Introdução à Inferência Bayesiana

Démerson André Polli

ENAP - 16/01/2020

Petrobras (PETR4)

```
petr4 = c(24.16,24.49,24.72,24.85,24.85,24.85,24.46,24.75,24.86,24.11,24.11,24.11,
          24.11,24.69,24.93,24.90,25.11,25.33,25.33,25.33,25.35,24.81,24.42,24.46,24.18,
          24.18,24.18,25.03,25.35,26.23,26.12,26.04,26.04,26.04,26.65,26.32,26.67,26.40,
          25.99,25.99,25.99,25.87,26.36,26.34,25.99,26.04,26.04,26.04,26.04,26.04,26.05,
          25.96,26.97,26.97,26.97,26.75,27.25,27.37,27.46,28.02,28.02,28.02,28.42,28.19,
          27.80,26.28,26.61,26.61,26.61,27.86,26.61,27.31,27.31,27.25,27.25,27.25,27.53,
          26.80,27.71,28.01,28.47,28.47,28.47,28.38,28.01,27.25,25.14,25.24,25.24,25.24,
          26.00,26.03,26.86,26.71,26.71,26.71,26.71,26.94,26.86,27.08,26.52,26.64,26.64,
          26.64,26.38,26.01,26.01,26.13,26.05,26.05,26.05,25.64,26.64,26.11,25.97,25.21,
          25.21,25.21,25.30,25.19,24.59,24.02,24.84,24.84,24.84,25.78,25.66,25.22,25.47,
          25.62,25.62,25.62,26.16,25.87,25.53,24.94,25.37,25.37,25.37,25.58,25.24,25.65,
          26.12,26.02,26.02,26.02,26.51,26.21,26.53,26.42,26.46,26.46,26.46,26.80,26.86,
          27.61,27.61,27.58,27.58,27.58,26.85,27.01,26.58,26.76,26.61,26.61,26.61,26.18,
          26.48,26.74,26.75,26.99,26.99,26.99,27.40,27.40,27.72,27.85,27.51,27.51,27.51,
          27.17,27.05,26.85,26.79,26.84,26.84,26.84,26.86,26.70,26.25,25.52,25.75,25.75,
          25.75,25.62,25.46,24.99,25.89,24.94,24.94,24.94,25.26,24.99,25.70,25.65,25.04,
          25.04,25.04,25.35,24.49,23.81,23.50,23.62,23.62,23.62,23.61,25.01,24.79,23.86,
          23.55,23.55,23.55,23.92,24.17,25.06,25.06,24.86,24.86,24.86,25.16,25.81,25.94,
          26.06,26.47,26.47,26.47,26.63,26.41,26.59,26.42,27.58,27.58,27.58,27.21,26.75,
          26.82,26.53,27.01,27.01,27.01,26.80,26.87,27.22,27.18,27.08,27.08,27.08,27.04,
          26.26,26.28,26.05,25.72,25.72,25.72,25.57,26.06,26.28,26.79,26.84,26.84,26.84,
          27.12,27.45,27.18,27.12,27.29,27.29,27.29,28.08,28.45,27.83,28.75,29.09,29.09,
          29.09,29.31,29.56,29.87,29.91,29.84,29.84,29.84,29.14,29.20,30.37,29.50,29.93,
          29.93,29.93,29.67,29.55,28.96,28.74,28.74,28.74,28.74,28.44,29.50,29.50,29.63,
          29.38,29.38,29.38,28.85,28.99,29.18,28.81,28.73,28.73,28.73,28.64,29.31,29.70,
          29.99,29.86,29.86,29.86,30.08,30.04,30.61,29.63,29.07,29.07,29.07,29.50,30.18,
          30.25,29.91,30.14,30.14,30.14,30.55,30.55,30.55,30.52,30.18,30.18,30.18,30.70,
          30.70,30.70,30.45,30.81,30.81,30.81,30.69,30.50,30.40,30.27,30.33,30.33,30.33,
          30.00,29.55)
```

