University of Campinas - UNICAMP

Time Series Analysis

Gross Domestic Product-French (GDP-current US$)

Supervisor: Rosangela Balini

———————————————————

Henri Makika (211042)

———————————————————

May 25, 2019

Introduction

Dans ce travail, il est donc question d'estimer, en utilisant le modèle ARIMA, les données de PIB de la France exprimées en dollars américains. La serie de données commence de 1960 à 2017. Pour ainsi, nous utilisons la méthodologie de Box-Jenkins et réaliser la prévision pour cinq (5) années après, c'est à dire 2018, 2019, 2020, 2021, 2022. Quatre opérations nous est demandée de réaliser :

1. Identification
2. Estimation
3. Vérification et
4. Prévision

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

```r
library(strucchange) # Pour le test de rupture structuralle
library(forecast)    # Pour ajuster et faire la prévision du modèle ARIMA
library(lmtest)      # Pour les tests d'hypothès de normalité
library(FinTS)       # Pour tester l'hypothèse d'hétérocédasticité
library(urca)        # Pour réaliser le test de racines unitaires (KPSS, ERS,...)
library(tseries)     # Pour les tests de normalité
library(TSA)         # Pour les tests de l'opérateur de retard (lags)
library(readxl)      # Pour lire les données xlsx (données sur Excel)
```

Lecture des données

```r
pib_fr <- read_excel("~/Videos/Unicamp_IE 2019/HO:236A Times Series/Ativ 1/Ativ 2/France.xlsx")
head(pib_fr)
```

```
## # A tibble: 6 x 2
##     Ano `GDP (current US$)`
##   <dbl>               <dbl>
## 1  1960        62651474947.
## 2  1961        68346741504.
## 3  1962        76313782252.
## 4  1963        85551113767.
## 5  1964        94906593388.
## 6  1965       102160571409.
```

Analyse de données

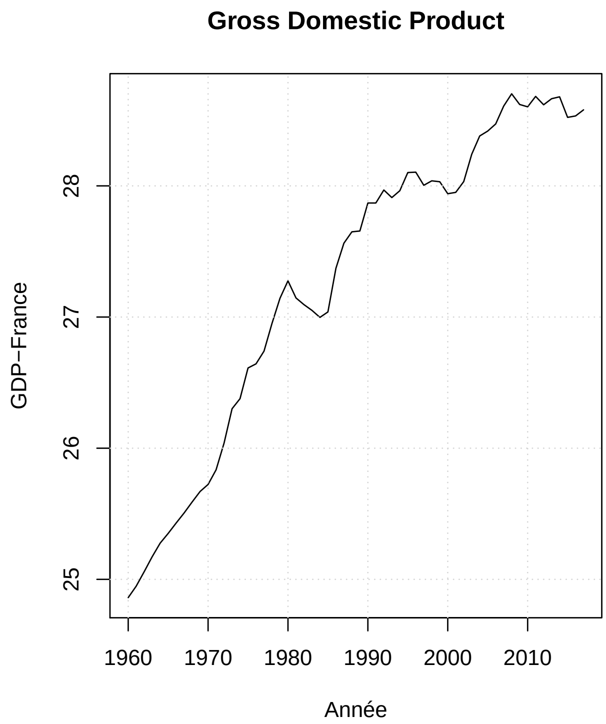
Il est donc ici question de transformer notre base de données en série temporelle, il s'agit donc d'exécuter la commande :

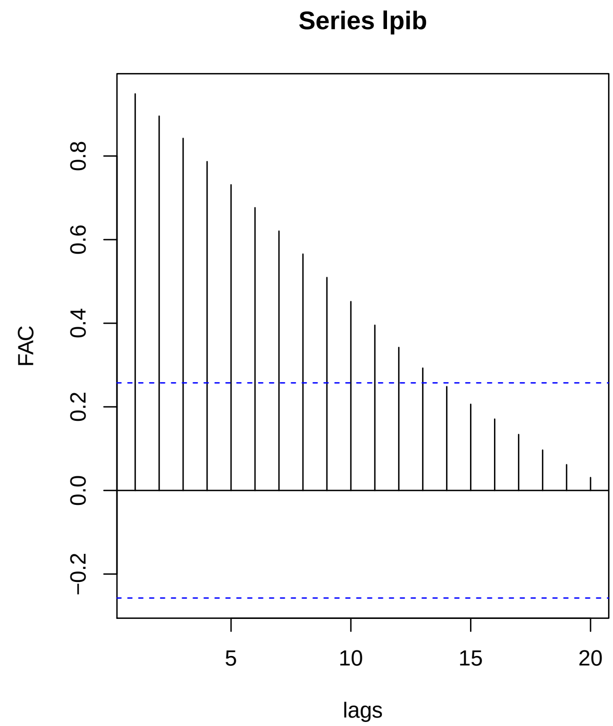
```r
pib = ts(pib_fr[,2], start = c(1960, 1), frequency = 1)
plot(pib, main = "Gross Domestic Product", xlab = "Année", ylab = "GDP-France")
grid()
```

1. Identification du modèle

Premièrement, nous aimerions savoir si la serie a une tendance ou pas, si elle est stationnaire ou non. Pour se faire, nous allons identifier par la méthode graphque.

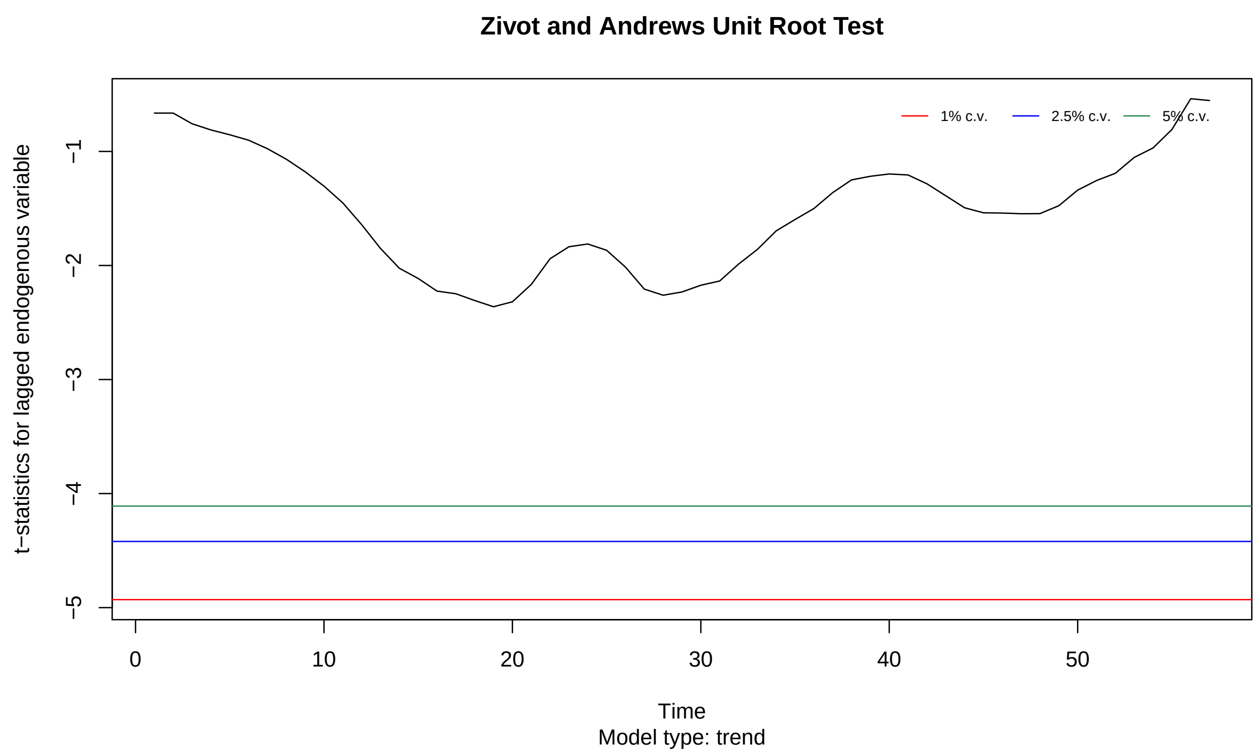
```
lpib <- log(pib)

par(mfrow = c(1, 2))
plot(lpib, main = "Gross Domestic Product", xlab = "Année", ylab = "GDP-France")
grid()
acf(lpib, lag.max = 20, type = "correlation", xlab = "lags",
    ylab = "FAC")
```





La *Gross Domestic Product* de la France n'est pas stationnaire et a une tendance temporelle. Tout au long du graphique de la serie, nous remarquons les chocs négatifs, cela peuvent-ils causer une *quebra estrutural*? Pour ainsi, nous vérifions en réalisant le test de Zivot et Andrews :

```
za.lpib <- ur.za(lpib, model = "trend")
plot(za.lpib)
```



Le graphique de Zivot et Andrews ne nous montre pas une rupture structurelle, ces chocs négatifs ne peuvent qu'être des chocs aléatoires endogènes.

```r
summary(za.lpib)
```

```
##
## ################################
## # Zivot-Andrews Unit Root Test #
## ################################
##
##
## Call:
## lm(formula = testmat)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.187272 -0.045679 -0.006131  0.059685  0.205758
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.223376   1.745555   2.420   0.0190 *
## y.l1         0.831240   0.071454  11.633 3.27e-16 ***
## trend        0.022033   0.009784   2.252   0.0285 *
## dt          -0.016031   0.006750  -2.375   0.0212 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.09093 on 53 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.9942, Adjusted R-squared:  0.9939
## F-statistic:  3018 on 3 and 53 DF,  p-value: < 2.2e-16
##
##
## Teststatistic: -2.3618
## Critical values: 0.01= -4.93 0.05= -4.42 0.1= -4.11
##
## Potential break point at position: 19
```

Le resultat du tableau nous montre qu'il y a une seule rupture struturelle (à la 19e pisition). Nous passons à présent au test de stationnarité de notre série.

Une des grandes questions dans l'étude de séries temporelles (chronologiques) est de savoir si celles-ci suivent un processus stationnaire. On entend par là le fait que la structure du processus sous-jacent supposé évolue ou non avec le temps. Si la structure reste la même, le processus est dit alors stationnaire.

On a deux types de stationnarité : la stationnarité forte et la stationnarité faible.

