Lista II: Q1

Alberson Miranda

2022-11-27

```r
# configurações
knitr::opts_chunk$set(
  fig.output = "70%"
)

# reproducibilidade
set.seed(1)

# pacotes
pacman::p_load(
    "ggplot2",
    "tsibble",
    "fable",
    "feasts",
    "fabletools",
    "urca"
)
```

1 MODELAGEM BOX-JENKINS: SÉRIE I

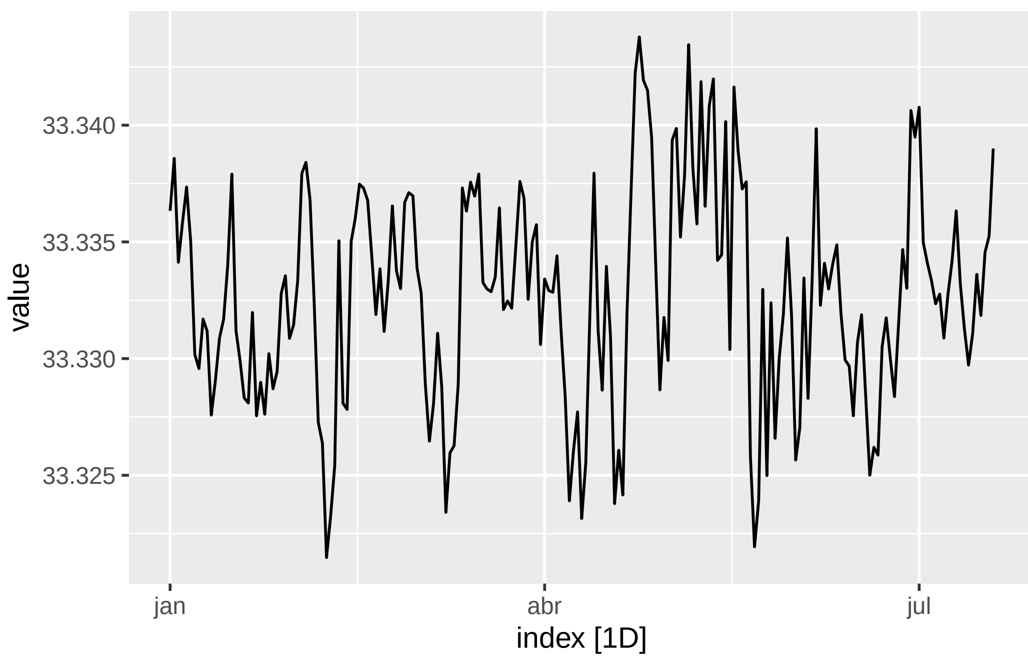
O primeiro passo é a importação e visualização da série. Como não há informação sobre o período, usarei diário e tentarei identificar a partir de um padrão sazonal, se houver.

```r
# importando dados
load("data/lista II.RData")
data = data.frame(
    value = conjunto1[, 1],
    index = seq(
        as.Date("2000-01-01"),
```

```
        by = 1,
        length.out = length(conjunto1[, 1])
    )
) |> tsibble(index = index)
```

A série é compacta, ou seja, de amplitude baixa, não requerindo transformação para redução de variância.

```
# plot série
autoplot(data, .vars = value)
```



O segundo passo é testar se a série é estacionária no primeiro momento. Não há evidências de raiz unitária tanto nos testes quanto nos gráficos de autocorrelação. Para os testes ADF, iniciei com a especificação com tendência. Não sendo significativo o coeficiente `tt`, passei para a especificação com *drift*, sendo tanto o intercepto quanto `z.lag.1` significativos, adoto esta como a correta especificação e, assim como nos testes de Phillips-Perron e KPSS, não há indicativo de raiz unitária.

```
# KPSS test
data |>
    features(value, unitroot_kpss)
```

```
# A tibble: 1 x 2
  kpss_stat kpss_pvalue
      <dbl>       <dbl>
1     0.108         0.1
```

```r
# Phillips-Perron test
data |>
    features(value, unitroot_pp)
```

```
# A tibble: 1 x 2
  pp_stat pp_pvalue
    <dbl>     <dbl>
1   -6.43      0.01
```

```r
# Augmented-Dickey-Fuller test
data |>
    (\(x) ur.df(x$value, selectlags = "AIC", type = "trend", lags = 12))() |>
    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
-0.0102211 -0.0022799  0.0000216  0.0020829  0.0102631

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.141e+01  2.057e+00   5.549 9.87e-08 ***
z.lag.1     -3.424e-01  6.171e-02  -5.549 9.87e-08 ***
tt           3.313e-06  4.807e-06   0.689    0.492
z.diff.lag  -8.474e-03  7.397e-02  -0.115    0.909
---
```

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

Residual standard error: 0.00356 on 184 degrees of freedom
Multiple R-squared:  0.171, Adjusted R-squared:  0.1574
F-statistic: 12.65 on 3 and 184 DF,  p-value: 1.487e-07


Value of test-statistic is: -5.5495 10.2859 15.411

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
data |>
    (\(x) ur.df(x$value, selectlags = "AIC", type = "drift", lags = 12))() |>
    summary()
```

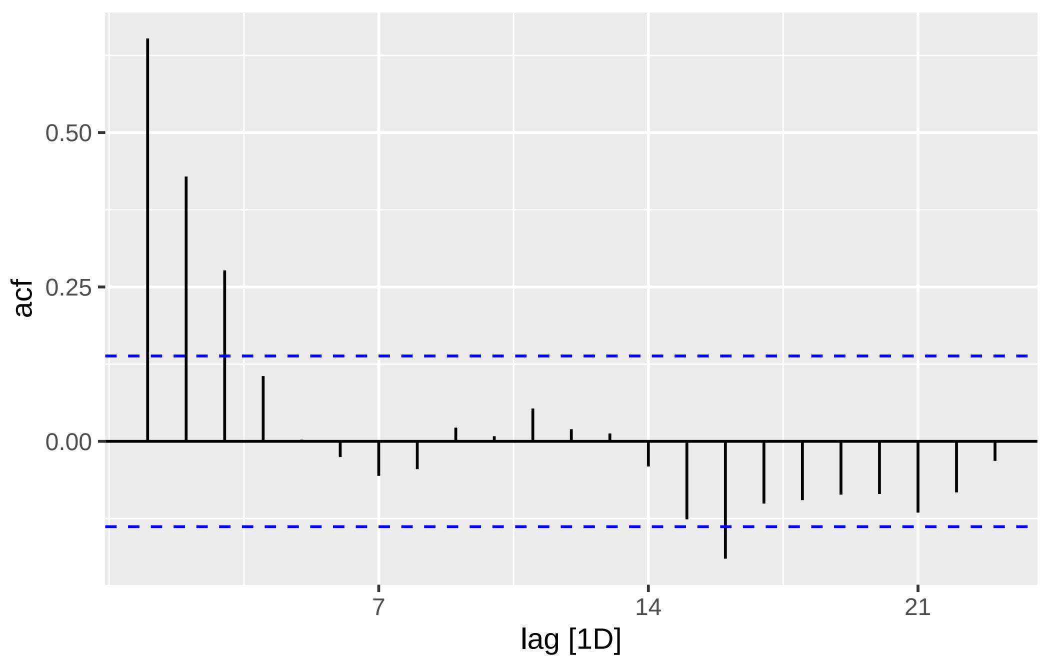
```
###############################################
# Augmented Dickey-Fuller Test Unit Root Test #
###############################################