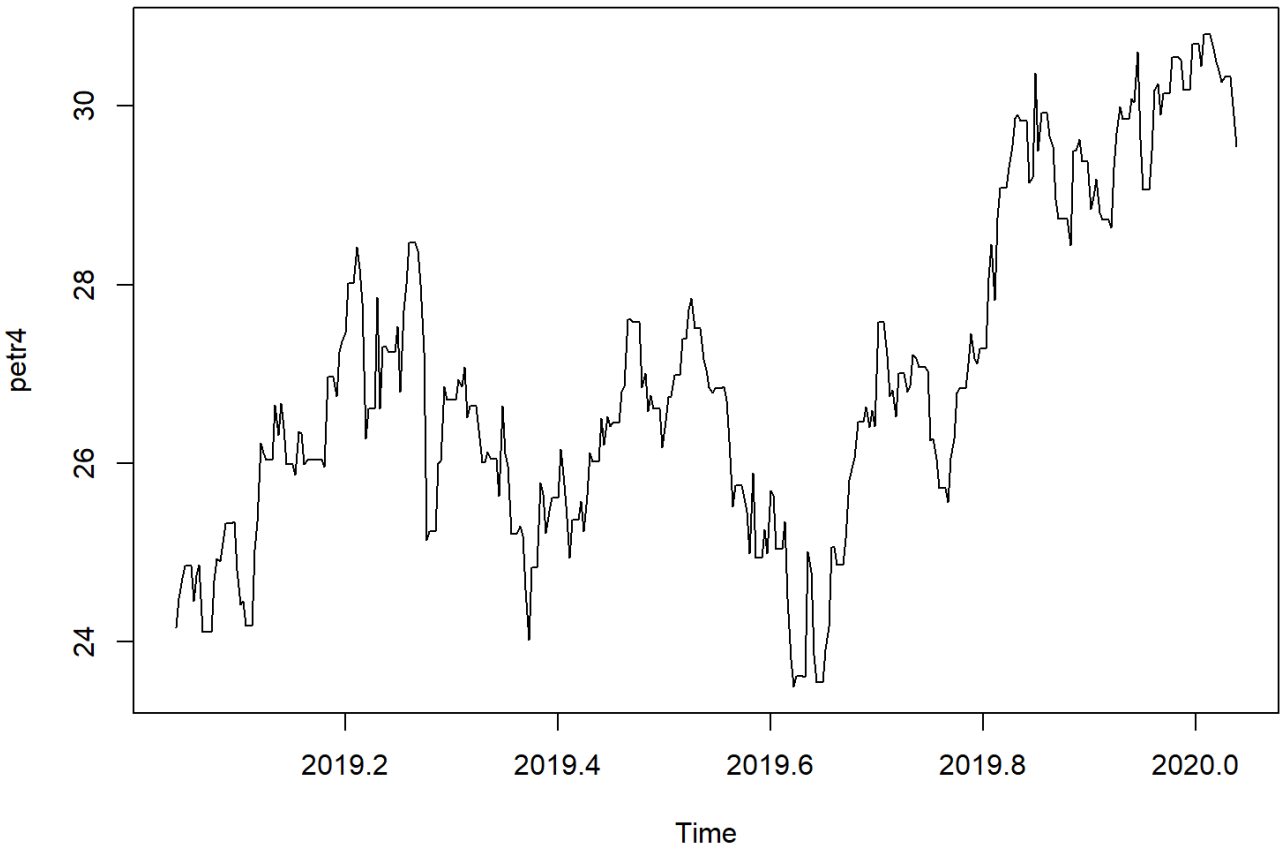
Petrobras (PETR4) - configurando a série

- Série de cotações da PETR4 (fechamento) entre 16 de janeiro de 2019 e 15 de janeiro de 2020.

```
inds <- seq(as.Date("2019-01-16"), as.Date("2020-01-15"), by = "day")

## Create a time series object
petr4 <- ts(petr4,
            start = c(2019, as.numeric(format(inds[1], "%j"))),
            frequency = 365)
```

Petrobras (PETR4) - série



Braskem (BRKM3)

```
brkm3 = c(47.02,45.46,44.01,44.01,44.01,44.40,46.35,47.36,47.51,47.51,47.51,47.51,46.87,
          47.17,50.49,50.33,50.33,50.33,50.33,51.11,52.09,52.36,50.14,50.14,50.14,50.14,
          49.87,51.21,51.62,53.47,52.59,52.59,52.59,53.53,53.54,54.02,53.14,53.65,53.65,
          53.65,53.74,54.99,53.08,52.48,53.54,53.54,53.54,53.54,53.54,53.50,52.09,52.08,
          52.08,52.08,52.69,52.62,52.62,53.10,53.10,53.10,53.10,52.96,51.62,51.62,49.95,
          49.95,49.95,49.95,50.43,51.12,48.68,51.16,51.63,51.63,51.63,50.14,48.67,48.67,
          48.09,46.83,46.83,46.83,47.22,47.69,47.43,47.70,49.04,49.04,49.04,46.97,46.25,
          46.19,46.34,46.34,46.34,46.34,45.79,45.27,44.79,44.93,44.64,44.64,44.64,45.74,
          46.61,46.61,44.72,44.67,44.67,44.67,43.44,42.32,40.90,39.07,38.75,38.75,38.75,
          36.51,37.00,37.67,36.02,35.44,35.44,35.44,38.65,40.69,39.91,40.11,39.62,39.62,
          39.62,38.56,38.94,38.58,38.06,38.94,38.94,38.94,37.90,33.56,33.76,33.34,32.83,
          32.83,32.83,32.67,33.10,33.10,33.94,34.13,34.13,34.13,33.78,34.08,34.08,34.08,
          34.75,34.75,34.75,34.87,34.26,34.08,33.59,34.07,34.07,34.07,34.40,34.18,34.40,
          35.16,36.46,36.46,36.46,36.54,36.54,36.02,36.07,36.41,36.41,36.41,36.13,35.29,
          35.24,34.95,34.90,34.90,34.90,34.75,35.34,35.53,34.57,34.17,34.17,34.17,33.68,
          34.06,33.92,33.15,32.34,32.34,32.34,32.03,32.51,32.51,31.97,31.79,31.79,31.79,
          31.47,31.41,30.76,29.55,29.40,29.40,29.40,29.69,28.62,27.84,27.60,27.62,27.62,
          27.62,27.06,27.16,27.90,28.81,28.82,28.82,28.82,29.08,28.77,29.95,30.32,30.57,
          30.57,30.57,31.25,30.72,30.81,30.22,30.07,30.07,30.07,29.69,30.37,30.71,31.10,
          31.37,31.37,31.37,32.43,32.09,31.54,32.13,32.04,32.04,32.04,32.76,32.03,30.84,
          30.91,30.98,30.98,30.98,29.75,29.00,28.75,29.50,29.78,29.78,29.78,30.10,30.44,
          29.81,29.81,29.50,29.50,29.50,29.15,29.30,29.11,29.15,28.69,28.69,28.69,29.00,
          28.94,29.19,28.88,28.94,28.94,28.94,29.50,29.36,30.16,30.10,30.51,30.51,30.51,
          31.81,31.02,30.50,30.49,30.49,30.49,30.49,29.50,28.90,28.90,29.00,29.35,29.35,
          29.35,29.10,29.00,29.05,29.00,28.61,28.61,28.61,28.70,28.99,29.20,28.86,29.50,
          29.50,29.50,29.93,29.60,29.62,29.72,29.20,29.20,29.20,31.49,30.15,30.17,30.00,
          30.00,30.00,30.00,30.10,30.10,30.10,30.60,31.38,31.38,31.38,31.54,31.54,31.54,
          32.75,33.46,33.46,33.46,33.87,34.09,34.99,36.50,37.85,37.85,37.85,36.10,35.82,
          35.00)
```