1. Une stationnarité est dit forte si : soit un processus temporel à valeurs réelles et en temps discret $Z_1, Z_2, ..., Z_t$. Il est dit stationnaire au sens fort si pour toute fonction f mesurable :

$$f(Z_1, Z_2, ..., Z_t) et f(Z_{1+k}, Z_{2+k}, ...Z_{t+k})$$

ont la même loi.

2. Soit un processus temporel à valeurs réelles et en temps discret $Z_1, Z_2, ..., Z_t$. Il est dit stationnaire au sens faible (ou «de second ordre», ou «en covariance») si :

$$E[Z_i] = \mu, \forall_i = 1...t$$

$$Var[Z_i] = \delta^2 \neq \infty, \forall_i = 1...t$$
$$Cov[Z_i, Z_{i-t}] = f(x) = \rho_k, \forall_i = 1...t, \forall_k = 1...t$$

La notion de stationnarité est importante dans la modélisation de séries temporelles, le problème de régression fallacieuse montrant qu'une régression linéaire avec des variables non-stationnaires n'est pas valide. Plus précisément, la distribution des paramètres de la régression ne suit plus une *loi de Student* mais un mouvement brownien. Dans le cas où les variables ne sont pas stationnaires, un concept très proche, celui de coïntégration, permet de déterminer le type de modèle à utiliser.

La stationnarité joue également un rôle important dans la prédiction de séries temporelles, l'intervalle de prédiction étant différent selon que la série est stationnaire ou non. on a deux types de non-stationnarité :

Lorsqu'une ou plus des conditions de stationnarité n'est pas remplie, la série est dite non-stationnaire. Ce terme recouvre cependant de nombreux types de non-stationnarité, dont deux sont ici exposés.

1. *Stationnarité en tendance* : Une série est stationnaire en tendance si la série obtenue en « enlevant » la tendance temporelle de la série originale est stationnaire.

Soit $X_t = bt + \varepsilon_t$, est dit stationnaire si : $Y_t = bt + \varepsilon_t - bt = \varepsilon_t$ où ε_t est un *bruit blanc*.

Soit $Y_t = X_t - bt$, est stationnaire si : $Y_t = bt + \varepsilon_t - bt = \varepsilon_t$ où ε_t est une *bruit blanc*.

2. *Stationnarité en différence* : soit $\Delta X_t = X_t - X_{t-1}$. Une série est stationnaire en différence si la série obtenue en différenciant les valeurs de la série originale est stationnaire.

3. *Ordre d'intégration d'une série temporelle* : Une série temporelle est dite intégrée d'ordre d, que l'on note $I(d)$, si la série obtenue après $diff$ différenciations est stationnaire.

Tests de stationnarité

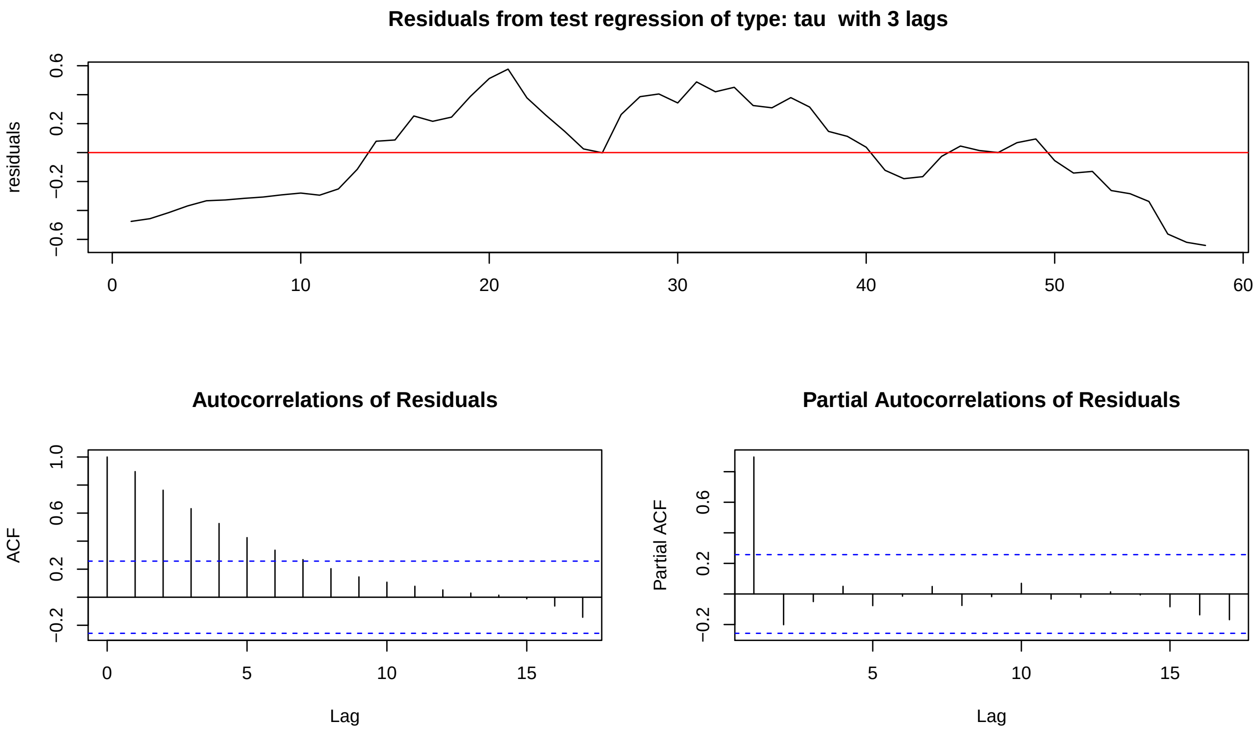
Si la fonction de densité n'est pas connue, ce qui est souvent le cas, il est utile de pouvoir déterminer par un test si la série est stationnaire ou non. Il en existe deux types, avec la stationnarité comme hypothèse nulle ou hypothèse alternative :

L'hypothèse nulle est la stationnarité (test de stationnarité): ici on fait le test de KPSS (1992); L'hypothèse nulle est la non-stationnarité (test de racine unitaire): il y a plusieurs tests dont le test de DF ou ADF (1981), le test de Phillips-Perron (1988) et le test de Elliot, Rothenberg et Stock [DF-GLS ou ERS (1996)].

Commençons par le test de stationnarité de KPSS :

1. Modèle avec tendance

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



Le test graphique étant informel ne va donc pas nous dire exactement si la série est stationnaire ou pas. S'il faut réjeter l'hypothèse nulle ou pas. Mais informellement il nous dit que la série n'est pas stationnaire. Donc on peut réjeter l'hypothèse nulle qui dit que la série est stationnaire.

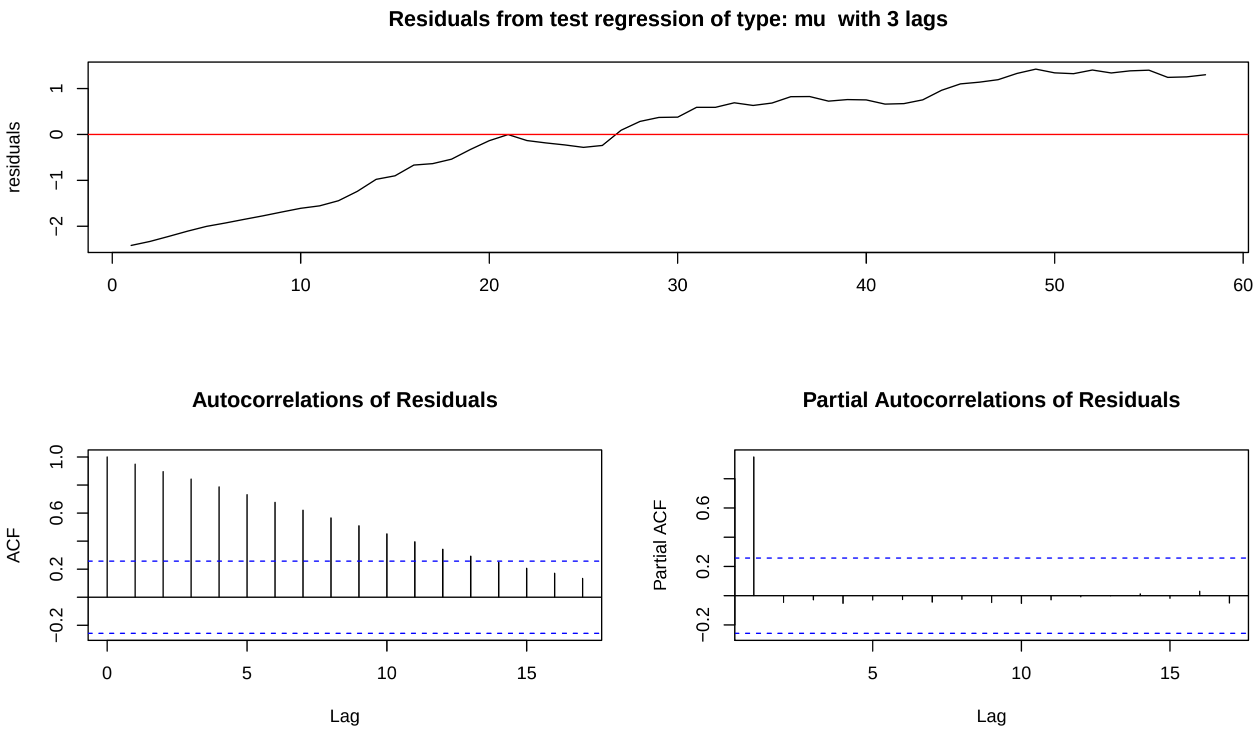
```
summary(kpss.lpib)
```

```
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: tau with 3 lags.
##
## Value of test-statistic is: 0.3373
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.119 0.146  0.176 0.216
```

On rejet l'hypothèse nulle qui dit que la série est stationnaire. L'analyse statistique montre que la *Critical value for a significance*, au seuil de 5% est en dehors de l'intervalle de confiance, cette valeur du test est de 0.146.

