03-exercices-ARIMA

Romain CAPLIEZ

Exercice 1 : Modéliser le PIB américain

Données : `GDP` à partir de la `FRED`

Question 1

Obtenir les données du PIB américain fournies par la FRED. Quelles sont les dates de début et de fin de la série ? Combien d'observation comporte-t-elle ?

Question 2

Représenter graphiquement la série et la décrire. Cette série semble-t-elle stationnaire ou non ?

Question 3

Stationnariser si besoin la série. La représenter, la décrire. Est-elle stationnaire ?

Question 4

Etudier la normalité ainsi que les fonctions d'auto-corrélation et auto-corrélation partielle de la série stationnaire. Quelle est la moyenne de la série ? Quel est sa variance et son écart-type ?

Question 5

Modéliser la série avec un AR(1) et étudier les propriétés des résidus. Que peut-on en déduire ?

Question 6

Modéliser la série avec un AR(8) et étudier les propriétés des résidus. Que peut-on en déduire ?

Question 7

Quel est le meilleur modèle selon le critère BIC en in-sample ? (Important : imposer la recherche aux modèles stationnaires. Etudier les propriétés des résidus. Que peut-on en déduire ?

Question 8

Représenter les valeurs prédites et les comparer avec les vraies valeurs de la série. Le modèle semble-t-il correctement modéliser la série ?

Question 9

Prédire les valeurs des 5 prochains trimestres et commenter.

Exercice 2 : Prédire la population de Lynx aux USA

Données : `lynx` disponibles de base dans R.

Vous devez prédire la population de lynx aux Etats-Unis sur les 5 prochaines années. Proposez au minimum trois modèles concurrents. Quel est le modèle le plus adapté à la prévision ? Semble-t-il être plus fiable que de l'aléatoire ?

Finissez par utiliser la fonction `forecast::dm.test()` *afin de tester si la prédiction de votre meilleur modèle est significativement meilleure que l'aléatoire. Qu'en est-il par rapport aux autres modèles ?*

Exercices 3 : Les importations de barium

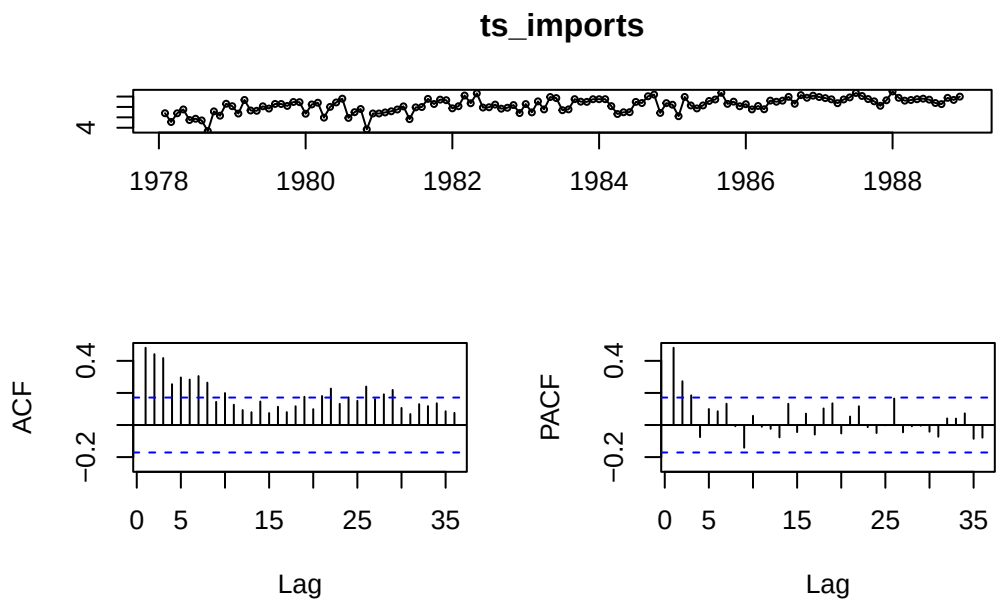
Vous travaillez comme économètre pour une organisation de commerce international. On vous remet le script R suivant qui analyse l'évolution des importations d'un produit chimique. Commentez les étapes de ce code, expliquez les résultats, justifier les différentes étapes. Quelles critiques pourriez-vous faire de cette analyse dont le but serait ultimement de prédire les futures importations chinoises de barium.

```
source(here::here("02-codes", "utils", "setup.R"))
```

```
ts_imports <-
  wooldridge::barium |>
  mutate(chnimp = log (chnimp)) |>
  select(chnimp) |>
  ts(start = c(1978, 2), frequency = 12) |>
  print()
```

```
          Jan      Feb      Mar      Apr      May      Jun      Jul      Aug
1978          5.395725 4.551748 5.390703 5.760231 4.741788 4.863866 4.701316
1979 6.072988 5.364973 6.670315 5.659620 5.631049 6.063546 5.843202 6.289558
1980 5.344423 6.239479 6.405794 4.971451 6.005210 6.431504 6.795391 4.943284
1981 5.369887 5.473146 5.582197 5.753371 6.055825 4.826600 5.960745 5.992814
1982 5.866108 6.068273 7.099575 6.355964 7.307619 5.940939 5.970232 6.225572
1983 6.277384 5.477051 6.535899 5.762089 6.954673 6.851957 5.688236 5.766626
1984 6.745087 6.748678 6.077268 5.321582 5.487149 5.497734 6.486773 6.376893
1985 6.214395 5.108040 6.964109 6.143805 5.856229 6.145952 6.575141 6.717512
1986 6.263136 5.761265 6.093162 5.776514 6.609456 6.500337 6.602950 6.970928
1987 6.953912 6.850960 6.733437 6.361160 6.719157 6.917762 7.343739 7.056311
1988 7.492491 6.880180 6.604633 6.667186 6.744136 6.797293 6.702404 6.379142
          Sep      Oct      Nov      Dec
1978 3.680923 5.571481 5.159044 6.307999
1979 6.283608 6.098813 6.475826 6.460429
1980 5.495020 5.800834 3.841300 5.371549
1981 6.776318 6.284121 6.704360 6.648742
1982 5.831947 5.920442 6.180760 5.411856
1983 6.751262 6.509971 6.489101 6.742833
1984 7.028595 7.192157 5.425230 6.368635
1985 7.359936 6.297095 6.508461 6.064426
1986 6.305049 7.137602 6.883992 7.083961
1987 6.760044 6.540015 6.112318 6.641706
1988 6.272374 6.875271 6.677159 6.991918
```

```
forecast::tsdisplay(ts_imports)
```




```
urca::ur.df(ts_imports, "trend", lags = 20, selectlags = "AIC") |> summary()
```

```
###############################################
# 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
-1.9849 -0.3173  0.0382  0.3333  1.0870

Coefficients:
             Estimate Std. Error t value     Pr(>|t|)
(Intercept)  4.023465   0.694564   5.793 0.0000000717 ***
```

```
z.lag.1     -0.724349   0.122276  -5.924 0.0000000395 ***
tt           0.007166   0.001984   3.612     0.000466 ***
z.diff.lag  -0.107121   0.096167  -1.114     0.267838
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.5377 on 106 degrees of freedom
Multiple R-squared:  0.4129,    Adjusted R-squared:  0.3963
F-statistic: 24.85 on 3 and 106 DF,  p-value: 0.000000000002979


Value of test-statistic is: -5.9239 11.7243 17.5683

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
urca::ur.df(ts_imports, "drift", lags = 20, selectlags = "AIC") |>
  summary()
```

```
###############################################
# 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
-1.95449 -0.27001  0.05389  0.29209  1.09007

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.42951    0.71780   3.385  0.00100 **
z.lag.1     -0.38462    0.11388  -3.377  0.00102 **
z.diff.lag1 -0.35477    0.11586  -3.062  0.00279 **
z.diff.lag2 -0.17203    0.09636  -1.785  0.07708 .
```

