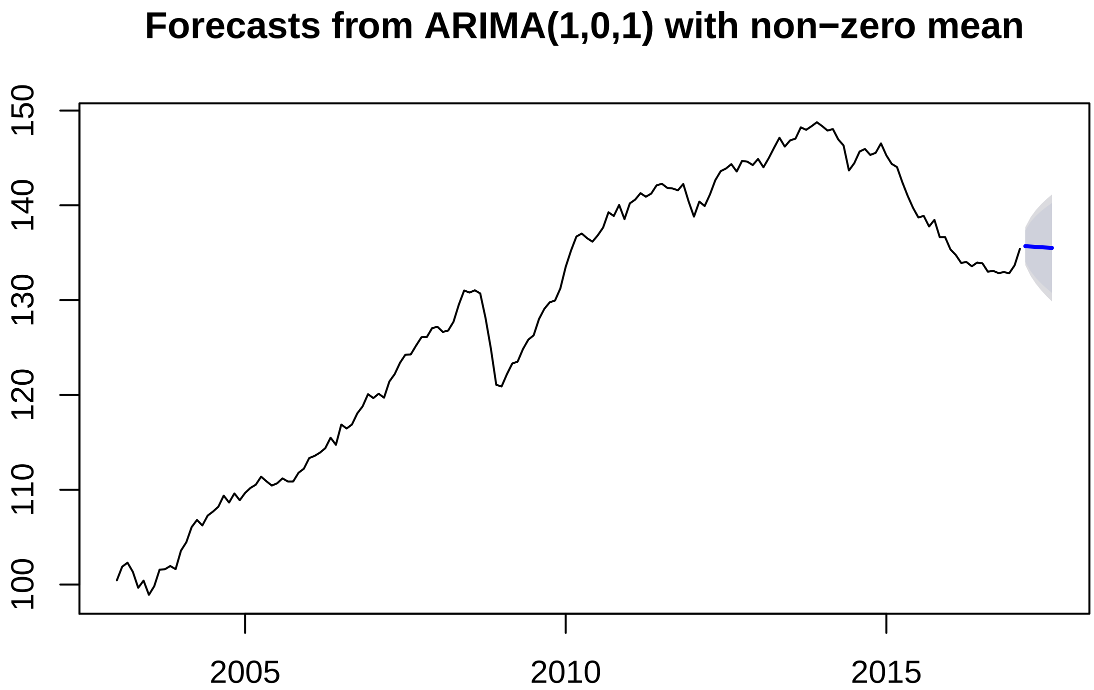
Nous choisirons le modèle 4 ou soit ARIMA (1, 0, 1) puisque AIC et BIC sont inférieur au modèle 3. Ainsi, la prévision sera réalisé à partir du modèle 4.

Prévision

```r
prev = forecast(model4, h = 6, level = c(0.90, 0.95), lambda = lambda,
                biasadj = FALSE)
print(prev)
```

```
##          Point Forecast    Lo 90    Hi 90    Lo 95    Hi 95
## Mar 2017       135.6959 134.0233 137.3672 133.7027 137.6873
## Apr 2017       135.6586 133.0560 138.2582 132.5570 138.7559
## May 2017       135.6215 132.3452 138.8928 131.7170 139.5190
## Jun 2017       135.5844 131.7534 139.4087 131.0187 140.1406
## Jul 2017       135.5474 131.2343 139.8520 130.4070 140.6757
## Aug 2017       135.5105 130.7658 140.2449 129.8557 141.1507
```

```r
par(mfrow = c(1, 1))
plot(prev)
```


Forecasts from ARIMA(1,0,1) with non−zero mean
150
140
130
120
110
100
2005
2010
2015