2. Modèle avec constance

```
kpss.pib1 <- ur.kpss(lpib, type = "mu", lags = "short")
plot(kpss.pib1)
```



Le modèle avec constante nous renseigne la même chose. La série n'est pas stationnaire. On rejet donc l'hypothèse nulle de la stationnarité.

```r
summary(kpss.pib1)
```

```
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 3 lags.
##
## Value of test-statistic is: 1.4621
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

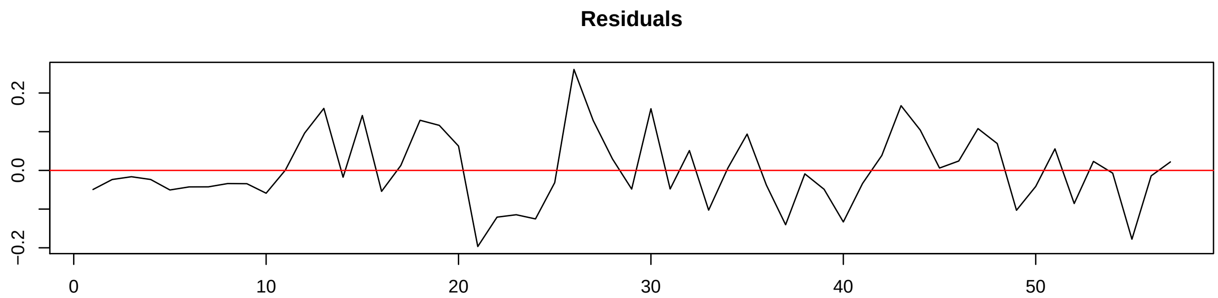
Aussi, le test statistique du test de KPSS nous dit de réjeter l'hypothèse nulle. La série est non stationnaire (au seuil de 5% $0.463 < 1.4621$). Nous passons à l'hypothèse nulle de la non-stationnarité.

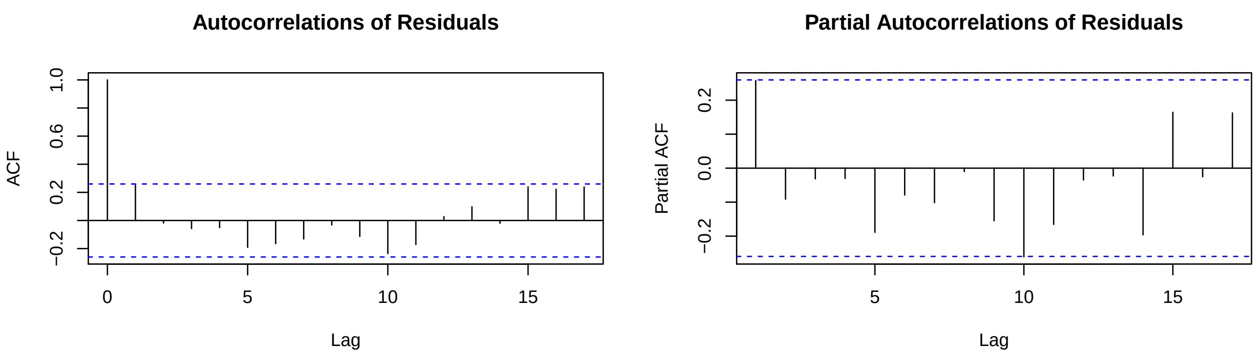
Test de racine unitaire

Le test de Dickey-Fuller (DF)

Modèle avec intercept et tendance

```r
df.lpib <- ur.df(lpib, type = "trend", lags = 0)
plot(df.lpib)
```





Le test de DF avec tendance et intercept nous indique de choisir le modèle AR(1), bien que cela soit difficile de le prédire. Bien que ça, la série n'est pas stationnaire.

```r
summary(df.lpib)
```

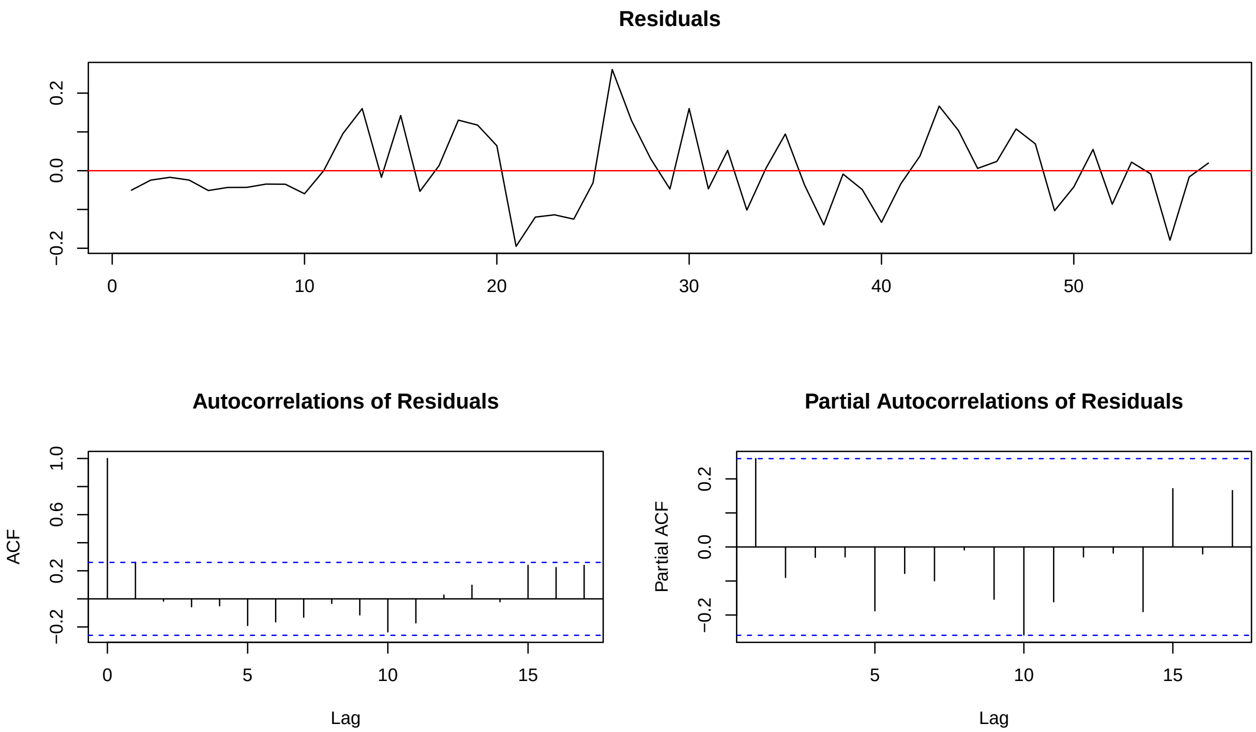
```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression trend
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + tt)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.19675 -0.04880 -0.01636  0.05561  0.26089
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.8154518  1.0385858   0.785    0.436
## z.lag.1     -0.0273050  0.0411286  -0.664    0.510
## tt          -0.0002062  0.0029531  -0.070    0.945
##
## Residual standard error: 0.09476 on 54 degrees of freedom
## Multiple R-squared:  0.1293, Adjusted R-squared:  0.09706
## F-statistic:  4.01 on 2 and 54 DF,  p-value: 0.02379
##
##
```

```
## Value of test-statistic is: -0.6639 11.6812 4.0099
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau3 -4.04 -3.45 -3.15
## phi2  6.50  4.88  4.16
## phi3  8.73  6.49  5.47
```

Les coefficients du modèle ne sont pas significatifs. Et notre série n'est pas non plus stationnaire.

Modèle sans tendance avec intercept

```r
df1.lpib <- ur.df(lpib, type = "drift", lags = 0)
plot(df1.lpib)
```



Le modèle sans tendance mais avec intercept nous amènes à conclure de la même manière que ci-haut. On ne peut pas réjeter l'hypothèse nulle qui dit que la série n'est pas stationnaire.

```r
summary(df1.lpib)
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression drift
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

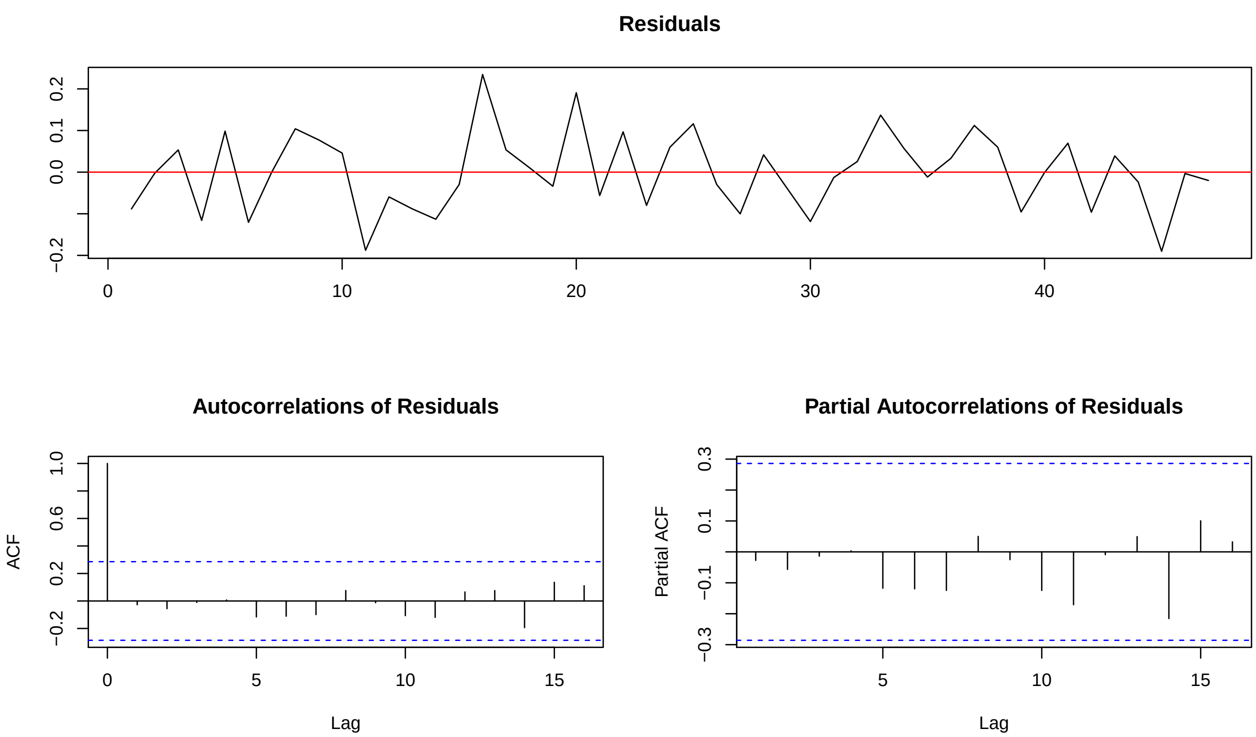
```
## -0.19504 -0.04869 -0.01710  0.05481  0.26091
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.88509    0.28723   3.081  0.00321 **
## z.lag.1     -0.03008    0.01053  -2.857  0.00603 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.09389 on 55 degrees of freedom
## Multiple R-squared:  0.1292, Adjusted R-squared:  0.1134
## F-statistic: 8.163 on 1 and 55 DF,  p-value: 0.006025
##
##
## Value of test-statistic is: -2.857 17.8423
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau2 -3.51 -2.89 -2.58
## phi1  6.70  4.71  3.86
```

Bien que les coefficients du modèle sont statistiquement significatifs, la série n'est pas stationnaire. On ne rejet pas l'hypothèse nulle. Nous appliquons alors le test de ADF (Dickey-Fuller Augmented).