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.5615 on 106 degrees of freedom
Multiple R-squared:  0.3598,    Adjusted R-squared:  0.3417
F-statistic: 19.86 on 3 and 106 DF,  p-value: 0.0000000002727


Value of test-statistic is: -3.3775 5.7285

Critical values for test statistics:
      1pct  5pct 10pct
tau2 -3.46 -2.88 -2.57
phi1  6.52  4.63  3.81
```

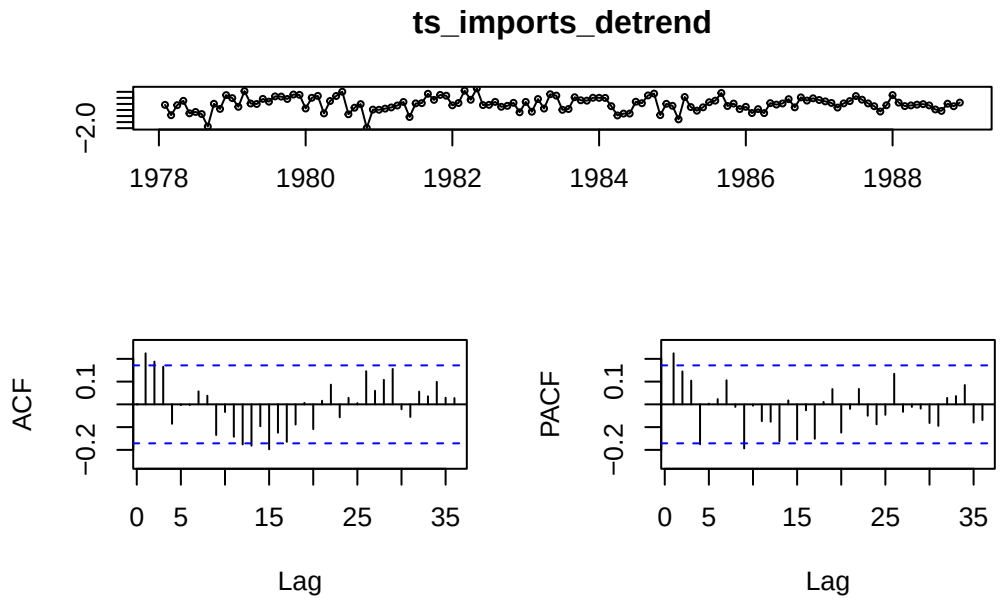
```r
urca::ur.df(ts_imports, "none", lags = 20, selectlags = "AIC") |>
  summary()
```