Test regression drift


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

Residuals:
       Min         1Q     Median         3Q        Max
-0.0101264 -0.0022878 -0.0000252  0.0020877  0.0103523

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 11.27905    2.04454   5.517 1.15e-07 ***
z.lag.1     -0.33838    0.06134  -5.517 1.15e-07 ***
z.diff.lag  -0.01060    0.07380  -0.144    0.886
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 0.003555 on 185 degrees of freedom
Multiple R-squared:  0.1688,	Adjusted R-squared:  0.1598 
F-statistic: 18.79 on 2 and 185 DF,  p-value: 3.73e-08


Value of test-statistic is: -5.5166 15.2347 

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

A seguir, pode-se perceber decaimento na ACF e um pico na PACF, sugerindo um processo AR(1).

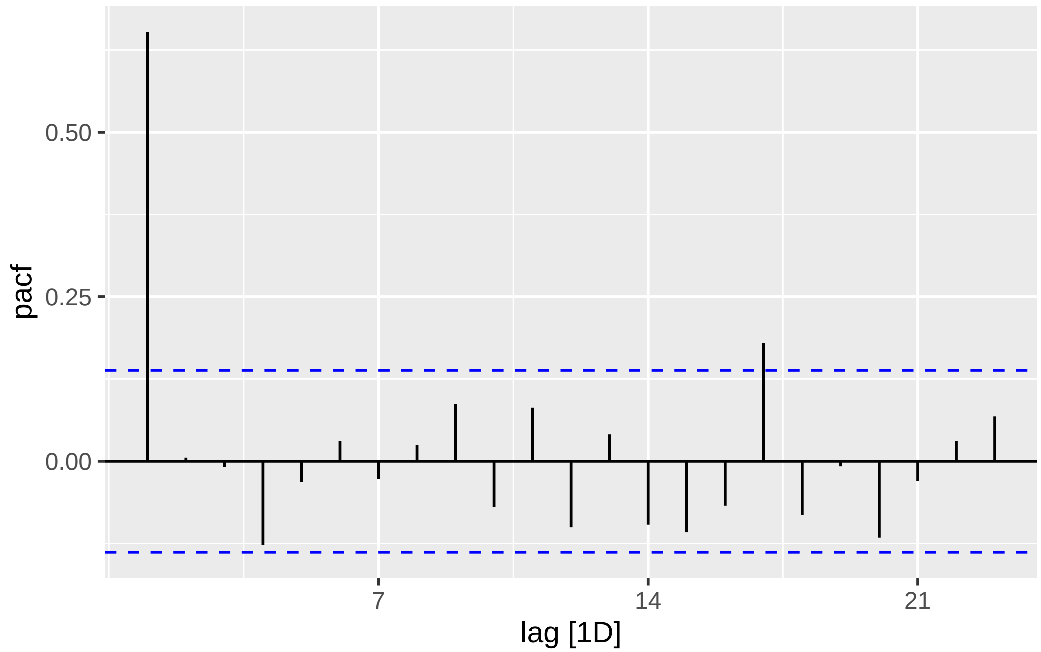
```r
# ACF
data |> ACF() |> autoplot()
```

```
Response variable not specified, automatically selected `var = value`
```




```r
# ACF
data |> PACF() |> autoplot()
```

```
Response variable not specified, automatically selected `var = value`
```



Além do AR(1), também realizei uma *grid search*, que consiste na estimação de todas as combinações possíveis de modelos ARIMA dada uma restrição de coeficientes. Neste caso, como a análise do correlograma sugere um AR(1), optei por uma restrição parcimoniosa, com no máximo 3 coeficientes (AR ou MA) e sem testar modelos integrados, uma vez que foi constatada a estacionaridade. Dentre os modelos estimados, o de menor critério de informação foi o AR(1), no mesmo sentido da análise visual do correlograma.

```
data_fit = data |>
  model(
    ar1 = ARIMA(
      value ~ 1 + pdq(1, 0, 0)
    ),
    search = ARIMA(
        value,
        stepwise = FALSE,
        trace = TRUE,
        order_constraint = p + q + P + Q <= 3 & (constant + d + D <= 1))
)
```