Test de Dickey-Fuller Augmented

Modèle avec tendance et intercepet

```
adf.lpib <- ur.df(lpib, type = "trend", lags = 10, selectlags = "BIC")
plot(adf.lpib)
```



Le graphique d'autocorrélation partielle nous indique le modèle MA(0) et l'autocorrélation nous indique AR(1). Il y a une différence entre les deux graphiques de l'autocorellation et autocorellation partielle pour le test de Dickey-Fuller et Dickey-Fuller Augmented.

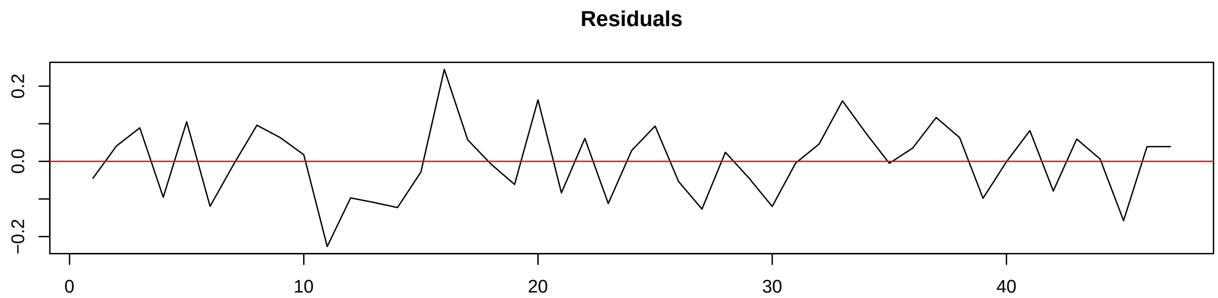
```
summary(adf.lpib)
```

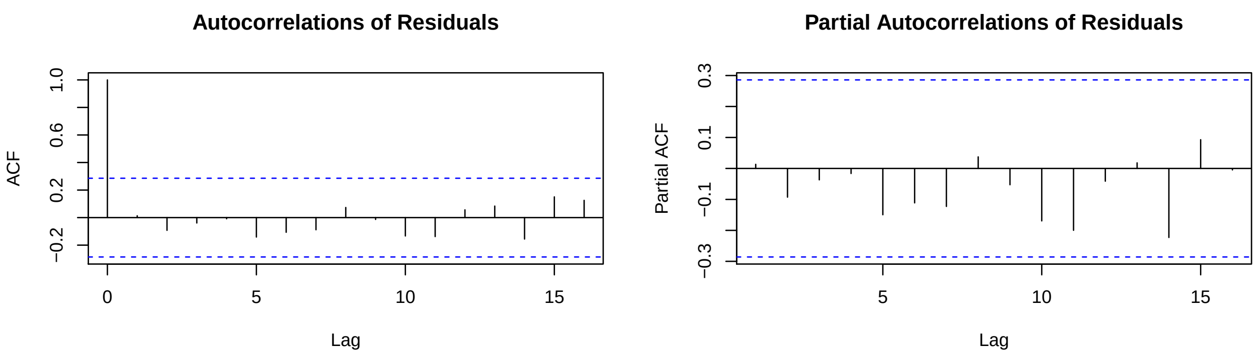
```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression trend
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.190166 -0.069639 -0.002274  0.058239  0.234591
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.900664   1.496788   2.606   0.0125 *
## z.lag.1     -0.147177   0.058359  -2.522   0.0155 *
## tt           0.006354   0.003626   1.752   0.0869 .
## z.diff.lag   0.282771   0.140524   2.012   0.0505 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.09515 on 43 degrees of freedom
## Multiple R-squared:  0.2901, Adjusted R-squared:  0.2405
## F-statistic: 5.857 on 3 and 43 DF,  p-value: 0.001908
##
##
## Value of test-statistic is: -2.5219 5.8137 5.4718
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau3 -4.04 -3.45 -3.15
## phi2  6.50  4.88  4.16
## phi3  8.73  6.49  5.47
```

Notre série est toujours non stationnaire, puisque la valeur critique au seuil de 5% nous indique qu'elle est en dehors de l'intervalle de confiance.

Modèle sans tendance et avec intercept

```
adf1.lpib <- ur.df(lpib, type = "drift", lags = 10, selectlags = "BIC")
plot(adf1.lpib)
```





L'analyse graphique du modèle sans tendance mais avec intercept nous dit la même chose, il s'agit du modèle AR(1). Nous vérifions la stationnarité avec le résultat de test.

```r
summary(adf1.lpib)
```

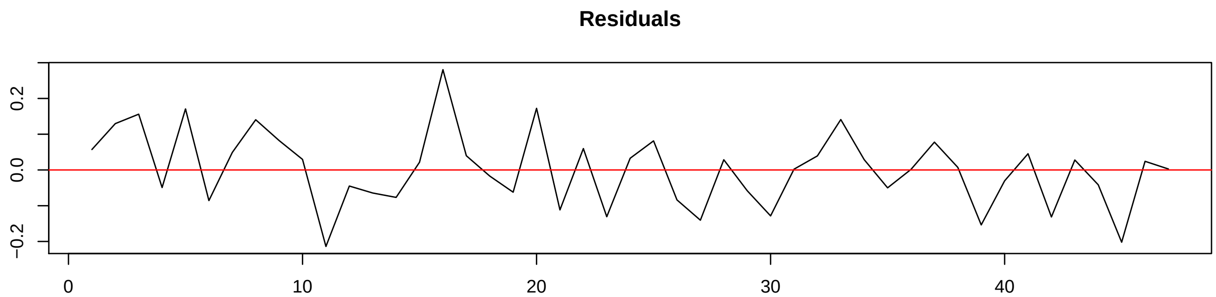
```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression drift
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + z.diff.lag)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.226492 -0.081253  0.006089  0.061991  0.244435
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.42458    0.50496   2.821  0.00715 **
## z.lag.1     -0.04975    0.01814  -2.742  0.00879 **
## z.diff.lag   0.21454    0.13816   1.553  0.12763
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.09737 on 44 degrees of freedom
## Multiple R-squared:  0.2394, Adjusted R-squared:  0.2048
## F-statistic: 6.924 on 2 and 44 DF,  p-value: 0.00243
```

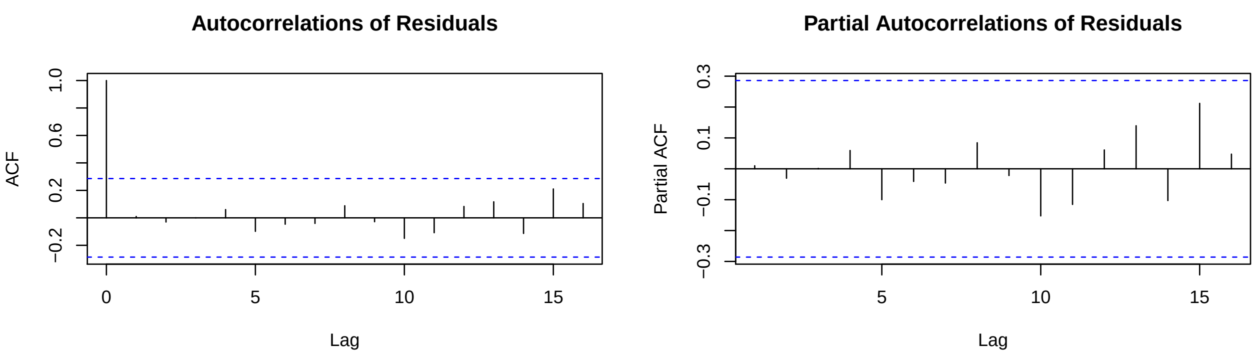
```
##
##
## Value of test-statistic is: -2.7421 6.8624
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau2 -3.51 -2.89 -2.58
## phi1  6.70  4.71  3.86
```

La valeur critique du test statistique au seuil de 5% (tau2 = 4.71) n'est pas à l'intérieur de l'intervalle de confiance, dans ce cas, on ne rejet pas l'hypothèse nulle. La série n'est pas stationnaire. La valeur du test statistique est de -2.74.