```
###############################################
# 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
-1.45179 -0.36769  0.04364  0.39448  1.20287

Coefficients:
             Estimate Std. Error t value        Pr(>|t|)
z.lag.1      0.002386   0.008545   0.279         0.78059
z.diff.lag1 -0.713802   0.096046  -7.432 0.0000000000326 ***
z.diff.lag2 -0.482855   0.113435  -4.257 0.0000458865452 ***
z.diff.lag3 -0.275501   0.117436  -2.346         0.02089 *
z.diff.lag4 -0.371227   0.117388  -3.162         0.00206 **
z.diff.lag5 -0.364895   0.113229  -3.223         0.00170 **
z.diff.lag6 -0.248136   0.095712  -2.593         0.01091 *
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.5627 on 103 degrees of freedom
Multiple R-squared:  0.3753,    Adjusted R-squared:  0.3329
F-statistic: 8.841 on 7 and 103 DF,  p-value: 0.0000000174


Value of test-statistic is: 0.2793

Critical values for test statistics:
      1pct  5pct 10pct
tau1 -2.58 -1.95 -1.62
```

```
ts_imports_detrend <- dynlm::dynlm(ts_imports ~
trend(ts_imports))[["residuals"]]
```

```
forecast::tsdisplay(ts_imports_detrend)
```




```
urca::ur.df(ts_imports_detrend, "trend", lags = 20, selectlags = "AIC") |>
  summary()
```

```
###############################################
# 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
-1.9849 -0.3173  0.0382  0.3333  1.0870

Coefficients:
              Estimate Std. Error t value     Pr(>|t|)
(Intercept)  0.0563332  0.1330953   0.423        0.673
z.lag.1     -0.7243490  0.1222758  -5.924 0.0000000395 ***
tt          -0.0006735  0.0016238  -0.415        0.679
z.diff.lag  -0.1071213  0.0961670  -1.114        0.268
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.5377 on 106 degrees of freedom
Multiple R-squared:  0.4129,    Adjusted R-squared:  0.3963
F-statistic: 24.85 on 3 and 106 DF,  p-value: 0.000000000002979


Value of test-statistic is: -5.9239 11.7169 17.5683

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
```

```
urca::ur.df(ts_imports_detrend, "drift", lags = 20, selectlags = "AIC") |>
  summary()
```

```
###############################################
# Augmented Dickey-Fuller Test Unit Root Test #
```

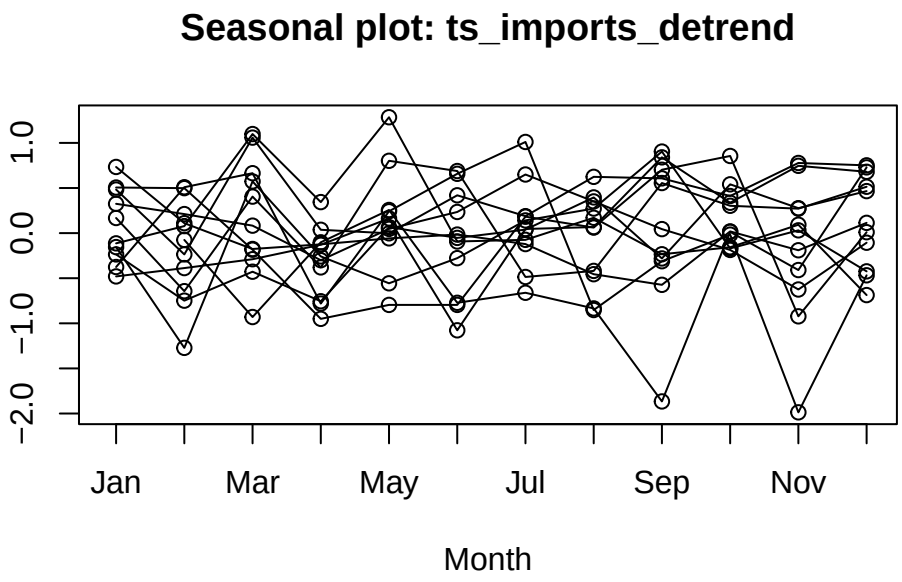
```
###############################################

Test regression drift


Call:
lm(formula = z.diff ~ z.lag.1 + 1 + z.diff.lag)

Residuals:
     Min       1Q   Median       3Q      Max
-1.95523 -0.30866  0.01618  0.33668  1.09955

Coefficients:
             Estimate Std. Error t value     Pr(>|t|)
(Intercept)  0.005395   0.051113   0.106        0.916
z.lag.1     -0.719012   0.121126  -5.936 0.0000000366 ***
z.diff.lag  -0.110047   0.095536  -1.152        0.252
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.5357 on 107 degrees of freedom
Multiple R-squared:  0.4119,    Adjusted R-squared:  0.4009
F-statistic: 37.47 on 2 and 107 DF,  p-value: 0.0000000000004623


Value of test-statistic is: -5.9361 17.6257

Critical values for test statistics:
      1pct  5pct 10pct
tau2 -3.46 -2.88 -2.57
phi1  6.52  4.63  3.81
```