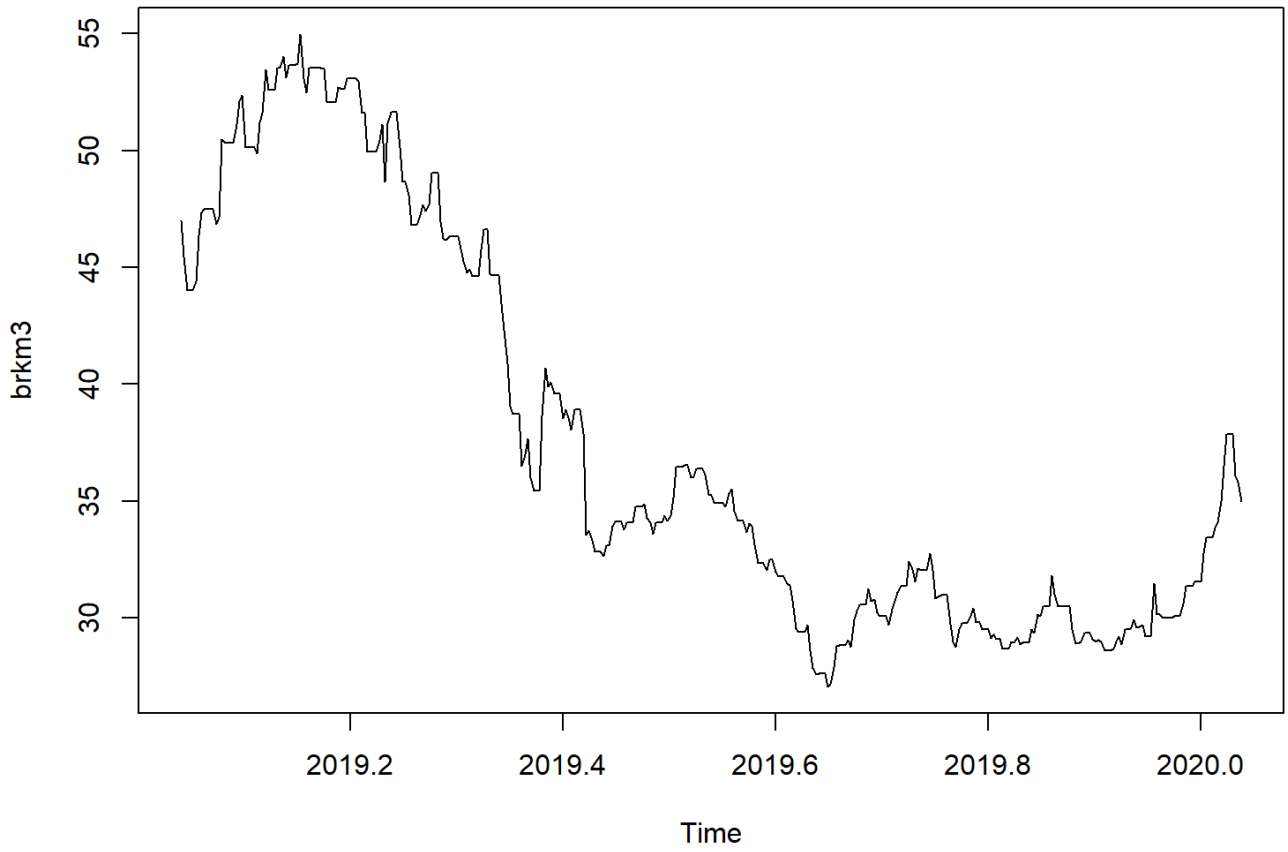
Petrobras (BRKM3) - configurando a série

- Série de cotações da BRKM3 (fechamento) entre 16 de janeiro de 2019 e 15 de janeiro de 2020.

```
inds <- seq(as.Date("2019-01-16"), as.Date("2020-01-15"), by = "day")

## Create a time series object
brkm3 <- ts(brkm3,
            start = c(2019, as.numeric(format(inds[1], "%j"))),
            frequency = 365)
```