Modèle sans intercept et sans tendance

```r
adf2.lpib <- ur.df(lpib, type = "none", lags = 10, selectlags = "BIC")
plot(adf2.lpib)
```





Les deux graphique d'autocorrélation et d'autocorrélation partielle nous indiquent le modèle AR(1).

```r
summary(adf2.lpib)
```

```
##
## ###############################################
## # 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
## -0.213987 -0.063303  0.007163  0.053040  0.280564
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## z.lag.1    0.0014068  0.0006295   2.235   0.0305 *
## z.diff.lag 0.3399665  0.1405653   2.419   0.0197 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1046 on 45 degrees of freedom
## Multiple R-squared:  0.3177, Adjusted R-squared:  0.2874
## F-statistic: 10.48 on 2 and 45 DF,  p-value: 0.0001836
##
##
## Value of test-statistic is: 2.2346
##
## Critical values for test statistics:
##      1pct  5pct 10pct
## tau1 -2.6 -1.95 -1.61
```

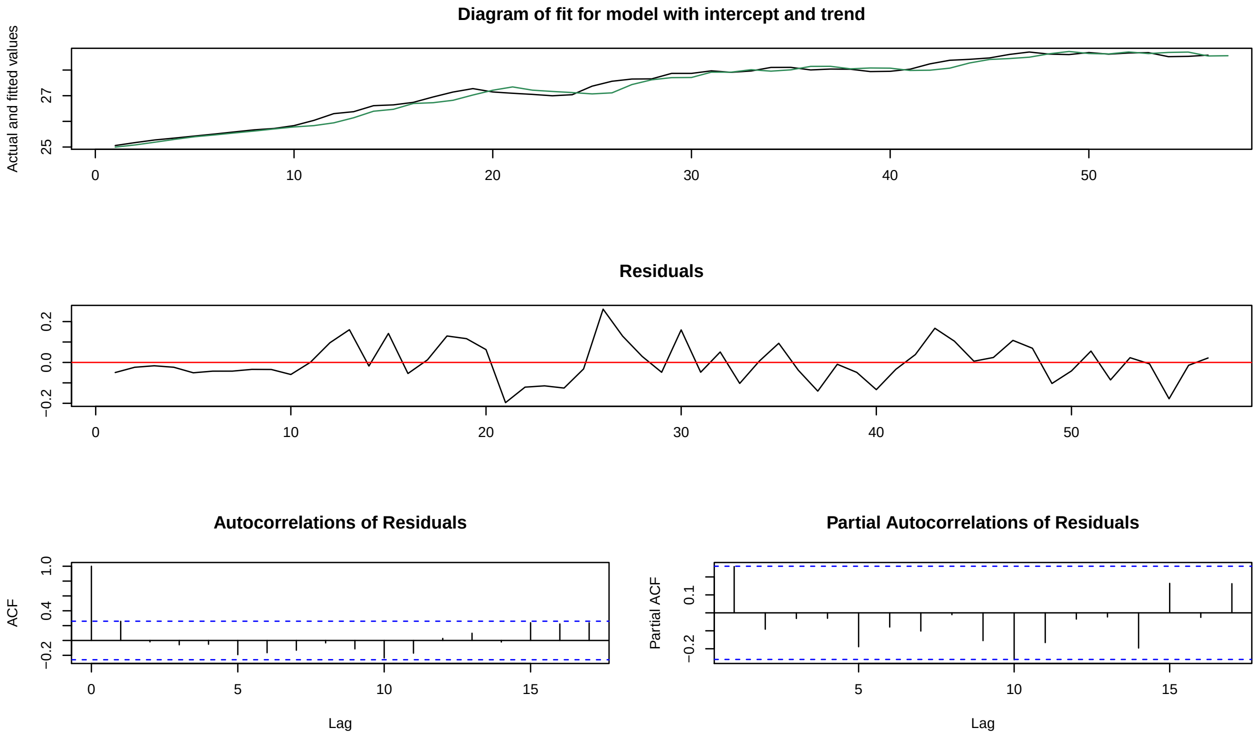
On ne rejète pas l'hypothèse nulle. Notre série n'est toujours pas stationnaire (-1.95 < 2.23), cette valeur est loin de l'intervalle de confiance.

Test de Phillips & Perron

Le test de Phillips-Perron tout comme celui de ADF, indique que l'hypothèse nulle est l'absence de stationnarité.

Modèle avec tendance

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



L'analyse graphique nous indique ARMA(1,1) pour le test de Phillips et Perron.

```r
summary(pp.lpib)
```

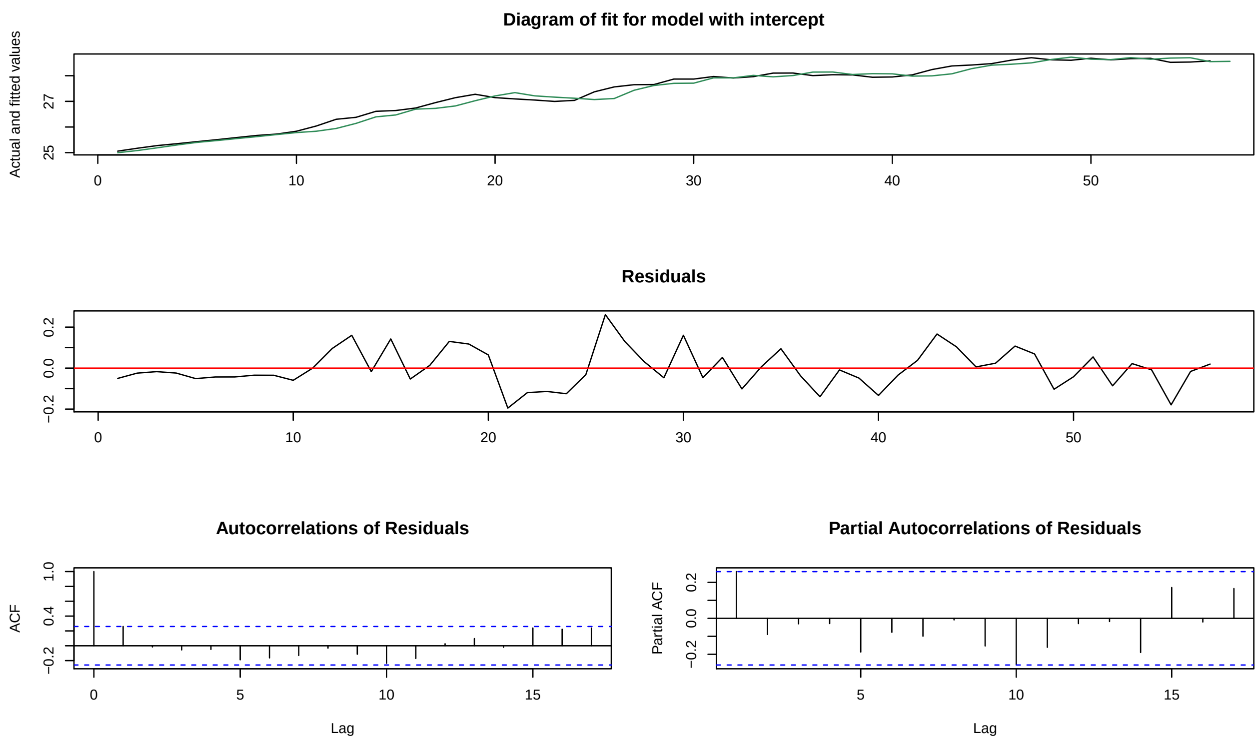
```
##
## ###################################
## # Phillips-Perron Unit Root Test #
## ###################################
##
## Test regression with intercept and trend
##
##
## Call:
## lm(formula = y ~ y.l1 + trend)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.19675 -0.04880 -0.01636  0.05561  0.26089
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.8095755  1.1196512   0.723    0.473
## y.l1         0.9726950  0.0411286  23.650   <2e-16 ***
## trend       -0.0002062  0.0029531  -0.070    0.945
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.09476 on 54 degrees of freedom
## Multiple R-squared:  0.9936, Adjusted R-squared:  0.9933
## F-statistic:  4167 on 2 and 54 DF,  p-value: < 2.2e-16
##
```

```
##
## Value of test-statistic, type: Z-tau  is: -0.9021
##
##            aux. Z statistics
## Z-tau-mu              0.2964
## Z-tau-beta            0.4930
##
## Critical values for Z statistics:
##                       1pct      5pct     10pct
## critical values -4.124945 -3.488947 -3.172654
```

Tout les coefficients du modèle ne sont pas statistiquement significatifs. Mais ici aussi comme les autres, on ne rejet pas l'hypothèse nulle, donc la série n'est pas stationnaire.

Modèle avec intercept et sans tendance

```r
pp1.lpib <- ur.pp(lpib, type = "Z-tau", model = "constant", lags = "short")
plot(pp1.lpib)
```



Sans forcement une bonne précision, au vu des graphique de l'autocorrélation et autocorrélation partielle on prend comme modèle ARMA(1,1).

```r
summary(pp1.lpib)
```

```
##
## ##################################
## # Phillips-Perron Unit Root Test #
## ##################################
##
## Test regression with intercept
##
##
## Call:
```

```
## lm(formula = y ~ y.l1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.19504 -0.04869 -0.01710  0.05481  0.26091
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.88509    0.28723   3.081  0.00321 **
## y.l1         0.96992    0.01053  92.126  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.09389 on 55 degrees of freedom
## Multiple R-squared:  0.9936, Adjusted R-squared:  0.9934
## F-statistic:  8487 on 1 and 55 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic, type: Z-tau  is: -2.553
##
##          aux. Z statistics
## Z-tau-mu            2.7464
##
## Critical values for Z statistics:
##                      1pct      5pct     10pct
## critical values -3.547748 -2.912708 -2.593707
```

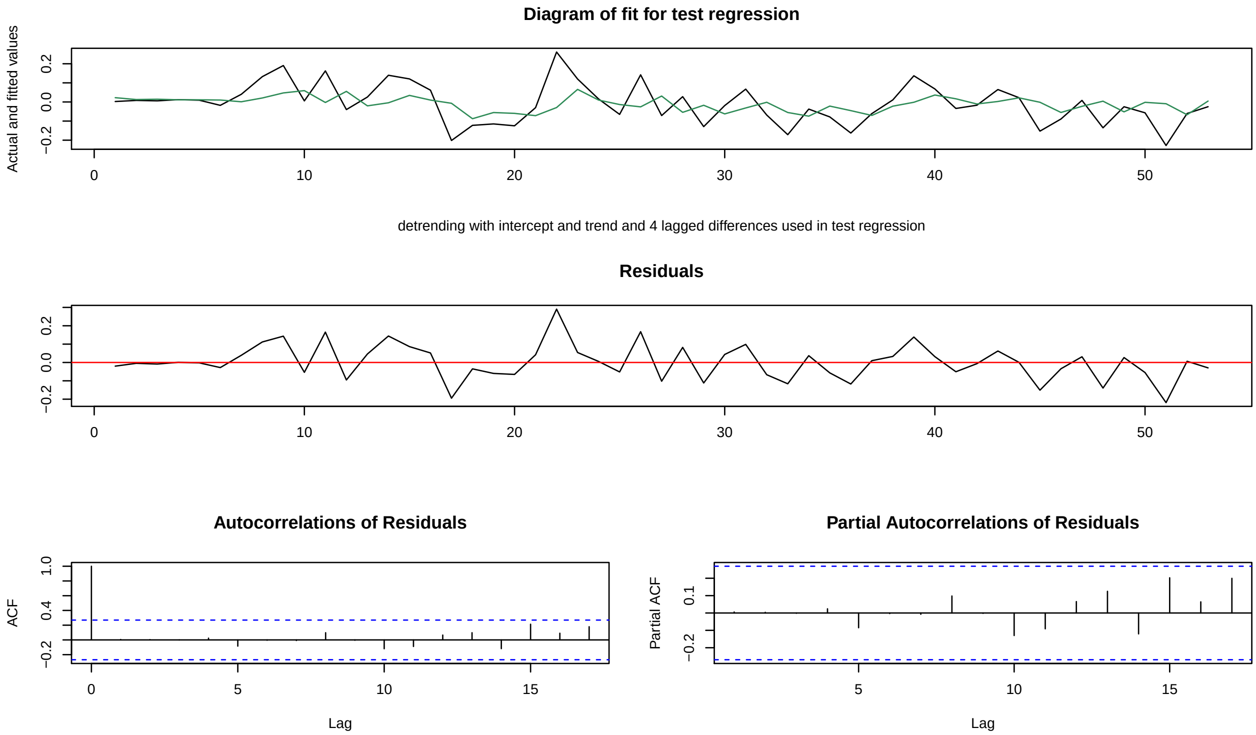
Le Z statistique est de 2.74, cette valeur à comparer à la valeur critique du test qui est de -2.91 est hors l'intervalle de confiance, on ne rejet donc pas l'hypothèse nulle.