```
urca::ur.df(ts_imports_detrend, "none", lags = 20, selectlags = "AIC") |>
  summary()
```

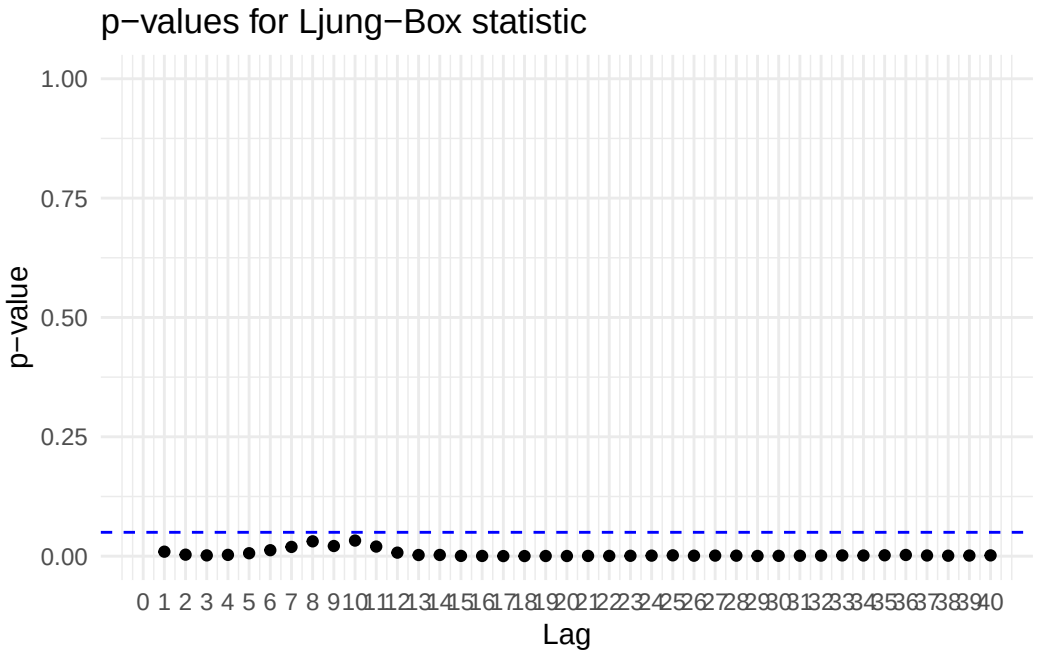
```
###############################################
# Augmented Dickey-Fuller Test Unit Root Test #
###############################################

Test regression none
```

```
Call:
lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)

Residuals:
     Min       1Q   Median       3Q      Max
-1.94973 -0.30357  0.02156  0.34186  1.10453

Coefficients:
           Estimate Std. Error t value     Pr(>|t|)
z.lag.1    -0.71853    0.12048  -5.964 0.0000000316 ***
z.diff.lag -0.11037    0.09505  -1.161        0.248
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.5332 on 108 degrees of freedom
Multiple R-squared:  0.4119,    Adjusted R-squared:  0.401
F-statistic: 37.82 on 2 and 108 DF,  p-value: 0.0000000000003562


Value of test-statistic is: -5.9637

Critical values for test statistics:
      1pct  5pct 10pct
tau1 -2.58 -1.95 -1.62
```

```r
seasonplot(ts_imports_detrend)
```




```
LSTS::Box.Ljung.Test(ts_imports_detrend, lag = 40)
```



```r
gen_arima <- forecast::auto.arima(ts_imports_detrend)

gen_arima |>
  summary()
```

```
Series: ts_imports_detrend
ARIMA(2,0,0)(1,0,0)[12] with zero mean

Coefficients:
         ar1     ar2     sar1
      0.1569  0.1513  -0.1470
s.e.  0.0885  0.0864   0.0936

sigma^2 = 0.2966:  log likelihood = -104.94
AIC=217.87   AICc=218.19   BIC=229.37

Training set error measures:
                      ME      RMSE       MAE      MPE     MAPE      MASE
Training set 0.0006727159 0.5383729 0.4175624 143.7747 192.7152 0.6061709
                     ACF1
Training set -0.009532465
```

```r
moments::agostino.test(gen_arima$residuals)
```

```
    D'Agostino skewness test

data:  gen_arima$residuals
skew = -0.51835, z = -2.41834, p-value = 0.01559
alternative hypothesis: data have a skewness
```