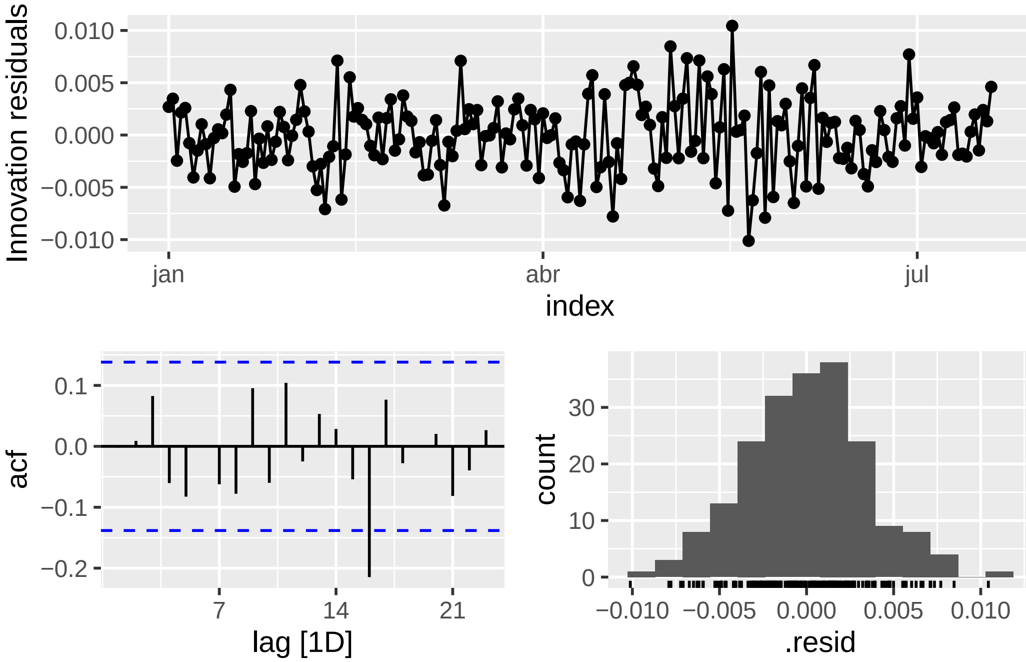
```
ARIMA(0,0,0)(0,0,0)[7]+c    -1589.386363
```

```
ARIMA(1,0,0)(0,0,0)[7]+c    -1700.363424
ARIMA(2,0,0)(0,0,0)[7]+c    -1698.315341
ARIMA(3,0,0)(0,0,0)[7]+c    -1695.712579
ARIMA(0,0,1)(0,0,0)[7]+c    -1668.586911
ARIMA(1,0,1)(0,0,0)[7]+c    -1698.303988
ARIMA(2,0,1)(0,0,0)[7]+c    -1696.822220
ARIMA(0,0,2)(0,0,0)[7]+c    -1682.955592
ARIMA(1,0,2)(0,0,0)[7]+c    -1696.222233
ARIMA(0,0,3)(0,0,0)[7]+c    -1695.230046
ARIMA(0,0,0)(1,0,0)[7]+c    -1585.461121
ARIMA(1,0,0)(1,0,0)[7]+c    -1696.000612
ARIMA(2,0,0)(1,0,0)[7]+c    -1693.062626
ARIMA(0,0,1)(1,0,0)[7]+c    -1663.280799
ARIMA(1,0,1)(1,0,0)[7]+c    -1693.950922
ARIMA(0,0,2)(1,0,0)[7]+c    -1678.426909
ARIMA(0,0,0)(2,0,0)[7]+c    -1578.757729
ARIMA(1,0,0)(2,0,0)[7]+c    -1688.504270
ARIMA(0,0,1)(2,0,0)[7]+c    -1656.733042
ARIMA(0,0,0)(0,0,1)[7]+c    -1588.036672
ARIMA(1,0,0)(0,0,1)[7]+c    -1699.009375
ARIMA(2,0,0)(0,0,1)[7]+c    -1697.025695
ARIMA(0,0,1)(0,0,1)[7]+c    -1666.716920
ARIMA(1,0,1)(0,0,1)[7]+c    -1696.968352
ARIMA(0,0,2)(0,0,1)[7]+c    -1681.620771
ARIMA(0,0,0)(1,0,1)[7]+c    -1595.850518
ARIMA(1,0,0)(1,0,1)[7]+c    -1697.251094
ARIMA(0,0,1)(1,0,1)[7]+c    -1667.413004
ARIMA(0,0,0)(2,0,1)[7]+c    Inf
ARIMA(0,0,0)(0,0,2)[7]+c    -1586.749289
ARIMA(1,0,0)(0,0,2)[7]+c    -1696.969600
ARIMA(0,0,1)(0,0,2)[7]+c    -1665.570336
ARIMA(0,0,0)(1,0,2)[7]+c    -1593.766397
ARIMA(0,0,0)(0,0,0)[7]      1982.061534
ARIMA(1,0,0)(0,0,0)[7]      Inf
ARIMA(2,0,0)(0,0,0)[7]      Inf
ARIMA(3,0,0)(0,0,0)[7]      Inf
ARIMA(0,0,1)(0,0,0)[7]      1725.595542
ARIMA(1,0,1)(0,0,0)[7]      Inf
ARIMA(2,0,1)(0,0,0)[7]      Inf
ARIMA(0,0,2)(0,0,0)[7]      1529.598405
ARIMA(1,0,2)(0,0,0)[7]      Inf
ARIMA(0,0,3)(0,0,0)[7]      1393.372344
ARIMA(0,0,0)(1,0,0)[7]      Inf
```

```
ARIMA(1,0,0)(1,0,0)[7]      Inf
ARIMA(2,0,0)(1,0,0)[7]      Inf
ARIMA(0,0,1)(1,0,0)[7]      Inf
ARIMA(1,0,1)(1,0,0)[7]      Inf
ARIMA(0,0,2)(1,0,0)[7]      Inf
ARIMA(0,0,0)(2,0,0)[7]      Inf
ARIMA(1,0,0)(2,0,0)[7]      Inf
ARIMA(0,0,1)(2,0,0)[7]      Inf
ARIMA(0,0,0)(0,0,1)[7]      1758.879724
ARIMA(1,0,0)(0,0,1)[7]      Inf
ARIMA(2,0,0)(0,0,1)[7]      Inf
ARIMA(0,0,1)(0,0,1)[7]      1555.857332
ARIMA(1,0,1)(0,0,1)[7]      Inf
ARIMA(0,0,2)(0,0,1)[7]      1426.549252
ARIMA(0,0,0)(1,0,1)[7]      Inf
ARIMA(1,0,0)(1,0,1)[7]      Inf
ARIMA(0,0,1)(1,0,1)[7]      Inf
ARIMA(0,0,0)(2,0,1)[7]      Inf
ARIMA(0,0,0)(0,0,2)[7]      1618.905610
ARIMA(1,0,0)(0,0,2)[7]      Inf
ARIMA(0,0,1)(0,0,2)[7]      1439.537620
ARIMA(0,0,0)(1,0,2)[7]      Inf

--- Re-estimating best models without approximation ---

ARIMA(1,0,0)(0,0,0)[7]+c    -1700.178926
```

Na etapa de diagnóstico, verificamos se os resíduos são ruído branco e aproximadamente normalmente distribuídos, o que indica que o modelo foi bem especificado. No teste de Ljung-Box, não há evidência suficiente para rejeitar a hipótese nula de autocorrelação dos resíduos. No mesmo sentido, o histograma também aponta para o diagnóstico positivo do modelo.

```r
# teste de Ljung-Box
data_fit |>
  dplyr::select(ar1) |>
  gg_tsresiduals()
```




```r
augment(data_fit) |>
  dplyr::filter(.model == "ar1") |>
  features(.innov, ljung_box, lag = 12, dof = 2)
```

```
# A tibble: 1 x 3
  .model lb_stat lb_pvalue
  <chr>    <dbl>     <dbl>
1 ar1       10.9     0.368
```