Test de DF-GLS (ERS-Elliot, Rotenberg e Stock, 1996)

Ici aussi l'hypothèse nulle est que la série n'est pas stationnaire.

Modèle avec tendance

```
ers.lpib <- ur.ers(lpib, type = "DF-GLS", model = "trend", lag.max = 4)
plot(ers.lpib)
```



En appliquant le test de ERS, les deux graphique de corrélation et corrélation partielle nous indiquent que l'on choisi le medèle AR(1).

```
summary(ers.lpib)
```

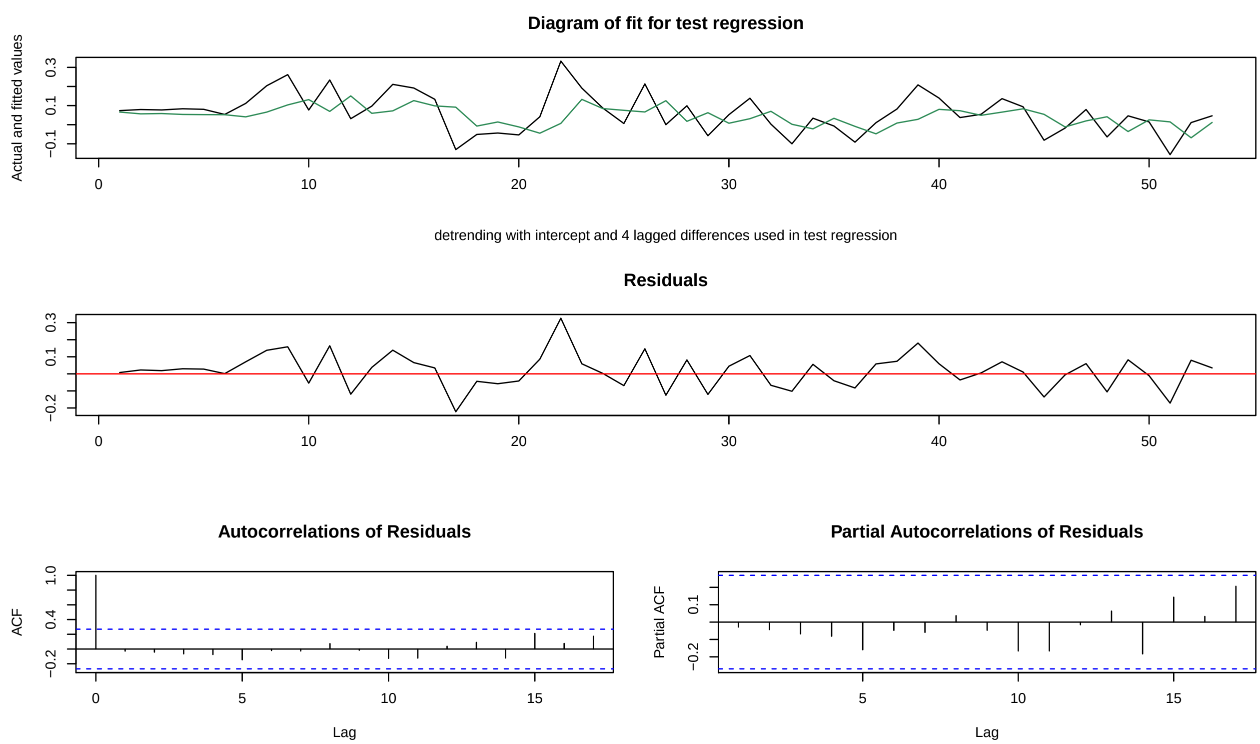
```
##
## ###############################################
## # Elliot, Rothenberg and Stock Unit Root Test #
## ###############################################
##
## Test of type DF-GLS
## detrending of series with intercept and trend
##
##
## Call:
## lm(formula = dfgls.form, data = data.dfgls)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.218915 -0.054924  0.000594  0.046217  0.290672
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## yd.lag       -0.05166    0.04582  -1.127    0.265
## yd.diff.lag1  0.37218    0.14422   2.581    0.013 *
## yd.diff.lag2  0.00075    0.15395   0.005    0.996
## yd.diff.lag3  0.03485    0.15868   0.220    0.827
## yd.diff.lag4  0.06669    0.15281   0.436    0.664
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.09977 on 48 degrees of freedom
## Multiple R-squared:  0.138,  Adjusted R-squared:  0.04817
## F-statistic: 1.536 on 5 and 48 DF,  p-value: 0.1963
##
##
## Value of test-statistic is: -1.1274
##
## Critical values of DF-GLS are:
##                  1pct  5pct 10pct
## critical values -3.58 -3.03 -2.74
```

Lorsqu'on compare la valeur du test qui est de -1.12 aux valeurs critique du modèle DF-GLS qui est de -3.03 au seuil de 5%, cette dernière est en dehors du seuil de confiance. Ainsi, l'hypothèse nulle ne peut être réjetée. Il y a présence de racine unitaire.

Modèle avec intercept

```r
ers1.pi = ur.ers(lpib, type = "DF-GLS", model = "constant", lag.max = 4)
plot(ers1.pi)
```



Le test graphique indique le modèle autoregréssif d'ordre 1, c'est à dire AR(1).

```r
summary(ers1.pi)
```

```
##
## ###############################################
## # 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
## -0.22208 -0.04375  0.02776  0.07054  0.32579
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## yd.lag       0.0009788  0.0097203   0.101  0.92021
## yd.diff.lag1 0.4239613  0.1436136   2.952  0.00487 **
## yd.diff.lag2 0.0169133  0.1558309   0.109  0.91402
## yd.diff.lag3 0.0847698  0.1614576   0.525  0.60198
## yd.diff.lag4 0.1348326  0.1484622   0.908  0.36831
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1048 on 48 degrees of freedom
## Multiple R-squared:  0.3031, Adjusted R-squared:  0.2305
## F-statistic: 4.176 on 5 and 48 DF,  p-value: 0.003135
##
##
## Value of test-statistic is: 0.1007
##
## Critical values of DF-GLS are:
##                  1pct  5pct 10pct
## critical values -2.6 -1.95 -1.62
```

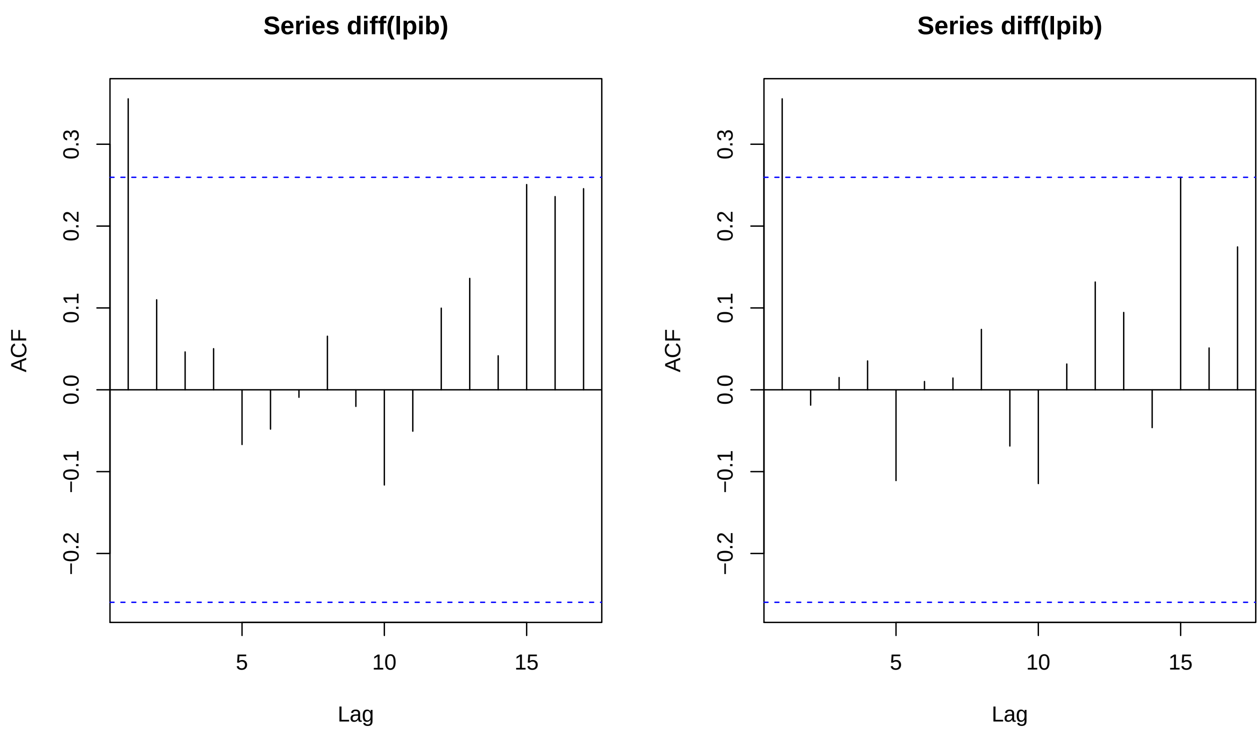
On ne rejet pas l'hypothèse nulle. Il y a présence d'une racine unitaire. Tous les tests de stationnarité et de racine unitaire ont été observés dans notre analyse mais le modèle indique l'absence de stationnarité. On retient tout simplement que la série GDP-France n'est pas stationnaire.

