Lista II: Q1

Alberson Miranda

2022-11-25

```
# 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

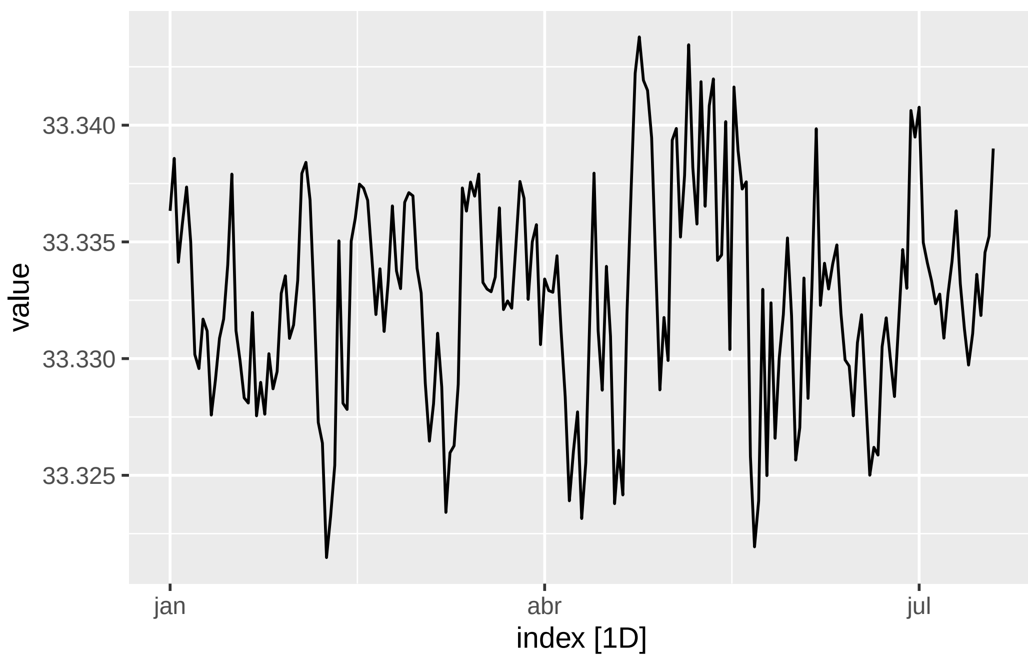
O primeiro passo é a importação e visualização da série. Como não há informação sobre o período, usarei diário e tentarei identificar a partir de um padrão sazonal, se houver.

```
# 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 inidcativo 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()
```

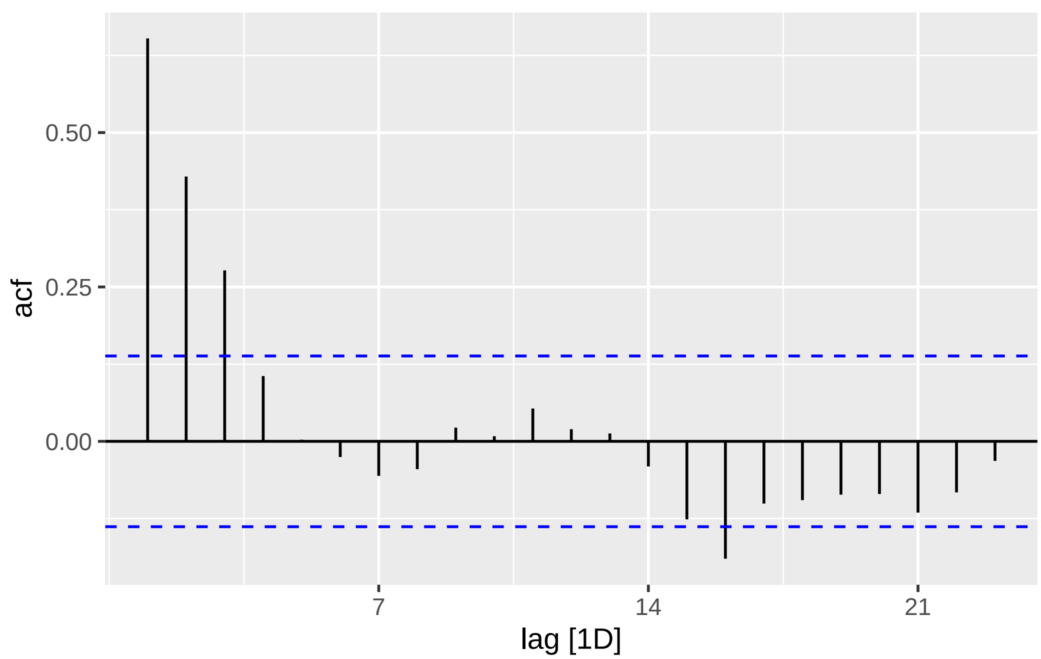
```
###############################################
# Augmented Dickey-Fuller Test Unit Root Test #
###############################################