Conclui-se pela seleção do AR(1), de seguinte equação:

$$y_t = \phi_0 + \phi_1 y_{t-1}$$

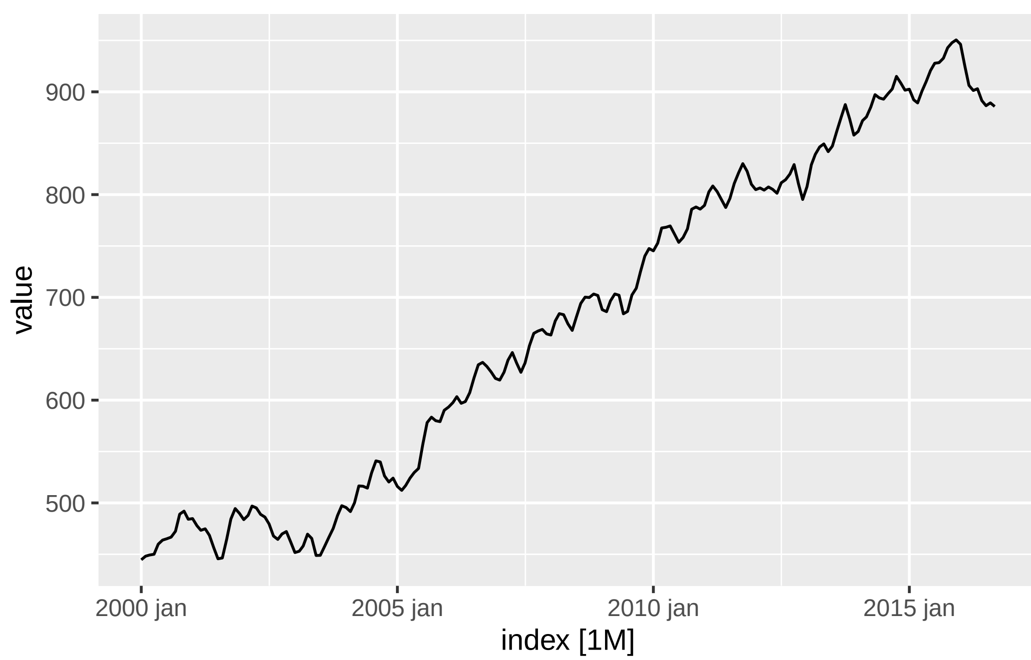
A MODELAGEM BOX-JENKINS: SÉRIE II

Visualizando a série, nota-se imediatamente que não é estacionária por conta da tendência.

```r
# importando dados
data = data.frame(
    value = conjunto1[, 6],
```

```r
    index = yearmonth(
      seq(
        as.Date("2000-01-01"),
        by = "month",
        length.out = length(conjunto1[, 1])
      )
    )
) |> tsibble(index = index)
```

```r
# plot série
autoplot(data, .vars = value)
```



A estratégia para correção é a diferenciação. Com o teste KPSS, a primeira sugestão é de que a série é estacionária em primeiras diferenças, sem raiz unitária sazonal.

```r
# KPSS test e sazonal
data |>
    features(value, unitroot_ndiffs)
```

```
# A tibble: 1 x 1
  ndiffs
```

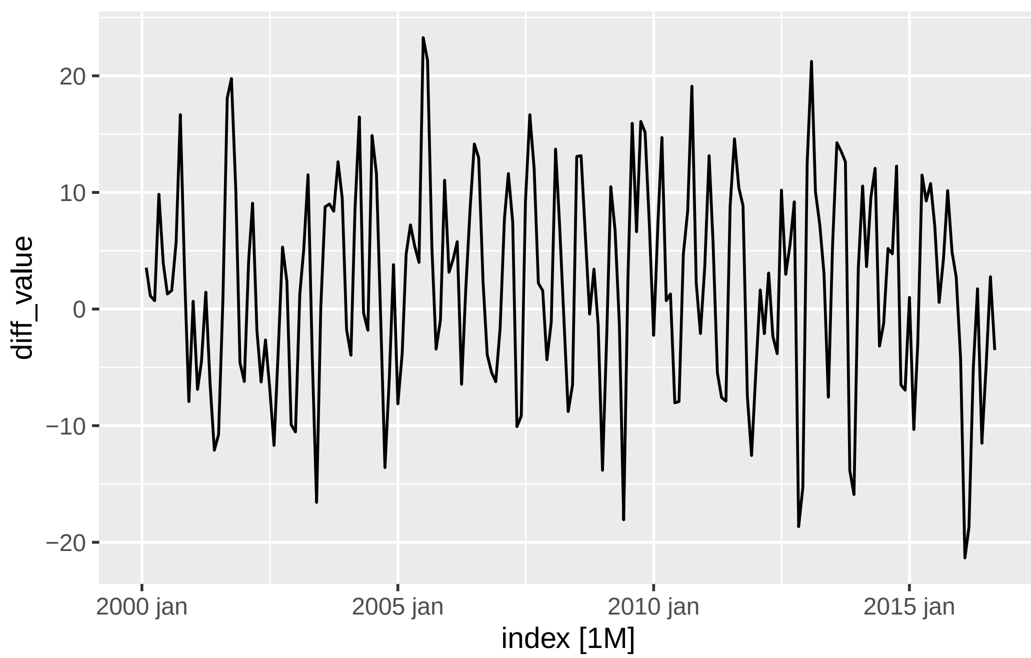
```
   <int>
1      1
```

```
data |>
    features(value, unitroot_nsdiffs)
```

```
# A tibble: 1 x 1
  nsdiffs
    <int>
1       0
```

```
# diferenciando
data$diff_value = difference(data$value)

data |>
  autoplot(.vars = diff_value)
```

```
Warning: Removed 1 row containing missing values (`geom_line()`).
```



Na série diferenciada, não há evidências de raiz unitária tanto nos testes quanto nos gráficos de autocorrelação. Para os testes ADF, iniciei com a especificação com tendência. Não sendo significativo o coeficiente `tt`, passei para a especificação com *drift*, sendo tanto o intercepto quanto `z.lag.1` significativos, adoto esta como a correta especificação e, assim como nos testes de Phillips-Perron e KPSS, não há indicativo de raiz unitária.

```r
# KPSS test
data |>
  features(diff_value, unitroot_kpss)
```

```
# A tibble: 1 x 2
  kpss_stat kpss_pvalue
      <dbl>       <dbl>
1     0.100         0.1
```

```r
# Phillips-Perron test
data |>
  features(diff_value, unitroot_pp)
```

```
# A tibble: 1 x 2
  pp_stat pp_pvalue
    <dbl>     <dbl>
1   -8.31      0.01
```

```r
# Augmented-Dickey-Fuller test
data |>
  na.omit() |>
  (\(x) ur.df(x$diff_value, selectlags = "AIC", type = "trend", lags = 12))() |>
  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
```

```
-17.9098  -3.4490   0.5295   4.2145  12.6605

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)   1.8911956  1.0526862   1.797 0.074163 .
z.lag.1      -0.7920838  0.2148502  -3.687 0.000304 ***
tt           -0.0007629  0.0084378  -0.090 0.928060
z.diff.lag1   0.5884153  0.2093105   2.811 0.005509 **
z.diff.lag2  -0.1205004  0.2031784  -0.593 0.553909
z.diff.lag3   0.4398270  0.1868001   2.355 0.019674 *
z.diff.lag4  -0.1387117  0.1797101  -0.772 0.441254
z.diff.lag5   0.4108668  0.1660764   2.474 0.014333 *
z.diff.lag6  -0.0965456  0.1633881  -0.591 0.555365
z.diff.lag7   0.3337387  0.1442319   2.314 0.021856 *
z.diff.lag8  -0.0726282  0.1409957  -0.515 0.607138
z.diff.lag9   0.1083096  0.1151223   0.941 0.348116
z.diff.lag10 -0.2205951  0.1098470  -2.008 0.046186 *
z.diff.lag11 -0.0415018  0.0804456  -0.516 0.606588
z.diff.lag12  0.1465711  0.0779209   1.881 0.061658 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.112 on 172 degrees of freedom
Multiple R-squared:  0.6248,    Adjusted R-squared:  0.5942
F-statistic: 20.46 on 14 and 172 DF,  p-value: < 2.2e-16


Value of test-statistic is: -3.6867 4.7354 7.0691

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
```