2. Estimation du modèle

Étant donné que la série GDP-France est non satationnaire, nous devons appliquer la première différence dans l'estimation du modèle considéré. Pour se faire :

```r
lpib <- log(lpib)
plot(diff(lpib)); abline(h=0, col = "grey")
grid()
```


```
par(mfrow = c(1, 2))
acf(diff(lpib), drop.lag.0 = TRUE, type = "correlation")
acf(diff(lpib), type = "partial")
```



De manière dont les graphiques se présente, il nous est difficile de déterminer s'il s'agit de quel modèle faut adopter. Mais aussi, les différents tests nous ont donnés différents modèles. Ainsi, nous estimons plusieurs modèles et au final nous choisirons un pour la prévision. Le critère de AIC et BIC va déterminer le modèle à prendre.

```
modelo1 = Arima(lpib, order = c(2, 1, 2))
modelo2 = Arima(lpib, order = c(1, 1, 1))
```

```r
modelo3 = Arima(lpib, order = c(0, 2, 2))

auto.arima(lpib, ic = "aic")
```

```
## Series: lpib
## ARIMA(1,2,1)
##
## Coefficients:
##          ar1      ma1
##       0.3038  -0.9322
## s.e.  0.1403   0.0549
##
## sigma^2 estimated as 1.276e-05:  log likelihood=237.21
## AIC=-468.43   AICc=-467.97   BIC=-462.35
```

```r
auto.arima(lpib, ic = "bic")
```

```
## Series: lpib
## ARIMA(1,2,1)
##
## Coefficients:
##          ar1      ma1
##       0.3038  -0.9322
## s.e.  0.1403   0.0549
##
## sigma^2 estimated as 1.276e-05:  log likelihood=237.21
## AIC=-468.43   AICc=-467.97   BIC=-462.35
```

3. Vérification des modèles

Nous allons vérifier, à l'aide de tests statistiques, la significance des coefficients de modèles (test de robusteste des coefficients).

```r
coeftest(modelo1)
```

```
##
## z test of coefficients:
##
##     Estimate Std. Error z value Pr(>|z|)
## ar1  1.14843    0.42356  2.7114   0.0067 **
## ar2 -0.15614    0.41795 -0.3736   0.7087
## ma1 -0.76927    0.41906 -1.8357   0.0664 .
## ma2 -0.14246    0.36548 -0.3898   0.6967
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Les coefficients du modèle ARIMA(2, 1, 2) ne sont pas significatifs, à l'exception de AR(1).

```r
coeftest(modelo2)
```

```
##
## z test of coefficients:
##
##     Estimate Std. Error z value  Pr(>|z|)
## ar1  0.84381    0.19173  4.4010 1.078e-05 ***
```

```
## ma1 -0.47790    0.37388 -1.2782    0.2012
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Les coefficients du modèle ARIMA(1,1,1) montre que AR(1) est statistiquement significatif.

```
coeftest(modelo3)
```

```
##
## z test of coefficients:
##
##     Estimate Std. Error z value  Pr(>|z|)
## ma1 -0.63145    0.12157 -5.1942 2.056e-07 ***
## ma2 -0.26516    0.11663 -2.2736   0.02299 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Les coefficients du modèle ARIMA (0, 2, 2) sont statistiquement significatifs.

Pour faire le choix d'un modèle, nous appelons le critère AIC et BIC. Nous prendrons la valeur la plus petite. On peut voir aussi l'ecart type de chaque modèle.

```
AIC(modelo1, modelo2, modelo3)
```

```
## Warning in AIC.default(modelo1, modelo2, modelo3): models are not all
## fitted to the same number of observations
```

```
##         df       AIC
## modelo1  5 -473.7149
## modelo2  3 -473.4344
## modelo3  3 -468.4123
```

```
BIC(modelo1, modelo2, modelo3)
```

```
## Warning in BIC.default(modelo1, modelo2, modelo3): models are not all
## fitted to the same number of observations
```

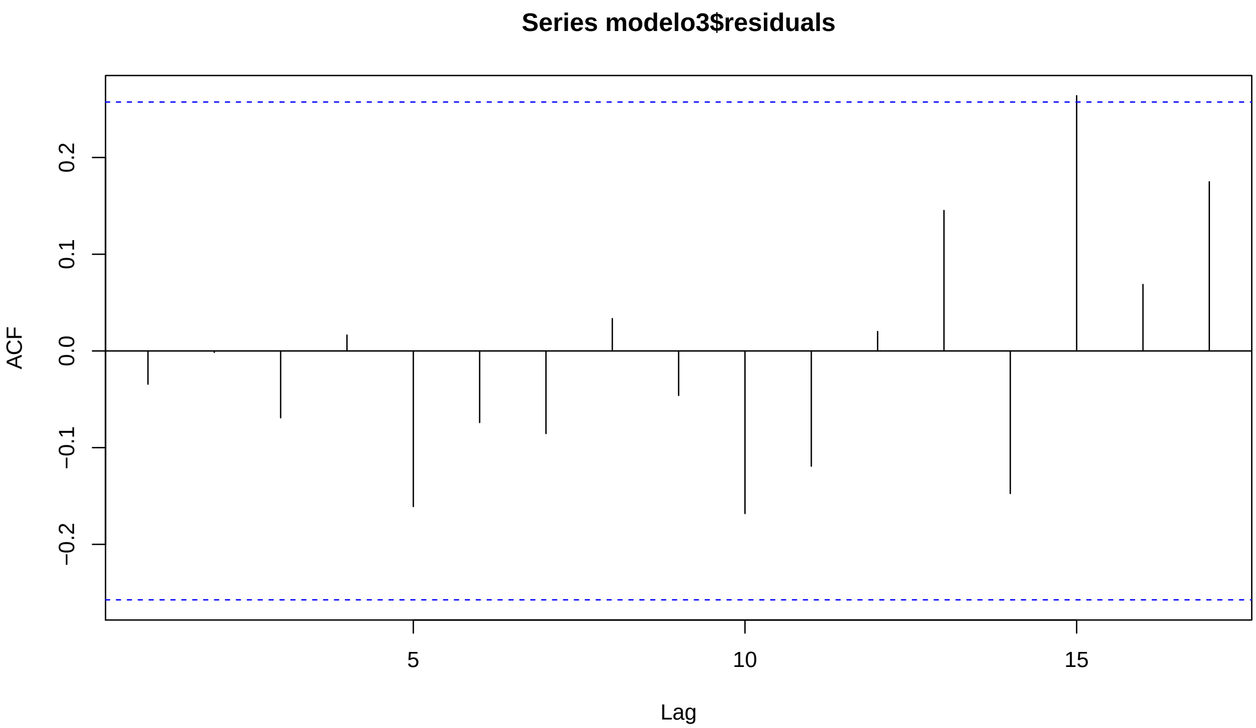
```
##         df       BIC
## modelo1  5 -463.4996
## modelo2  3 -467.3053
## modelo3  3 -462.3363
```

Bien que à différence peu, nous optons pour le modèle ARIMA (0, 2, 2) qui indique la valeur inférieur du critère AIC et BIC.

Nous allons à présent voir le comportement des résidus de notre modèle choisi.

1. Test d'autocorrelation résiduelle

```
acf(modelo3$residuals, drop.lag.0 = TRUE)
```



Le test informel nous indique que les résidus sont normalement distribués.

2. 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:

3. Test de Box & Pierce (1970)

```r
Box.test(modelo3$residuals, lag = 10, type = "Box-Pierce", fitdf = 1)
```

```
##
##  Box-Pierce test
##
## data:  modelo3$residuals
## X-squared = 4.4317, df = 9, p-value = 0.8808
```

Les coefficients d'autocorrelation résiduelles sont statistiquement égales à zéro. On rejet donc l'hypothèse nulle. Le p-value est significatif.

4. Test de Ljung & Box (1978)

```r
Box.test(modelo3$residuals, lag = 10, type = "Ljung-Box", fitdf = 1)
```

```
##
##  Box-Ljung test
##
## data:  modelo3$residuals
## X-squared = 5.2347, df = 9, p-value = 0.8134
```

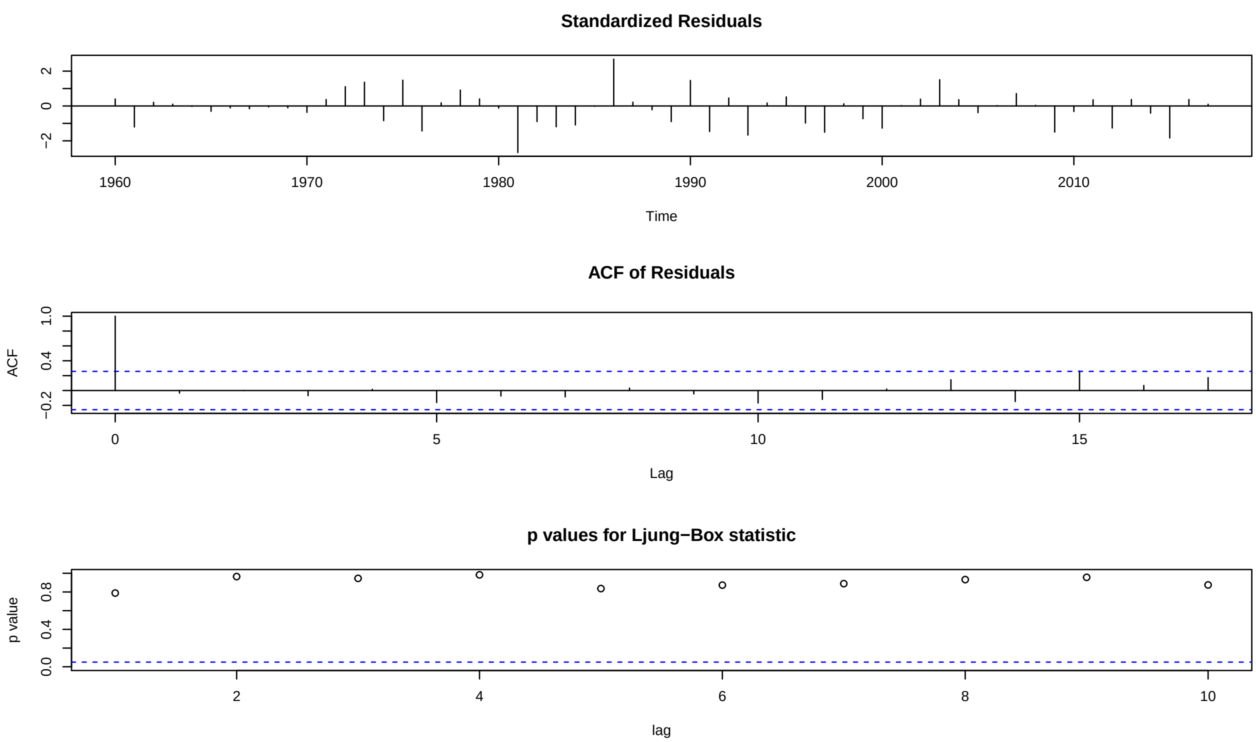
P-value est supérieur au seuil de 5%, ce qui revient à rejeter l'hypothèse nulle.