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).

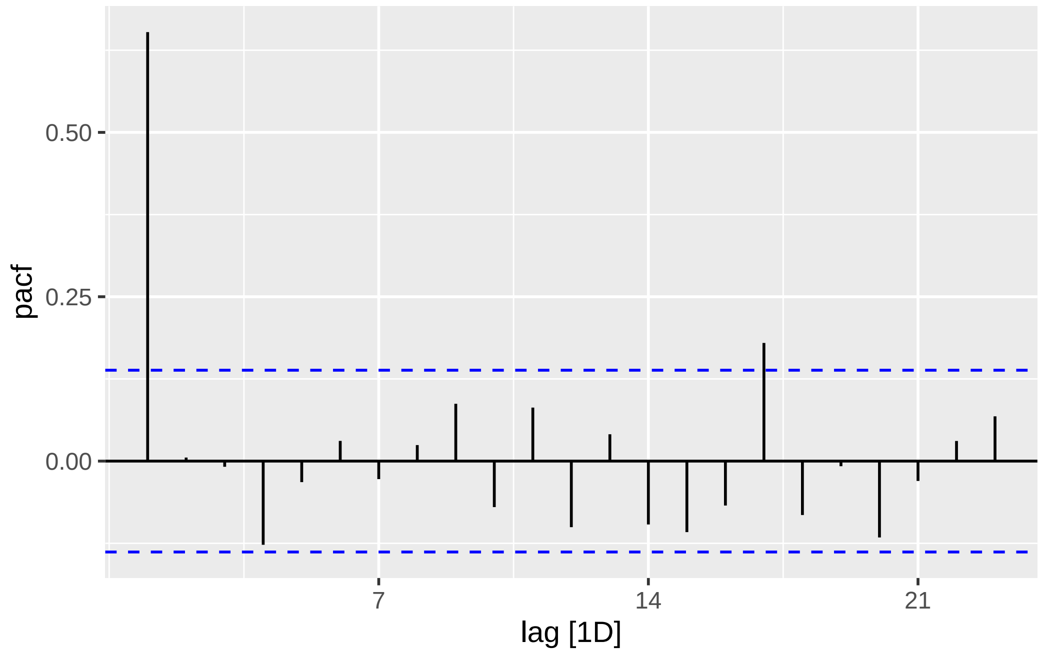
```r
# ACF
data |> ACF() |> autoplot()
```

```
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.