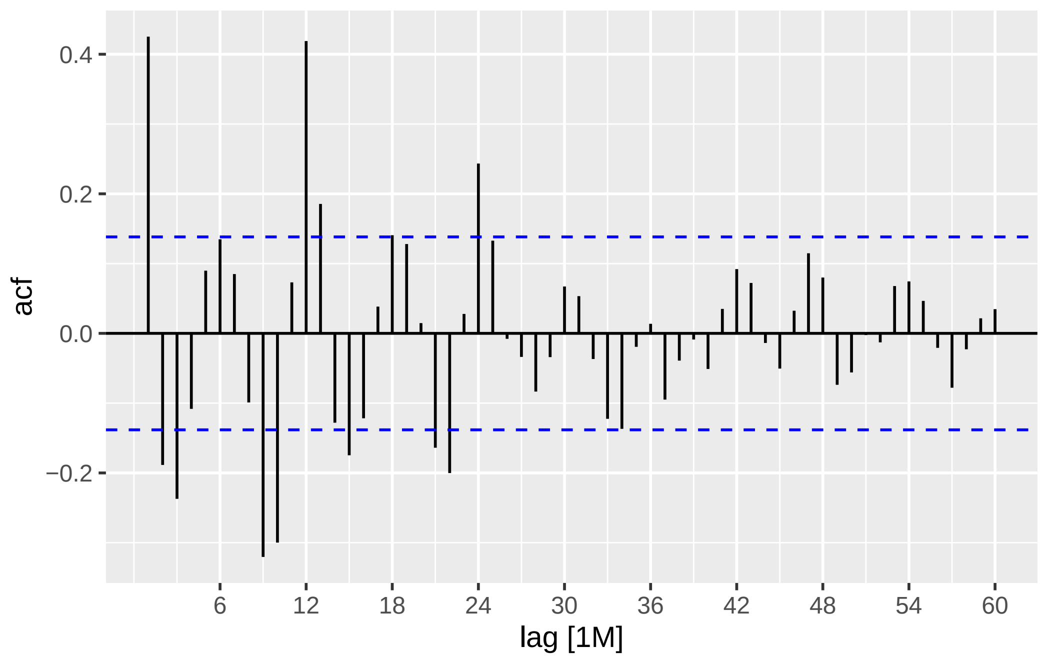
```
data |>
  na.omit() |>
  (\(x) ur.df(x$diff_value, selectlags = "AIC", type = "drift", lags = 12))() |>
  summary()
```

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

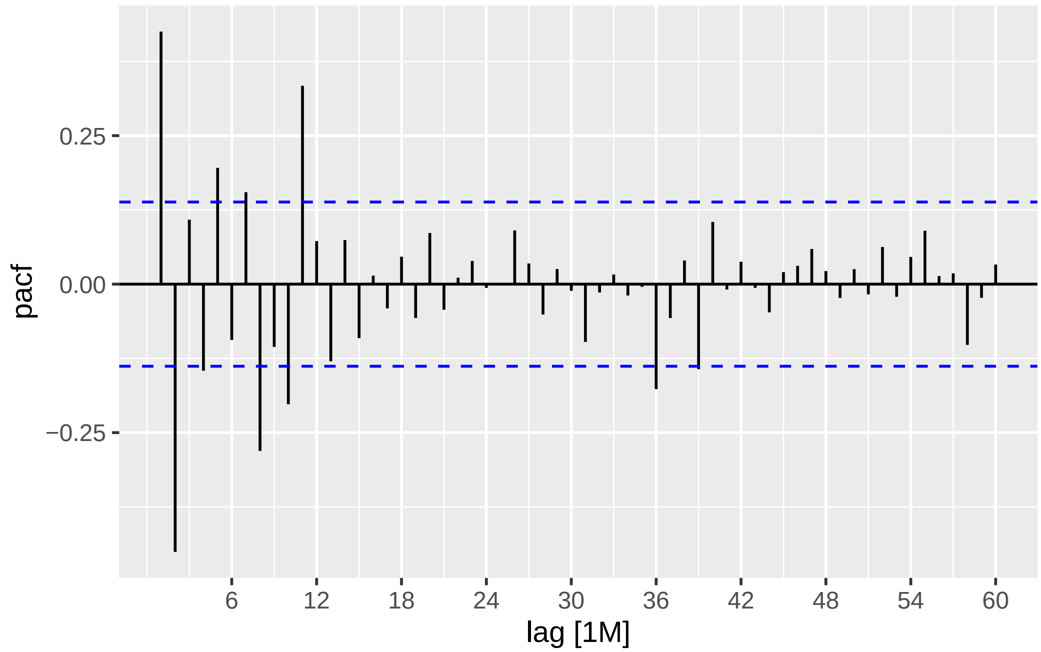
```
# Augmented Dickey-Fuller Test Unit Root Test #
###############################################

Test regression drift


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

Residuals:
     Min       1Q   Median       3Q      Max
-17.9540  -3.4530   0.4861   4.1709  12.5999

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   1.81874    0.68064   2.672 0.008258 **
z.lag.1      -0.79544    0.21100  -3.770 0.000224 ***
z.diff.lag1   0.59173    0.20548   2.880 0.004483 **
z.diff.lag2  -0.11726    0.19941  -0.588 0.557291
z.diff.lag3   0.44286    0.18324   2.417 0.016697 *
z.diff.lag4  -0.13592    0.17652  -0.770 0.442369
z.diff.lag5   0.41352    0.16300   2.537 0.012070 *
z.diff.lag6  -0.09400    0.16048  -0.586 0.558816
z.diff.lag7   0.33595    0.14174   2.370 0.018885 *
z.diff.lag8  -0.07066    0.13890  -0.509 0.611619
z.diff.lag9   0.10976    0.11366   0.966 0.335547
z.diff.lag10 -0.21934    0.10866  -2.019 0.045066 *
z.diff.lag11 -0.04082    0.07987  -0.511 0.609893
z.diff.lag12  0.14712    0.07746   1.899 0.059183 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.095 on 173 degrees of freedom
Multiple R-squared:  0.6248,    Adjusted R-squared:  0.5966
F-statistic: 22.16 on 13 and 173 DF,  p-value: < 2.2e-16


Value of test-statistic is: -3.7698 7.14

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

A seguir, pode-se perceber decaimento nas lags sazonais na ACF, indicando a presença de componente sazonal. Além disso, a análise dos correlogramas sugerem modelos de ordem inferior ou igual a 3 — três picos significativos na ACF e dois na PACF. Porém, a identificação de modelos ARMA é inconclusiva a partir da análise do correlograma exclusivamente.