```r
moments::anscombe.test(gen_arima$residuals)
```

```
    Anscombe-Glynn kurtosis test

data:  gen_arima$residuals
kurt = 3.6305, z = 1.5326, p-value = 0.1254
alternative hypothesis: kurtosis is not equal to 3
```

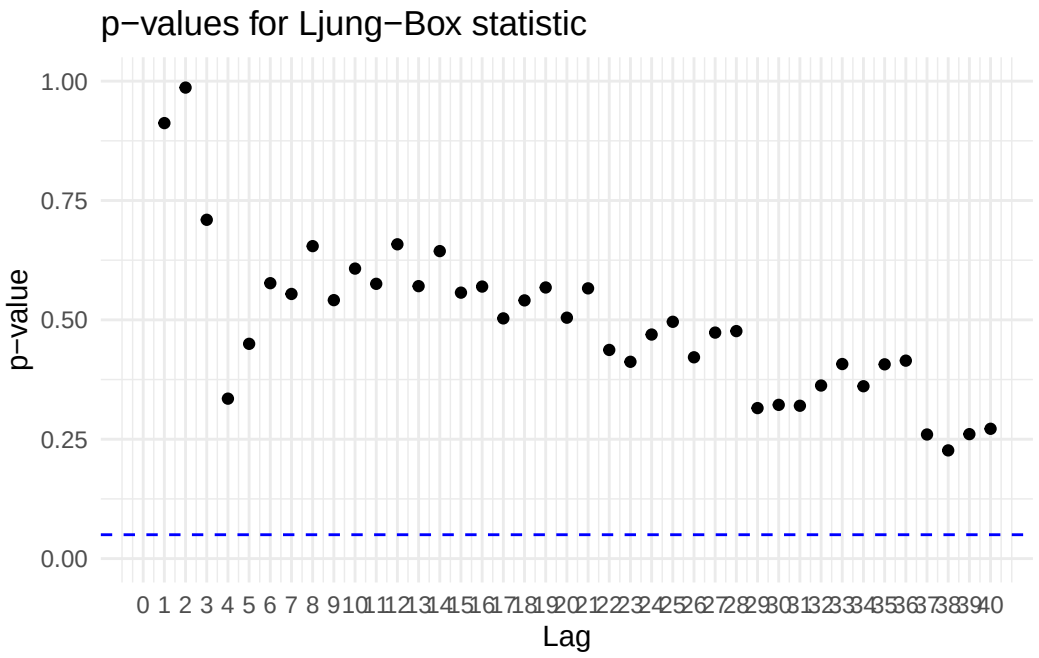
```
moments::jarque.test(as.numeric(gen_arima$residuals))
```

```
    Jarque-Bera Normality Test

data:  as.numeric(gen_arima$residuals)
JB = 8.0365, p-value = 0.01798
alternative hypothesis: greater
```

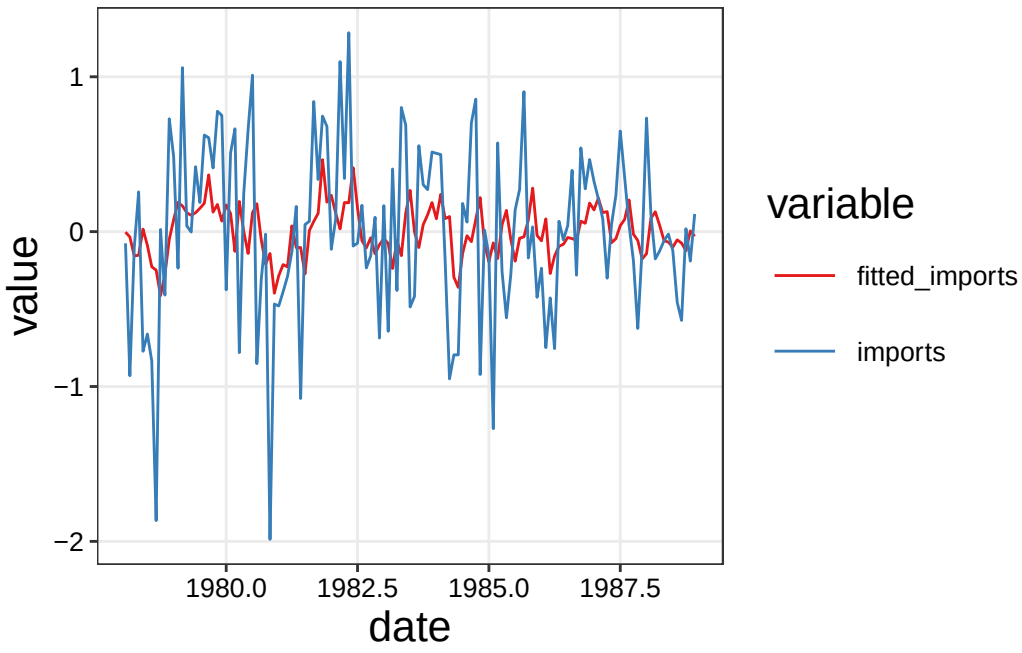
```
stats::shapiro.test(gen_arima$residuals)
```