```r
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))
)
```

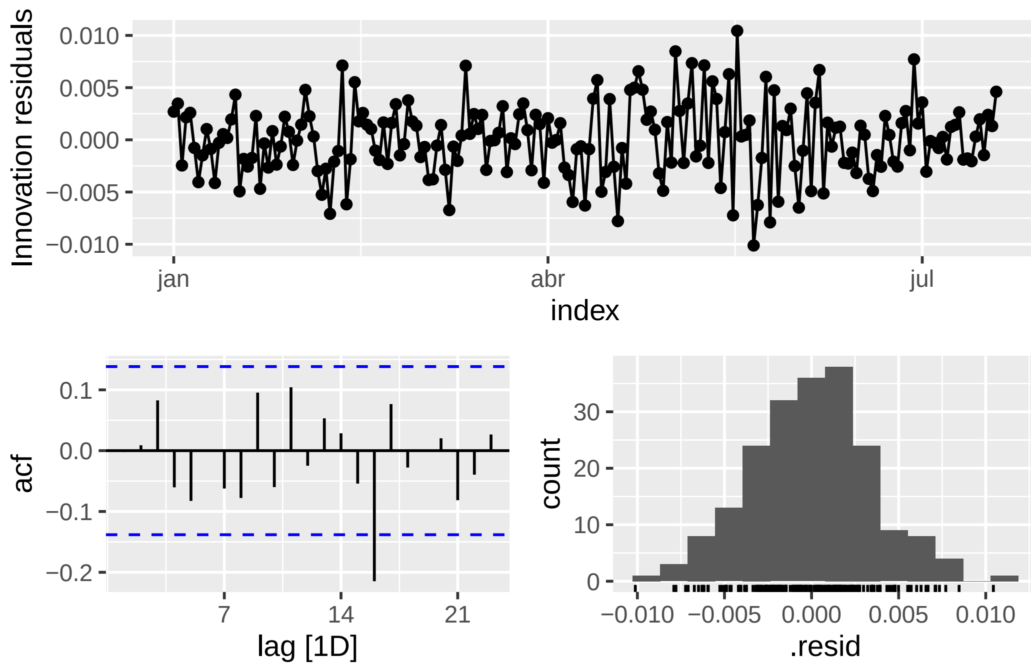
```
ARIMA(0,0,0)(0,0,0)[7]+c    -1589.386363
```

```
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
```

Antes de realizar predições, verificamos se os resíduos são ruído branco, o que indica que o modelo foi bem especificado.

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



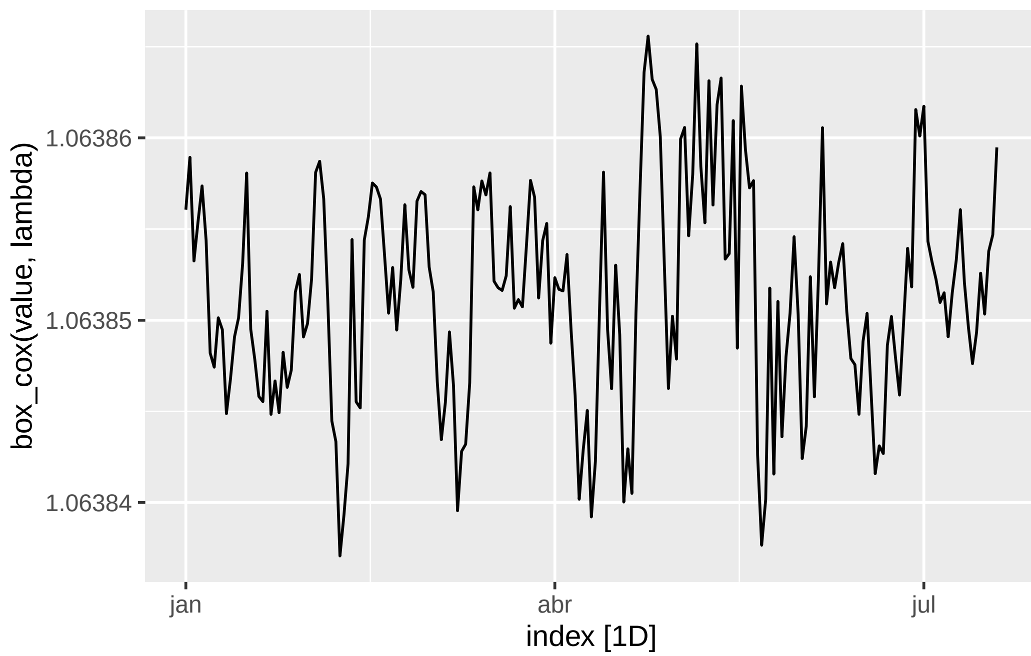

```
augment(data_fit) |>
  dplyr::filter(.model == "ar1") |>
  features(.innov, ljung_box, lag = 12, dof = 2)
```

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

A MODELAGEM BOX-JENKINS: SÉRIE II

A série não parece estável. Recomendável transformação para estabilização da variância.

```
lambda = data |>
  features(value, features = guerrero) |>
  (\(x) x[["lambda_guerrero"]])()

data |>
    autoplot(box_cox(value, lambda))
```


1.06386
1.06385
1.06384
box_cox(value, lambda)
jan
abr
jul
index [1D]