```
for (i in 1:10) {
  b = Box.test(modelo3$residuals, i, type = "Ljung-Box")$p.value
  print(b)
}
```

```
## [1] 0.7882568
## [1] 0.9645131
## [1] 0.945424
## [1] 0.9831225
## [1] 0.8362852
## [1] 0.8730921
## [1] 0.8889786
## [1] 0.9321662
## [1] 0.9565092
## [1] 0.8749603
```

Tous les p-values du test sont statisquement significatifs c.à.d supérieur au seuil de 5%. L'hypothese nulle est donc rejeter. On peut aussi toute fois voir ce comportement avec le graphique de Ljung & Box :

```
tsdiag(modelo3, gof.lag = 10)
```



Le graphique en bas indique que les p-values sont hauts, alors on rejet l'hypothèse nulle a ce point.

5. Test de normalité des résidues

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

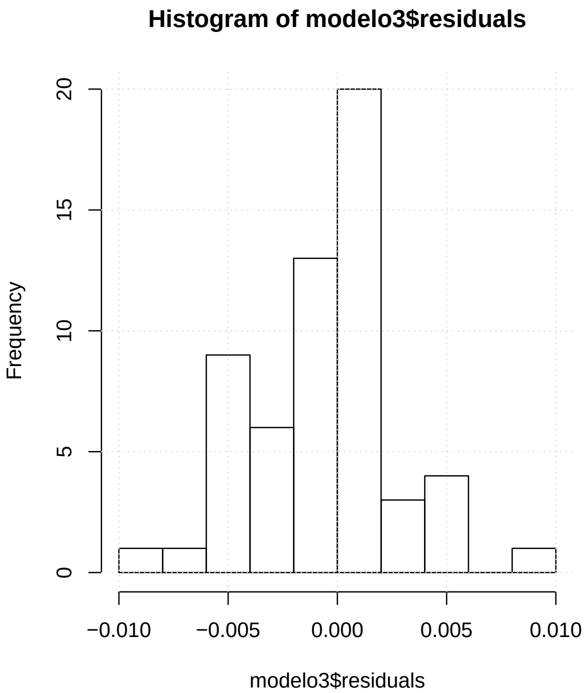
1. *Test de Jarque-Bera*: Ce test est logiquement utilisé pour les grands échantillons.

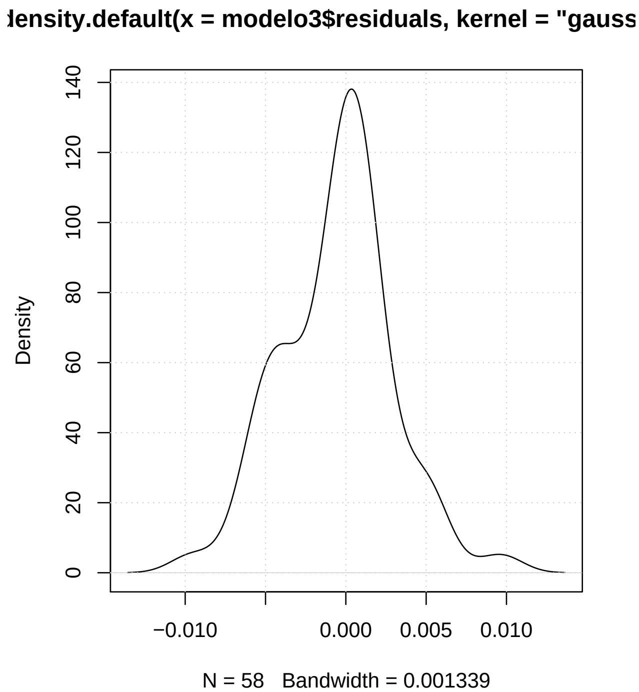
Le test de Jarque-Bera (JB) analyse si les moments de la série estimée (dans ce cas, les résidus) sont les mêmes que la normale. Dans cette hypothèse, l'asymétrie est égale à zéro et le kurtosis est égal à 3.

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(modelo3$residuals)
grid()
plot(density(modelo3$residuals, kernel = "gaussian"))
grid()
```





L'nanlyse graphique nous fait remarquer que les erreurs sont normalement distribuées.

1. Test de Jarque-Bera

```
jarque.bera.test(modelo3$residuals)
```

```
##
##  Jarque Bera Test
##
## data:  modelo3$residuals
## X-squared = 0.85002, df = 2, p-value = 0.6538
```

On rejet l'hypothèse nulle qui dit que les résidus ne sont pas normalement distribués.

2. Test de Shapiro-Wilk

```
shapiro.test(modelo3$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  modelo3$residuals
## W = 0.97241, p-value = 0.2079
```

Au seuil de 5%, on rejet l'hypothèse nulle. Les résidus suivent une loi normale.

6. Test d'hétéroscédasticité (ARCH-LM)

Lorsque les variances des résidus des variables examinées sont différentes. L'hypthèse nulle est la présence homoscédasticité.

```
ArchTest(modelo3$residuals, lags = 10)
```

```
##
##  ARCH LM-test; Null hypothesis: no ARCH effects
##
## data:  modelo3$residuals
## Chi-squared = 12.229, df = 10, p-value = 0.27
```

Au seuil de 5%, on ne rejet pas l'hypothèse nulle. On est en présence d'homoscédasticité des erreurs. C'est à dire que la variance est constante pour toute les variables.

4. Prévision du modèle ARIMA (0, 2, 2)

Avant de faire la prévision, nous allons transformer notre 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).

```
lambda <- BoxCox.lambda(lpib)
print(lambda)
```

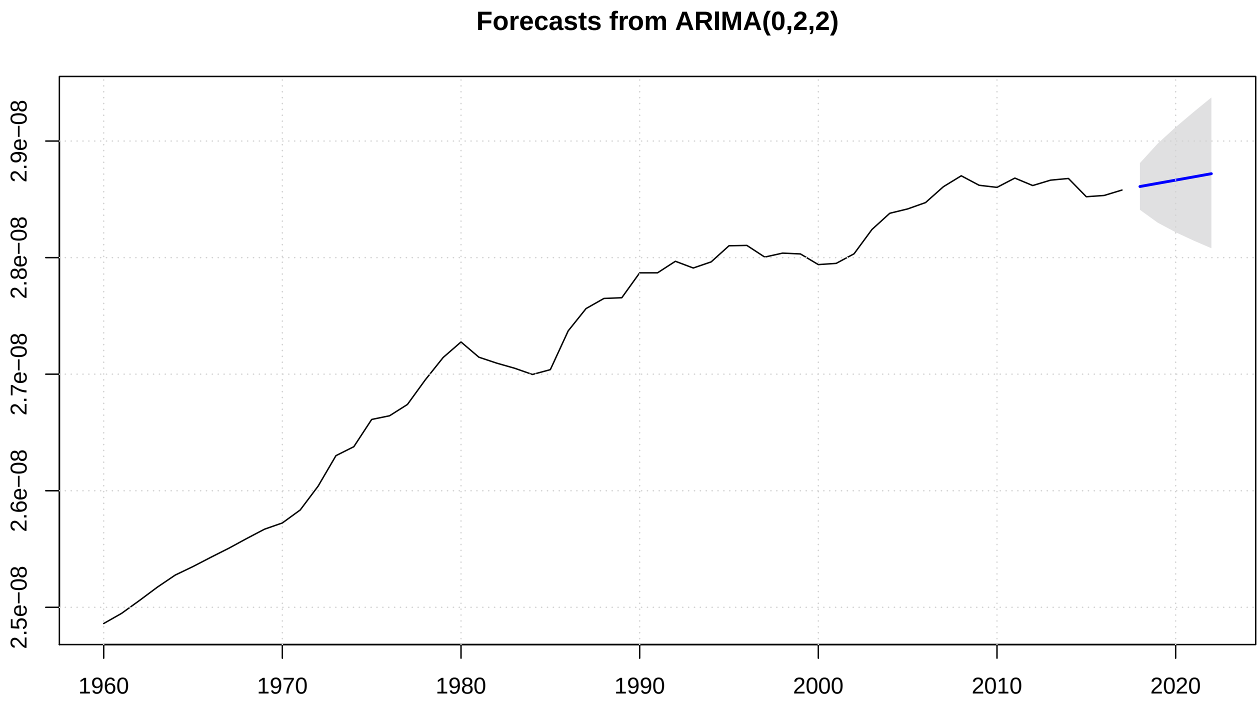
```
## [1] 1.999924
```

En réalité, nous n'allons pas appliquer puisque la série a été transformé en log (voir ci-dessus).

```
prevision <- forecast(modelo3, h =5, level = c(0.90, 0,95),
                      biasadj = FALSE)
print(prevision)
```

```
##      Point Forecast     Lo 0     Hi 0   Lo 0.9   Hi 0.9    Lo 95    Hi 95
## 2018       3.353751 3.353751 3.353751 3.353711 3.353792 3.346751 3.360752
## 2019       3.354710 3.354710 3.354710 3.354642 3.354779 3.342844 3.366576
## 2020       3.355669 3.355669 3.355669 3.355579 3.355760 3.339953 3.371385
## 2021       3.356628 3.356628 3.356628 3.356517 3.356738 3.337429 3.375827
## 2022       3.357587 3.357587 3.357587 3.357457 3.357716 3.335076 3.380097
```

```
sca <- 1e9
prevision$mean <- exp(prevision$mean) / sca
prevision$upper <- exp(prevision$upper) / sca
prevision$lower <- exp(prevision$lower) / sca
prevision$x  <- exp(prevision$x) / sca

plot(prevision)
grid()
```


Forecasts from ARIMA(0,2,2)
2.9e−08
2.8e−08
2.7e−08
2.6e−08
2.5e−08
1960
1970
1980
1990
2000
2010
2020