```
    Shapiro-Wilk normality test

data:  gen_arima$residuals
W = 0.97938, p-value = 0.04368
```

```
LSTS::Box.Ljung.Test(gen_arima$residuals, lag = 40)
```



```
tibble(
  imports = gen_arima$x,
  fitted_imports = gen_arima$fitted,
  date = time(gen_arima$x)
) |>
  pivot_longer(
    cols = !date,
    names_to = "variable",
    values_to = "value"
  ) |>
  ggplot(aes(x = date, y = value, color = variable)) +
  geom_line() +
  scale_color_brewer(palette = "Set1")
```

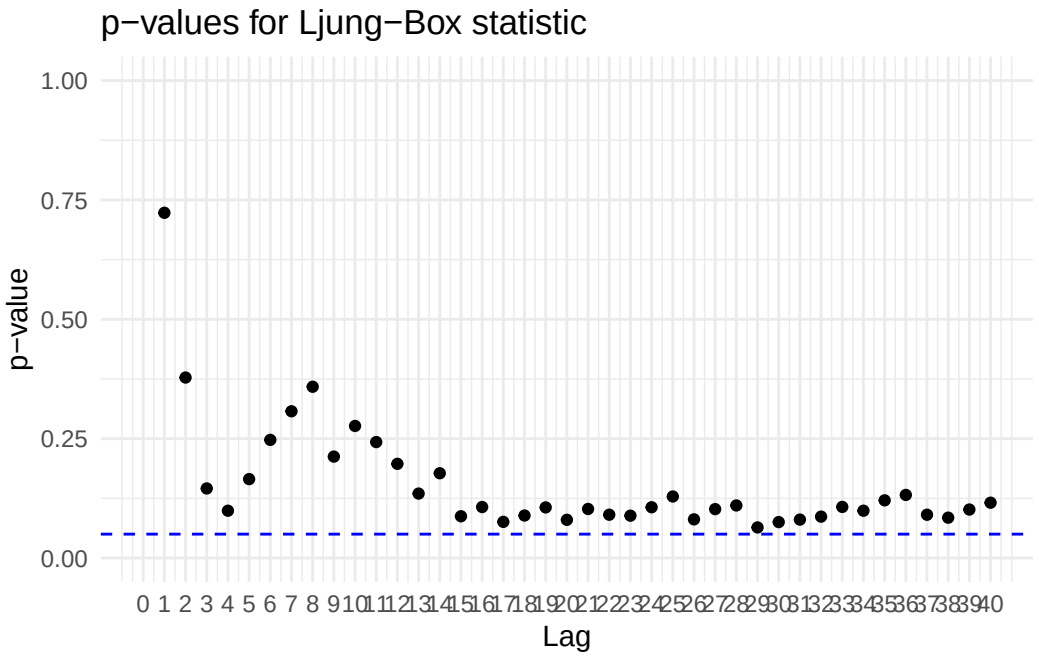



```
arma <- Arima(ts_imports_detrend, order = c(1,0,0))

arma |>
  summary()
```

```
Series: ts_imports_detrend
ARIMA(1,0,0) with non-zero mean
```

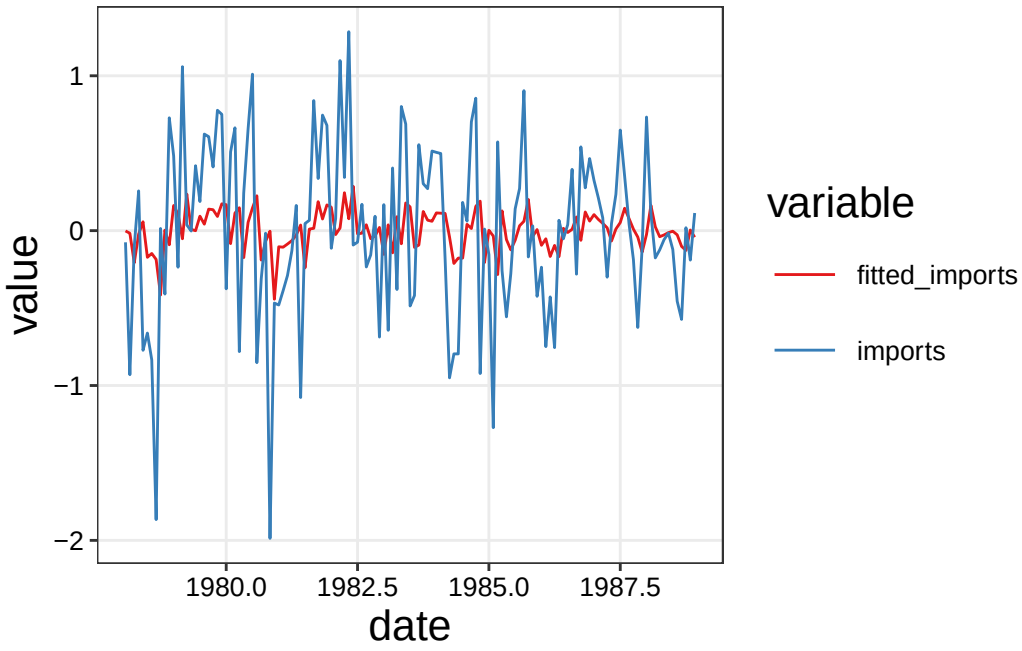
```
Coefficients:
         ar1    mean
      0.2228  0.0001
s.e.  0.0848  0.0617

sigma^2 = 0.307:  log likelihood = -107.56
AIC=221.12   AICc=221.31   BIC=229.74

Training set error measures:
                      ME      RMSE       MAE      MPE     MAPE      MASE
Training set 0.0001441666 0.5498681 0.4262859 140.6703 157.6763 0.6188347
                    ACF1
Training set -0.0306101
```

```
LSTS::Box.Ljung.Test(arma$residuals, lag = 40)
```




```
tibble(
  imports = arma$x,
  fitted_imports = arma$fitted,
  date = time(arma$x)
) |>
  pivot_longer(
```