```
# ACF
data |> ACF(diff_value, lag_max = 60) |> autoplot()
```




```
# ACF
data |> PACF(diff_value, lag_max = 60) |> autoplot()
```



Para avaliar candidatos a modelos, realizei uma *grid search* impondo as seguintes restrições:

1. $d = 1$, $D = 0$ e $d + D +$ constante $<= 2$
2. $p + q + P + Q <= 5$

```
data |>
  model(
    search = ARIMA(
      value,
      ic = "aic",
      stepwise = FALSE,
      trace = TRUE,
      order_constraint = (p + q + P + Q <= 5) &
        d == 1 &
        D == 0 &
        constant == 1
  )
)
```

```
ARIMA(0,1,0)(0,0,0)[12]+c   1437.746540
ARIMA(1,1,0)(0,0,0)[12]+c   1400.743577
ARIMA(2,1,0)(0,0,0)[12]+c   1357.908576
```

```
ARIMA(3,1,0)(0,0,0)[12]+c   1358.280059
ARIMA(4,1,0)(0,0,0)[12]+c   1355.675578
ARIMA(5,1,0)(0,0,0)[12]+c   1350.559786
ARIMA(0,1,1)(0,0,0)[12]+c   1336.639861
ARIMA(1,1,1)(0,0,0)[12]+c   1340.055957
ARIMA(2,1,1)(0,0,0)[12]+c   1340.606033
ARIMA(3,1,1)(0,0,0)[12]+c   1339.452417
ARIMA(4,1,1)(0,0,0)[12]+c   1337.362051
ARIMA(0,1,2)(0,0,0)[12]+c   1338.490236
ARIMA(1,1,2)(0,0,0)[12]+c   1336.992350
ARIMA(2,1,2)(0,0,0)[12]+c   1338.867528
ARIMA(3,1,2)(0,0,0)[12]+c   1340.520759
ARIMA(0,1,3)(0,0,0)[12]+c   1336.234907
ARIMA(1,1,3)(0,0,0)[12]+c   1337.488084
ARIMA(2,1,3)(0,0,0)[12]+c   Inf
ARIMA(0,1,4)(0,0,0)[12]+c   1336.347799
ARIMA(1,1,4)(0,0,0)[12]+c   1339.128960
ARIMA(0,1,5)(0,0,0)[12]+c   1334.782453
ARIMA(0,1,0)(1,0,0)[12]+c   1402.570657
ARIMA(1,1,0)(1,0,0)[12]+c   1349.906800
ARIMA(2,1,0)(1,0,0)[12]+c   1325.197821
ARIMA(3,1,0)(1,0,0)[12]+c   1321.281273
ARIMA(4,1,0)(1,0,0)[12]+c   1316.627967
ARIMA(0,1,1)(1,0,0)[12]+c   1292.725358
ARIMA(1,1,1)(1,0,0)[12]+c   1300.378066
ARIMA(2,1,1)(1,0,0)[12]+c   1296.747908
ARIMA(3,1,1)(1,0,0)[12]+c   1299.490618
ARIMA(0,1,2)(1,0,0)[12]+c   1294.719267
ARIMA(1,1,2)(1,0,0)[12]+c   1293.050580
ARIMA(2,1,2)(1,0,0)[12]+c   1298.047996
ARIMA(0,1,3)(1,0,0)[12]+c   1296.664829
ARIMA(1,1,3)(1,0,0)[12]+c   1294.284391
ARIMA(0,1,4)(1,0,0)[12]+c   1298.276716
ARIMA(0,1,0)(2,0,0)[12]+c   1397.176416
ARIMA(1,1,0)(2,0,0)[12]+c   1348.459565
ARIMA(2,1,0)(2,0,0)[12]+c   1323.790985
ARIMA(3,1,0)(2,0,0)[12]+c   1318.819544
ARIMA(0,1,1)(2,0,0)[12]+c   1292.680208
ARIMA(1,1,1)(2,0,0)[12]+c   1301.825113
ARIMA(2,1,1)(2,0,0)[12]+c   1296.302358
ARIMA(0,1,2)(2,0,0)[12]+c   1294.562441
ARIMA(1,1,2)(2,0,0)[12]+c   1300.031592
ARIMA(0,1,3)(2,0,0)[12]+c   1296.534677
```