Braskem (BRKM3) - série



IBOV

```
ibov = c( 94.39, 95.35, 96.10, 96.10, 96.10, 96.01, 95.10, 96.56, 97.68, 97.68, 97.68,
          97.68, 95.44, 95.64, 97.00, 97.39, 97.86, 97.86, 97.86, 98.59, 98.31, 94.64,
          94.11, 95.34, 95.34, 95.34, 94.41, 96.17, 96.12, 98.02, 97.53, 97.53, 97.53,
          96.51, 97.66, 96.55, 96.93, 97.89, 97.89, 97.89, 97.24, 97.60, 97.31, 95.58,
          94.60, 94.60, 94.60, 94.60, 94.60, 94.22, 94.34, 95.37, 95.37, 95.37, 98.03,
          97.83, 98.90, 98.61, 99.14, 99.14, 99.14, 99.99, 99.59, 98.04, 96.73, 93.74,
          93.74, 93.74, 93.66, 95.31, 91.90, 94.39, 95.42, 95.42, 95.42, 96.05, 95.39,
          94.49, 96.31, 97.11, 97.11, 97.11, 97.37, 96.29, 95.95, 94.82, 92.88, 92.88,
          92.88, 93.08, 94.33, 93.29, 94.58, 94.58, 94.58, 94.58, 94.59, 95.92, 95.05,
          96.39, 96.24, 96.24, 96.24, 96.19, 96.35, 96.35, 95.53, 96.01, 96.01, 96.01,
          95.01, 94.02, 95.60, 94.81, 94.26, 94.26, 94.26, 91.73, 92.09, 91.62, 90.02,
          89.99, 89.99, 89.99, 91.95, 94.49, 94.36, 93.91, 93.63, 93.63, 93.63, 94.86,
          96.39, 96.57, 97.46, 97.03, 97.03, 97.03, 97.02, 97.38, 96.00, 97.21, 97.82,
          97.82, 97.82, 97.73, 98.96, 98.32, 98.77, 98.04, 98.04, 98.04, 97.62, 99.40,
         100.30,100.30,102.01,102.01,102.01,102.06,100.09,100.69,100.72,100.97,100.97,
         100.97,101.34,100.61,102.04,103.64,104.09,104.09,104.09,104.53,104.53,105.82,
         105.14,103.90,103.90,103.90,103.80,103.78,103.86,104.72,103.45,103.45,103.45,
         103.95,103.70,104.12,102.66,102.82,102.82,102.82,103.48,102.93,101.81,102.13,
         102.67,102.67,102.67,100.10,102.16,102.78,104.11,103.91,103.91,103.91,101.92,
         103.46,100.26, 99.06, 99.81, 99.81, 99.81, 99.47, 99.22,101.20,100.01, 97.67,
          97.67, 97.67, 96.43, 97.28, 98.19,100.52,101.14,101.14,101.14,100.63, 99.68,
         100.94,102.25,102.94,102.94,102.94,103.18,103.03,103.45,104.33,103.50,103.50,
         103.50,103.26,104.62,104.53,104.34,104.82,104.82,104.82,104.64,103.88,104.48,
         105.32,105.08,105.08,105.08,104.75,104.05,101.03,101.52,102.55,102.55,102.55,
         100.57, 99.98,101.25,101.82,103.83,103.83,103.83,104.30,104.49,105.42,105.02,
         104.73,104.73,104.73,106.02,107.38,107.54,106.99,107.36,107.36,107.36,108.19,
         107.56,108.41,107.22,108.20,108.20,108.20,108.65,108.72,108.35,109.58,107.63,
         107.63,107.63,108.37,106.75,106.06,106.56,106.56,106.56,106.56,106.27,105.86,
         105.86,107.50,108.69,108.69,108.69,108.42,107.06,107.71,108.29,108.23,108.23,
         108.23,108.93,108.96,110.30,110.62,111.13,111.13,111.13,110.98,110.67,110.96,
         112.14,112.57,112.57,112.57,111.90,112.62,114.32,115.13,115.12,115.12,115.12,
         115.07,115.07,115.07,117.20,116.53,116.53,116.53,115.65,115.65,115.65,118.57,
         117.78,117.78,117.78,116.59,116.66,116.17,115.95,115.50,115.50,115.50,117.33,
         117.63,116.41)
```