```
    cols = !date,
    names_to = "variable",
    values_to = "value"
  ) |>
  ggplot(aes(x = date, y = value, color = variable)) +
  geom_line() +
  scale_color_brewer(palette = "Set1")
```

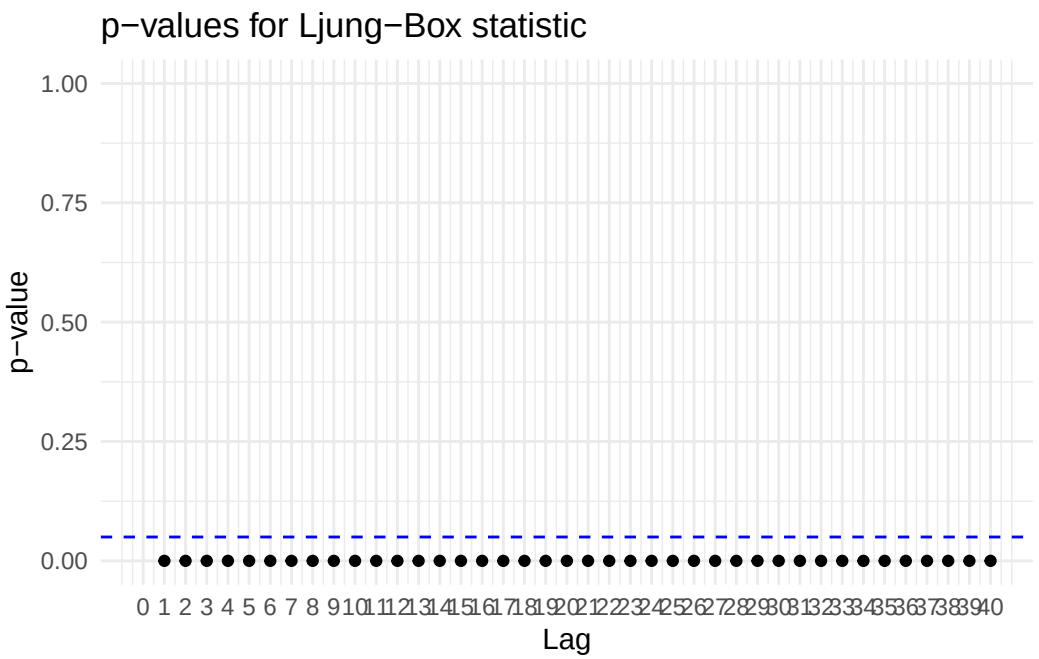



```
rw <- Arima(ts_imports_detrend, order = c(0,1,0))

rw |>
  summary()
```

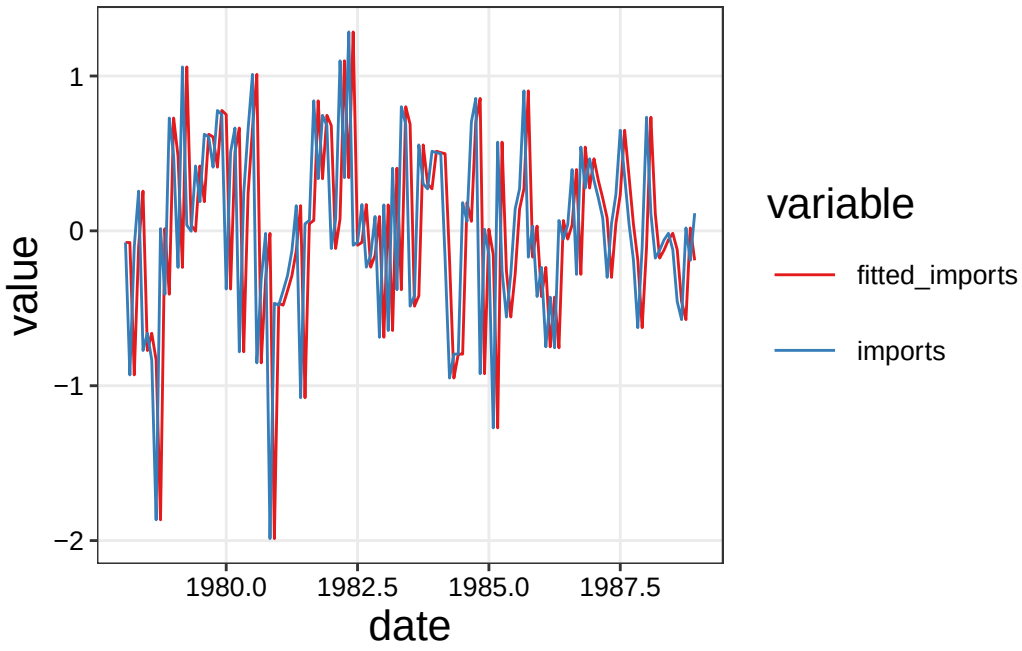
```
Series: ts_imports_detrend
ARIMA(0,1,0)

sigma^2 = 0.4976:  log likelihood = -139.09
AIC=280.18   AICc=280.21   BIC=283.05

Training set error measures:
                     ME      RMSE       MAE      MPE     MAPE      MASE
Training set 0.001443722 0.7026842 0.5271425 281.8516 433.4365 0.7652472
```