```
ARIMA(0,1,0)(0,0,1)[12]+c   1409.998876
ARIMA(1,1,0)(0,0,1)[12]+c   1363.609338
ARIMA(2,1,0)(0,0,1)[12]+c   1334.956399
ARIMA(3,1,0)(0,0,1)[12]+c   1331.560493
ARIMA(4,1,0)(0,0,1)[12]+c   1327.464316
ARIMA(0,1,1)(0,0,1)[12]+c   1301.724023
ARIMA(1,1,1)(0,0,1)[12]+c   1305.101113
ARIMA(2,1,1)(0,0,1)[12]+c   1308.088883
ARIMA(3,1,1)(0,0,1)[12]+c   1309.653416
ARIMA(0,1,2)(0,0,1)[12]+c   1303.721696
ARIMA(1,1,2)(0,0,1)[12]+c   Inf
ARIMA(2,1,2)(0,0,1)[12]+c   Inf
ARIMA(0,1,3)(0,0,1)[12]+c   1304.958754
ARIMA(1,1,3)(0,0,1)[12]+c   Inf
ARIMA(0,1,4)(0,0,1)[12]+c   1306.870600
ARIMA(0,1,0)(1,0,1)[12]+c   1403.448311
ARIMA(1,1,0)(1,0,1)[12]+c   1351.702381
ARIMA(2,1,0)(1,0,1)[12]+c   1327.078436
ARIMA(3,1,0)(1,0,1)[12]+c   1323.279364
ARIMA(0,1,1)(1,0,1)[12]+c   1294.639355
ARIMA(1,1,1)(1,0,1)[12]+c   1302.358114
ARIMA(2,1,1)(1,0,1)[12]+c   1298.747827
ARIMA(0,1,2)(1,0,1)[12]+c   1296.625418
ARIMA(1,1,2)(1,0,1)[12]+c   1295.049111
ARIMA(0,1,3)(1,0,1)[12]+c   1298.576122
ARIMA(0,1,0)(2,0,1)[12]+c   1390.775075
ARIMA(1,1,0)(2,0,1)[12]+c   1348.261266
ARIMA(2,1,0)(2,0,1)[12]+c   1324.585829
ARIMA(0,1,1)(2,0,1)[12]+c   1293.364484
ARIMA(1,1,1)(2,0,1)[12]+c   1298.988623
ARIMA(0,1,2)(2,0,1)[12]+c   1295.002440
ARIMA(0,1,0)(0,0,2)[12]+c   1390.940068
ARIMA(1,1,0)(0,0,2)[12]+c   1347.710071
ARIMA(2,1,0)(0,0,2)[12]+c   1324.551274
ARIMA(3,1,0)(0,0,2)[12]+c   1322.586718
ARIMA(0,1,1)(0,0,2)[12]+c   1291.857638
ARIMA(1,1,1)(0,0,2)[12]+c   1294.617182
ARIMA(2,1,1)(0,0,2)[12]+c   1298.023506
ARIMA(0,1,2)(0,0,2)[12]+c   1293.478278
ARIMA(1,1,2)(0,0,2)[12]+c   Inf
ARIMA(0,1,3)(0,0,2)[12]+c   1295.340501
ARIMA(0,1,0)(1,0,2)[12]+c   1400.796589
ARIMA(1,1,0)(1,0,2)[12]+c   1351.778712
```

```
ARIMA(2,1,0)(1,0,2)[12]+c   1323.978327
ARIMA(0,1,1)(1,0,2)[12]+c   1296.346899
ARIMA(1,1,1)(1,0,2)[12]+c   1302.532166
ARIMA(0,1,2)(1,0,2)[12]+c   1298.302977
ARIMA(0,1,0)(2,0,2)[12]+c   1391.092422
ARIMA(1,1,0)(2,0,2)[12]+c   1349.476212
ARIMA(0,1,1)(2,0,2)[12]+c   1295.167957

--- Re-estimating best models without approximation ---

ARIMA(0,1,1)(0,0,2)[12]+c   1298.688664


# A mable: 1 x 1
                              search
                             <model>
1 <ARIMA(0,1,1)(0,0,2)[12] w/ drift>
```

A estratégia adotada foi selecionar os modelos com $\Delta \text{AIC} < 2$ em relação ao modelo de menor AIC, além de sua combinação por média simples. São eles:

```
tibble::tribble(
  ~Modelo, ~AIC, ~delta_AIC,
  "ARIMA(0,1,1)(0,0,2)[12]+c", 1291.86, 0,
  "ARIMA(0,1,1)(2,0,0)[12]+c", 1292.68, 0.82,
  "ARIMA(0,1,1)(1,0,0)[12]+c", 1292.73, 0.87,
  "ARIMA(1,1,2)(1,0,0)[12]+c", 1293.05, 1.19,
  "ARIMA(0,1,1)(2,0,1)[12]+c", 1293.36, 1.51,
  "ARIMA(0,1,2)(0,0,2)[12]+c", 1293.48, 1.62
)
```

```
# A tibble: 6 x 3
  Modelo                       AIC delta_AIC
  <chr>                      <dbl>     <dbl>
1 ARIMA(0,1,1)(0,0,2)[12]+c 1292.      0
2 ARIMA(0,1,1)(2,0,0)[12]+c 1293.      0.82
3 ARIMA(0,1,1)(1,0,0)[12]+c 1293.      0.87
4 ARIMA(1,1,2)(1,0,0)[12]+c 1293.      1.19
5 ARIMA(0,1,1)(2,0,1)[12]+c 1293.      1.51
6 ARIMA(0,1,2)(0,0,2)[12]+c 1293.      1.62
```

```r
data_fit = data |>
  model(
    arima011002 = ARIMA(
      value ~ 1 + pdq(0, 1, 1) + PDQ(0, 0, 2)
    ),
    arima011200 = ARIMA(
      value ~ 1 + pdq(0, 1, 1) + PDQ(2, 0, 0)
    ),
    arima011100 = ARIMA(
      value ~ 1 + pdq(0, 1, 1) + PDQ(1, 0, 0)
    ),
    arima112100 = ARIMA(
      value ~ 1 + pdq(1, 1, 2) + PDQ(1, 0, 0)
    ),
    arima011201 = ARIMA(
      value ~ 1 + pdq(0, 1, 1) + PDQ(2, 0, 0)
    ),
    arima012002 = ARIMA(
      value ~ 1 + pdq(0, 1, 2) + PDQ(0, 0, 2)
    )
  ) |>
  dplyr::mutate(combinacao = (
    arima011002 + arima011200 + arima011100 + arima112100 + arima011201 + arima012002
    ) / 6)
```

Na etapa de diagnóstico, todos modelos são considerados aptos para previsão, ao não apresentarem evidências para rejeitar as hipóteses nulas dos testes de Ljung-Box de ausência de autocorrelação serial e ARCH-LM de ausência de hetoscedasticidade condicional.

```r
modelos = names(data_fit)
names(modelos) = names(data_fit)

# teste de Ljung-Box
lapply(modelos, function(x) {

  augment(data_fit) |>
  dplyr::filter(.model == x) |>
  features(.innov, ljung_box, lag = 24, dof = 5)
})
```

```
$arima011002
```

```
# A tibble: 1 x 3
  .model      lb_stat lb_pvalue
  <chr>         <dbl>     <dbl>
1 arima011002    22.5     0.260

$arima011200
# A tibble: 1 x 3
  .model      lb_stat lb_pvalue
  <chr>         <dbl>     <dbl>
1 arima011200    22.5     0.260

$arima011100
# A tibble: 1 x 3
  .model      lb_stat lb_pvalue
  <chr>         <dbl>     <dbl>
1 arima011100    22.3     0.268

$arima112100
# A tibble: 1 x 3
  .model      lb_stat lb_pvalue
  <chr>         <dbl>     <dbl>
1 arima112100    20.4     0.370

$arima011201
# A tibble: 1 x 3
  .model      lb_stat lb_pvalue
  <chr>         <dbl>     <dbl>
1 arima011201    22.5     0.260