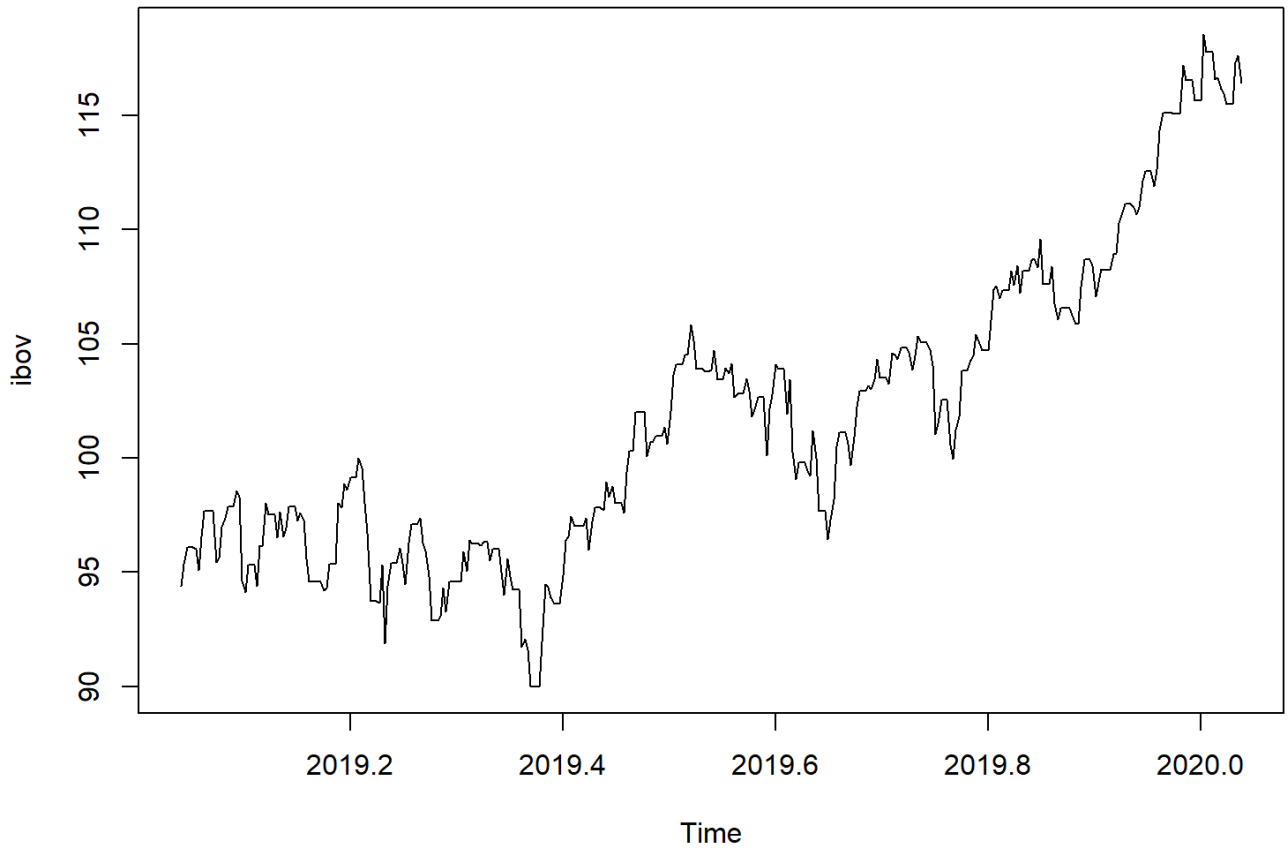
IBOV - configurando a série

- Série de IBOV entre 16 de janeiro de 2019 e 15 de janeiro de 2020.

```
inds <- seq(as.Date("2019-01-16"), as.Date("2020-01-15"), by = "day")

## Create a time series object
ibov <- ts(ibov,
           start = c(2019, as.numeric(format(inds[1], "%j"))),
           frequency = 365)
```

IBOVESPA - série



Modelos de Espaço de Estados (I)

O modelo *Dynamic Linear Model* pode ser definido na forma

$$\boldsymbol{y}_t = \boldsymbol{F}_t\boldsymbol{\theta}_t + \boldsymbol{\nu}_t, \quad \boldsymbol{\nu}_t \sim N(\boldsymbol{0}, \boldsymbol{V}_t)$$

$$\boldsymbol{\theta}_t = \boldsymbol{G}_t\boldsymbol{\theta}_{t-1} + \boldsymbol{\omega}_t, \quad \boldsymbol{\omega}_t \sim N(\boldsymbol{0}, \boldsymbol{W}_t)$$

$$\boldsymbol{\theta}_0 \sim N(m_0, C_0)$$

- Estes modelos estão implementados no pacote `dlm` no `R`.
- O pacote `dlm` permite o ajuste por *máxima verossimilhança* ou *inferência bayesiana* de modelos de espaço de estados.
  - *`dLmModARMA()`: processo ARMA*
  - *`dLmModPoly()`: modelos polinomiais*
  - *`dLmModReg()`: regressão linear*
  - *`dLmModSeas()`: modelo sazonal (periódico)*
  - *`dLmModTrig()`: modelo sazonal (trigonométrico)*

Modelo de Espaço de estados (II)

```
if(!require("dlm")) install.packages("dlm")
if(!require("numDeriv")) install.packages("numDeriv")

library(dlm)
library(numDeriv)
```

Modelo de Espaço de estados (III)

```
build = function(theta) {
  dlmModPoly(order = 1, dV = theta[1], dW = theta[2])
}

(fit = dlmMLE(petr4, parm = c(100, 2), build, lower = c(rep(1e-4, 2))))
```

```
## $par
## [1] 0.009248528 0.136705220
##
## $value
## [1] -149.6691
##
## $counts
## function gradient
##       60       60
##
## $convergence
## [1] 0
##
## $message
## [1] "CONVERGENCE: REL_REDUCTION_OF_F <= FACTR*EPSMCH"
```