```
                    ACF1
Training set -0.4770779
```

```
LSTS::Box.Ljung.Test(rw$residuals, lag = 40)
```



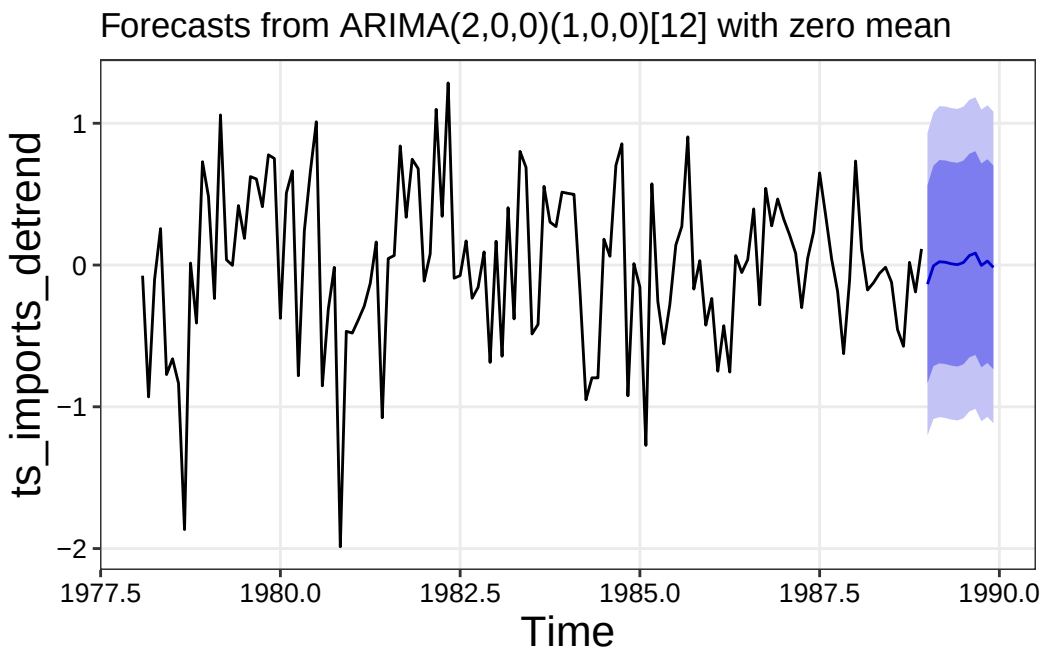

```
tibble(
  imports = rw$x,
  fitted_imports = rw$fitted,
  date = time(rw$x)
) |>
  pivot_longer(
    cols = !date,
    names_to = "variable",
    values_to = "value"
  ) |>
  ggplot(aes(x = date, y = value, color = variable)) +
  geom_line() +
  scale_color_brewer(palette = "Set1")
```



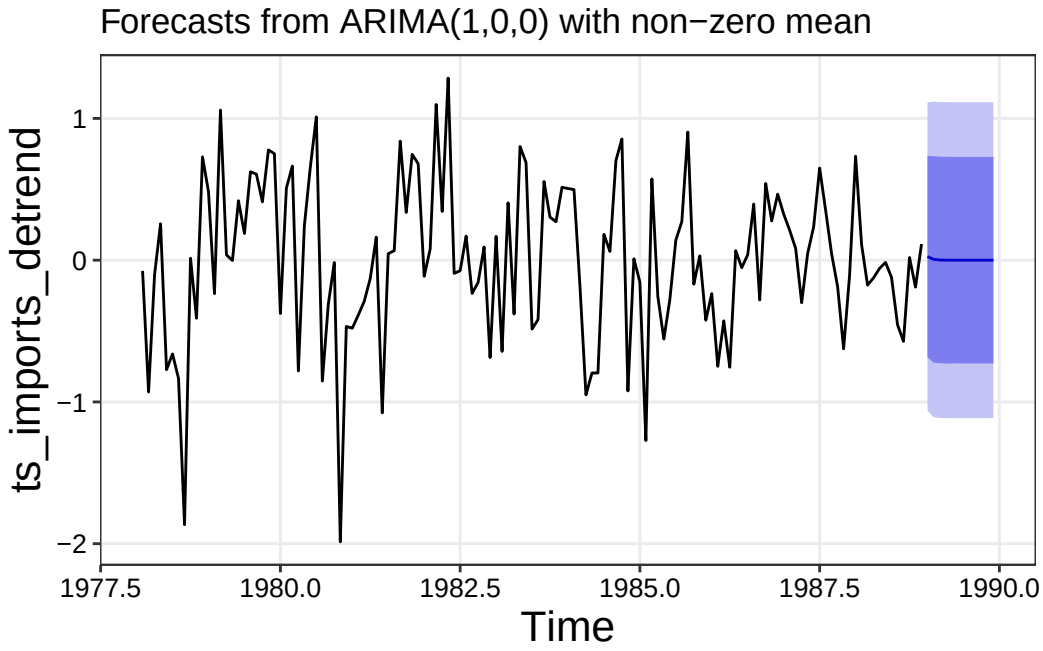

```
tibble(
  model = c("best ARIMA", "AR(1)", "RW"),
  AIC = c(AIC(gen_arima), AIC(arma), AIC(rw)),
  BIC = c(BIC(gen_arima), BIC(arma), BIC(rw)),
  RMSE = c(performance::rmse(gen_arima), rmse(arma), rmse(rw))
)
```

```
# A tibble: 3 x 4
  model        AIC   BIC  RMSE
  <chr>      <dbl> <dbl> <dbl>
1 best ARIMA  218.  229. 0.538
2 AR(1)       221.  230. 0.550
3 RW          280.  283. 0.703
```

```
autoplot(forecast(gen_arima, h = 12))
```




```
autoplot(forecast(arma, h = 12))
```



```r
autoplot(forecast(rw, h = 12))
```