$arima012002
# A tibble: 1 x 3
  .model      lb_stat lb_pvalue
  <chr>         <dbl>     <dbl>
1 arima012002    22.3     0.267

$combinacao
# A tibble: 1 x 3
  .model     lb_stat lb_pvalue
  <chr>        <dbl>     <dbl>
1 combinacao    22.0     0.284
```

```r
# teste ARCH-LM
lapply(modelos, function(x) {

  augment(data_fit) |>
  dplyr::filter(.model == x) |>
  features(.innov, stat_arch_lm, lags = 24)
})
```

```
$arima011002
# A tibble: 1 x 2
  .model      stat_arch_lm
  <chr>              <dbl>
1 arima011002        0.123

$arima011200
# A tibble: 1 x 2
  .model      stat_arch_lm
  <chr>              <dbl>
1 arima011200        0.127

$arima011100
# A tibble: 1 x 2
  .model      stat_arch_lm
  <chr>              <dbl>
1 arima011100        0.124

$arima112100
# A tibble: 1 x 2
  .model      stat_arch_lm
  <chr>              <dbl>
1 arima112100        0.123

$arima011201
# A tibble: 1 x 2
  .model      stat_arch_lm
  <chr>              <dbl>
1 arima011201        0.127

$arima012002
# A tibble: 1 x 2
  .model      stat_arch_lm
  <chr>              <dbl>
```

```
1 arima012002        0.122

$combinacao
# A tibble: 1 x 2
  .model      stat_arch_lm
  <chr>              <dbl>
1 combinacao        0.125
```

Para testar a performance dos modelos, a série será particionada em 80%-20%, com os modelos treinados na primeira parte e treinada na última. Escolhendo, por fim, o modelo pelo critério menor erro absoluto percentual médio, a seleção ficaria com o SARIMA(0,1,1)(0,0,2), que também foi o de menor AIC.

```r
# separando amostra treino
data_treino = subset(data, index <= yearmonth("2013 apr"))

# ajustando o treino
data_treino_fit = data_treino |>
  model(
    arima011002 = ARIMA(
      value ~ 1 + pdq(0, 1, 1) + PDQ(0, 0, 2)
    ),
    arima011200 = ARIMA(
      value ~ 1 + pdq(0, 1, 1) + PDQ(2, 0, 0)
    ),
    arima011100 = ARIMA(
      value ~ 1 + pdq(0, 1, 1) + PDQ(1, 0, 0)
    ),
    arima112100 = ARIMA(
      value ~ 1 + pdq(1, 1, 2) + PDQ(1, 0, 0)
    ),
    arima011201 = ARIMA(
      value ~ 1 + pdq(0, 1, 1) + PDQ(2, 0, 0)
    ),
    arima012002 = ARIMA(
      value ~ 1 + pdq(0, 1, 2) + PDQ(0, 0, 2)
    )
  ) |>
  dplyr::mutate(combinacao = (
    arima011002 + arima011200 + arima011100 + arima112100 + arima011201 + arima012002
    ) / 6)
```

```r
# realizando previsões para fora do treino
data_treino_fc = data_treino_fit |>
  fabletools::forecast(h = 42)

# plotando
data_treino_fc |>
  autoplot(
    data |> filter_index("2012-01" ~ .),
    level = NULL
  )
```

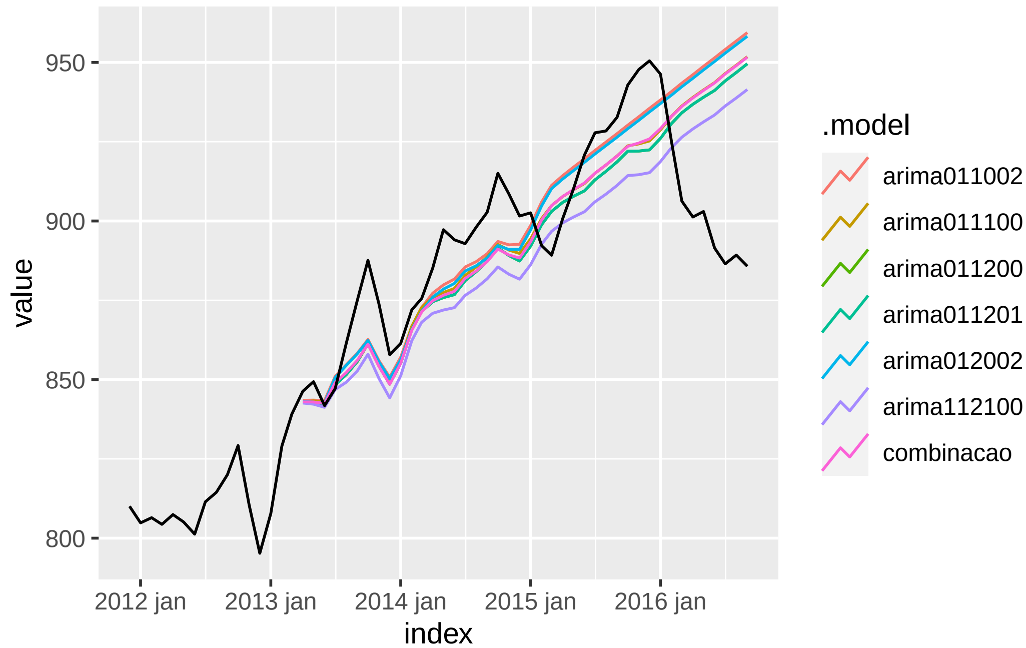


Figure 1: avaliação de performance

```r
# calculando acurácia
accuracy(data_treino_fc, data)
```

```
# A tibble: 7 x 10
  .model      .type     ME  RMSE   MAE      MPE  MAPE  MASE RMSSE  ACF1
  <chr>       <chr>  <dbl> <dbl> <dbl>    <dbl> <dbl> <dbl> <dbl> <dbl>
1 arima011002 Test  -4.63   26.1  18.0 -0.524    2.00 0.531 0.655 0.860
2 arima011100 Test  -0.185  24.3  18.2 -0.0348   2.03 0.539 0.610 0.851
3 arima011200 Test   1.36   24.0  18.6  0.136    2.06 0.549 0.603 0.849
```

```
4 arima011201 Test   1.36   24.0  18.6  0.136    2.06 0.549 0.603 0.849
5 arima012002 Test  -3.61   25.9  18.3 -0.410    2.04 0.541 0.650 0.859
6 arima112100 Test   6.79   24.2  20.2  0.736    2.24 0.597 0.607 0.846
7 combinacao  Test   0.183  24.4  18.5  0.00671  2.06 0.547 0.612 0.852
```

E sua equação:

$$(1-L)y_t = (1-\Theta_1 L^{12} - \Theta_2 L^{24})(1-\theta_1 L)\epsilon_t$$