Modelo de Espaço de estados (IV)

```
model <- build(fit$par)
drop(V(model))
```

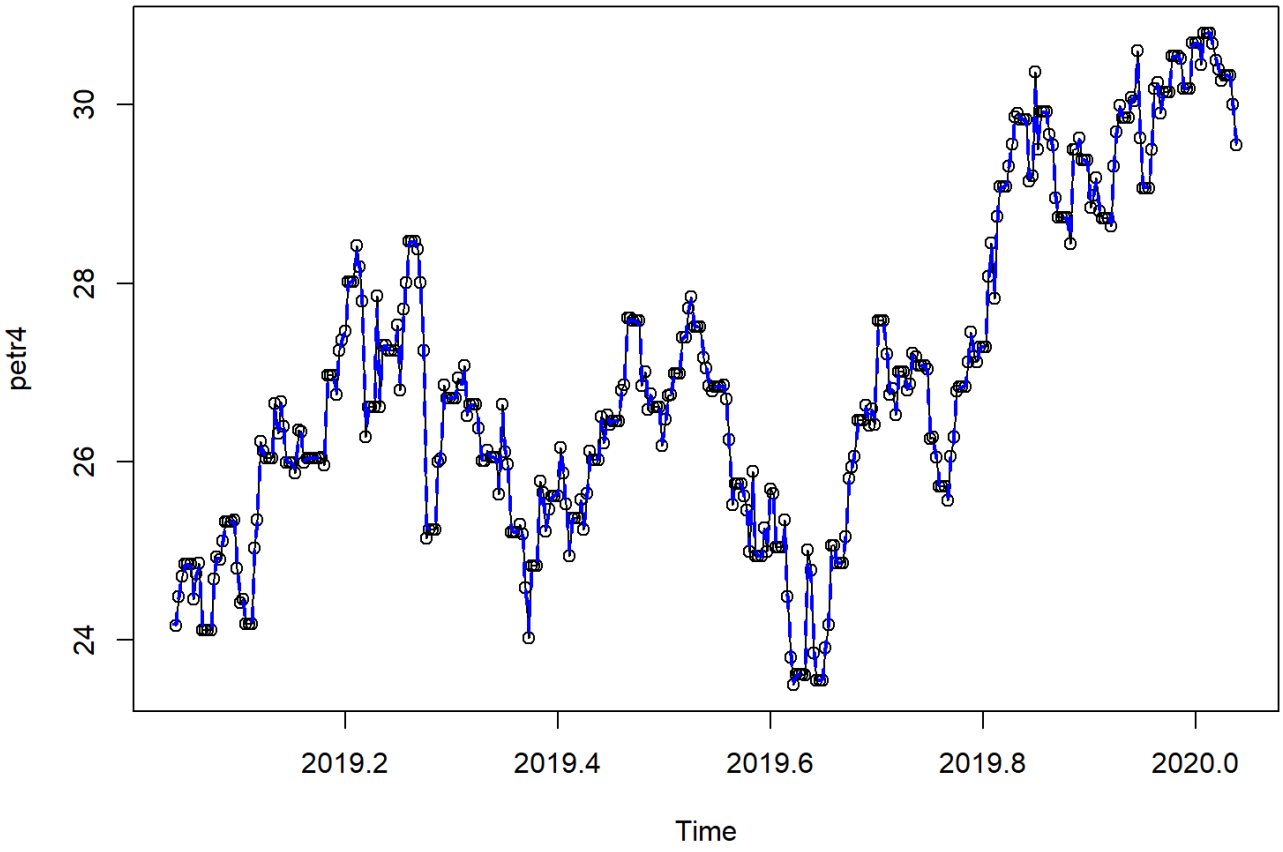
```
## [1] 0.009248528
```

```
drop(W(model))
```

```
## [1] 0.1367052
```

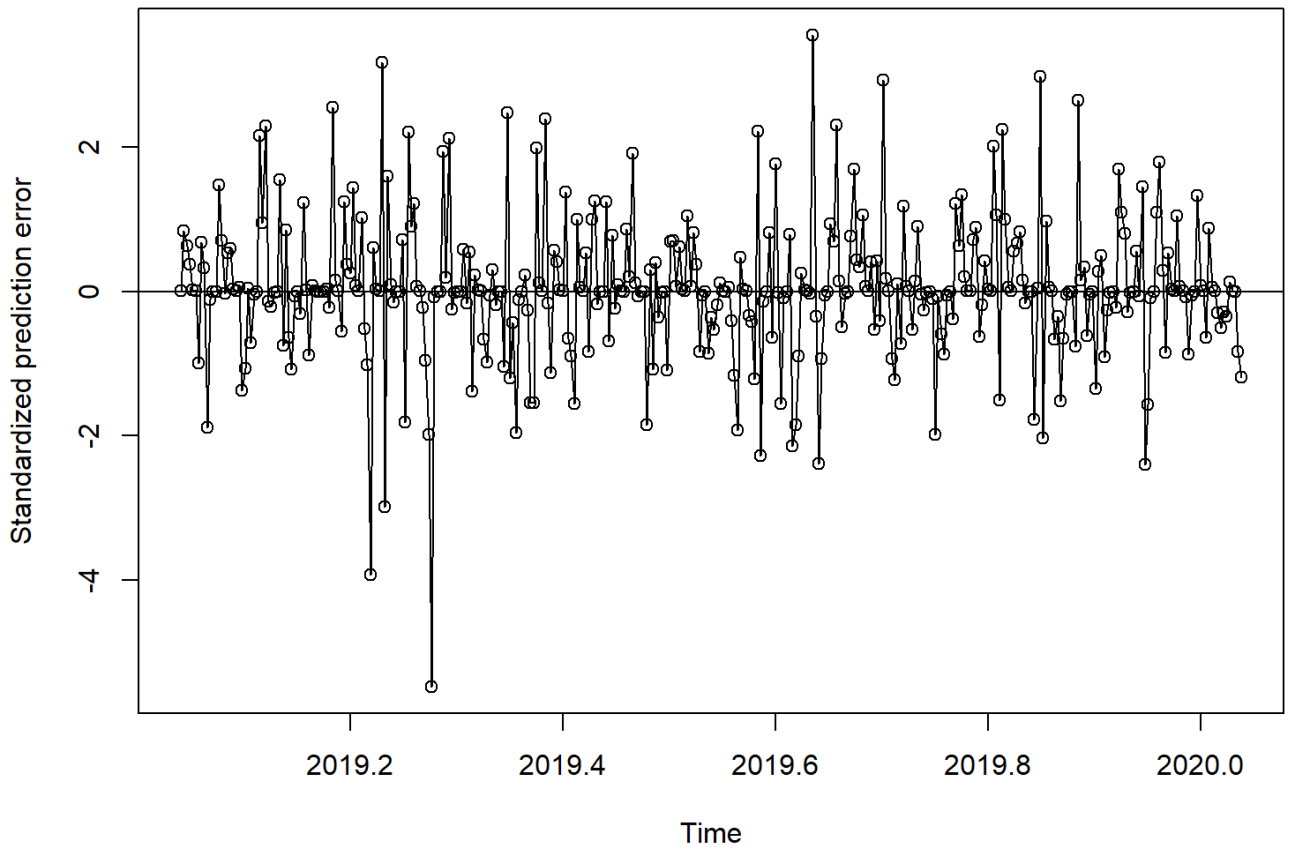
Modelo de Espaço de estados (V)

```
smoothed <- dlmSmooth(petr4, model)
plot(petr4, type = "o")
lines(smoothed$s, lty = 2, col = "blue", lwd= 2)
```



Modelo de Espaço de estados (V)

```
filtered <- dlmFilter(petr4, model)
plot(residuals(filtered, sd = FALSE), type = "o",
     ylab = "Standardized prediction error")
abline(h = 0)
```



Modelo de Espaço de estados (VI)

```
build = function(theta) {
  dlmModPoly(order = 2, dV = theta[1], dW = theta[2:3])
}

(fit = dlmMLE(petr4, parm = c(100, 2, 2), build, lower = c(rep(1e-4, 3))))
```

```
## $par
## [1] 0.008886788 0.137671306 0.000100000
##
## $value
## [1] -135.736
##
## $counts
## function gradient
##       88       88
##
## $convergence
## [1] 52
##
## $message
## [1] "ERROR: ABNORMAL_TERMINATION_IN_LNSRCH"
```

Modelo de Espaço de estados (VII)

```
model <- build(fit$par)
drop(V(model))
```

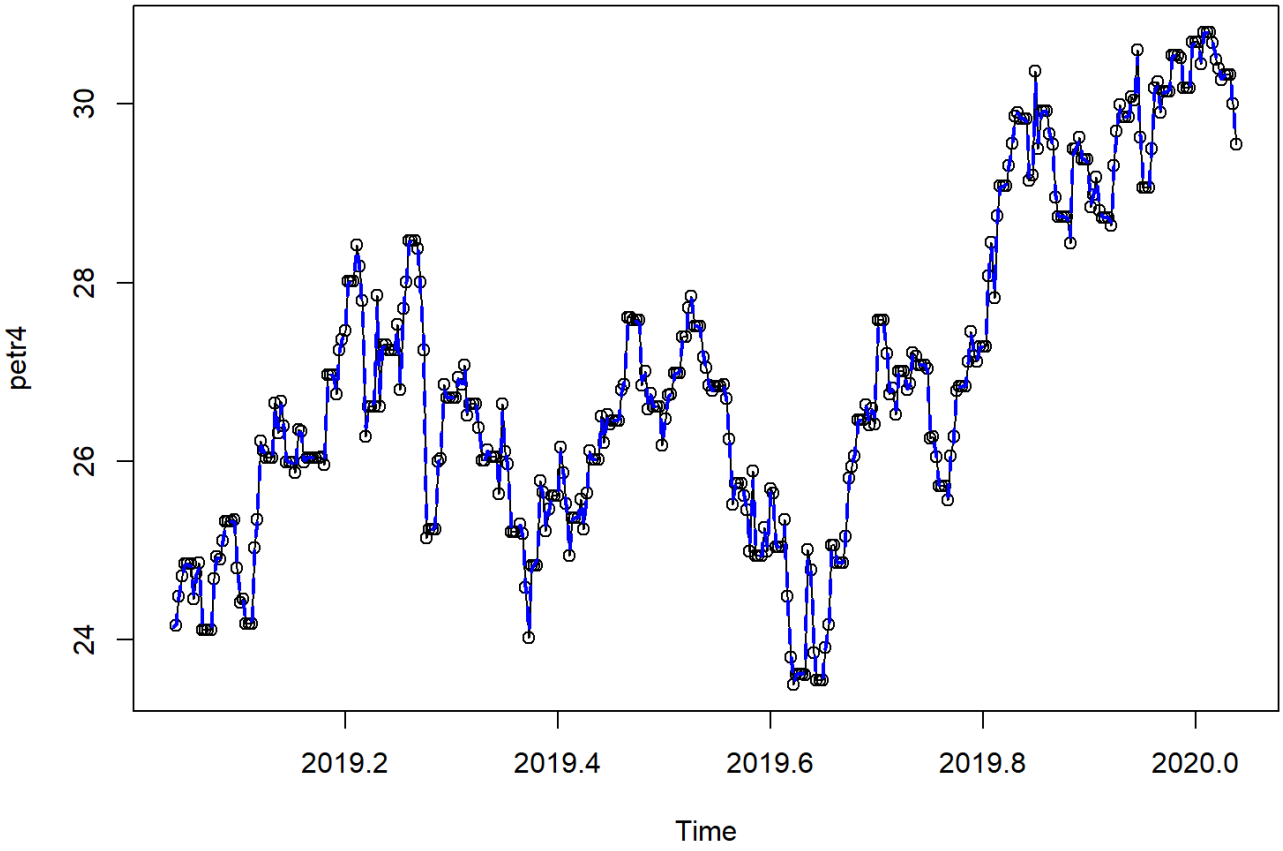
```
## [1] 0.008886788
```

```
drop(W(model))
```

```
##           [,1]  [,2]
## [1,] 0.1376713 0e+00
## [2,] 0.0000000 1e-04
```

Modelo de Espaço de estados (VIII)

```
smoothed <- dlmSmooth(petr4, model)
plot(petr4, type = "o")
lines(smoothed$s[,1], lty = 2, col = "blue", lwd= 2)
```



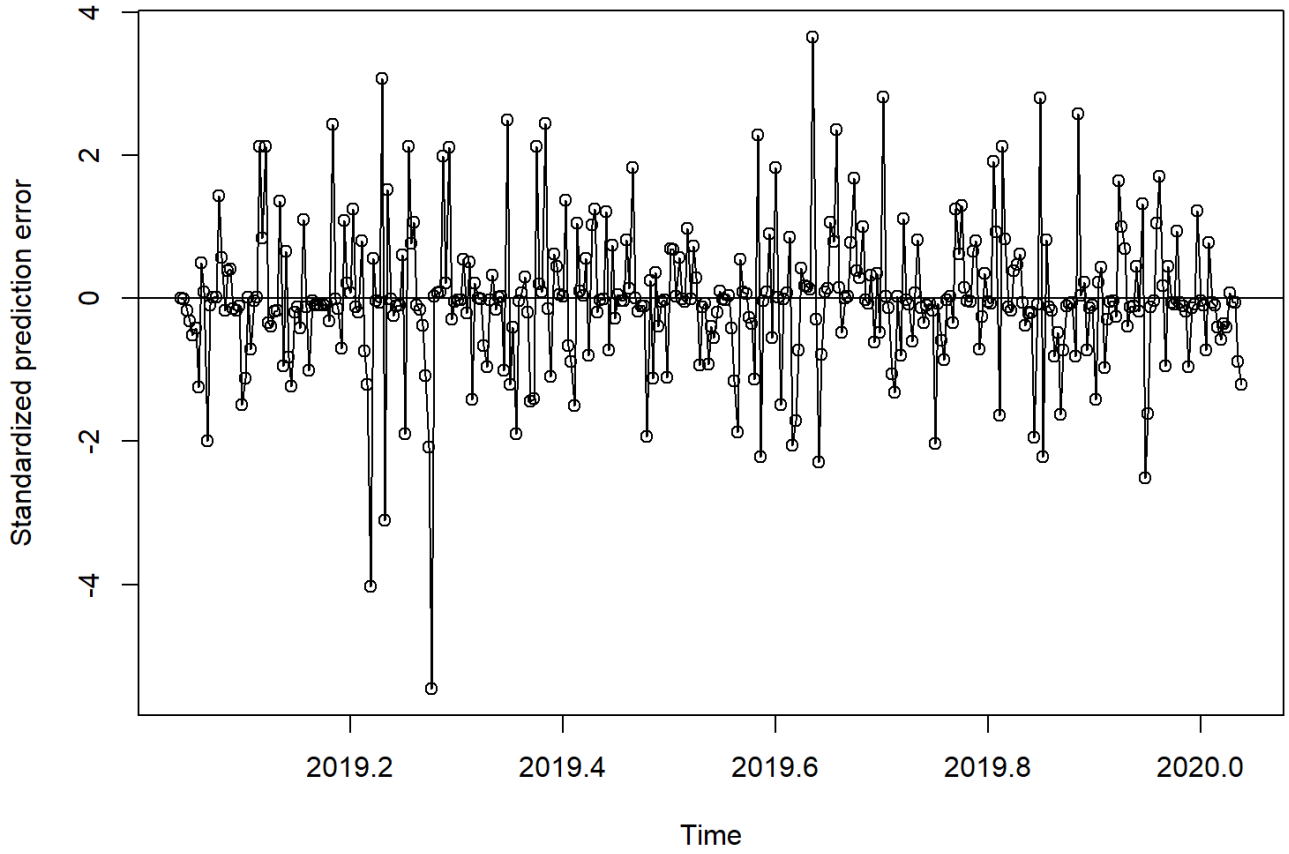
Modelo de Espaço de estados (IX)

```
build = function(theta) {
  dlmModPoly(order = 2, dV = theta[1], dW = theta[2:3]) +
    dlmModARMA(ar = theta[4:5], ma = theta[6:7], sigma2 = theta[8])
}

fit = dlmMLE(petr4, parm = c(100, 2, 2, .1, .1, .1, .1, 1.5), build,
             lower = c(rep(1e-4, 3), rep(-Inf, 4), 1e-4))
```

Modelo de Espaço de estados (X)

```
filtered <- dlmFilter(petr4, model)
plot(residuals(filtered, sd = FALSE), type = "o",
     ylab = "Standardized prediction error")
abline(h = 0)
```



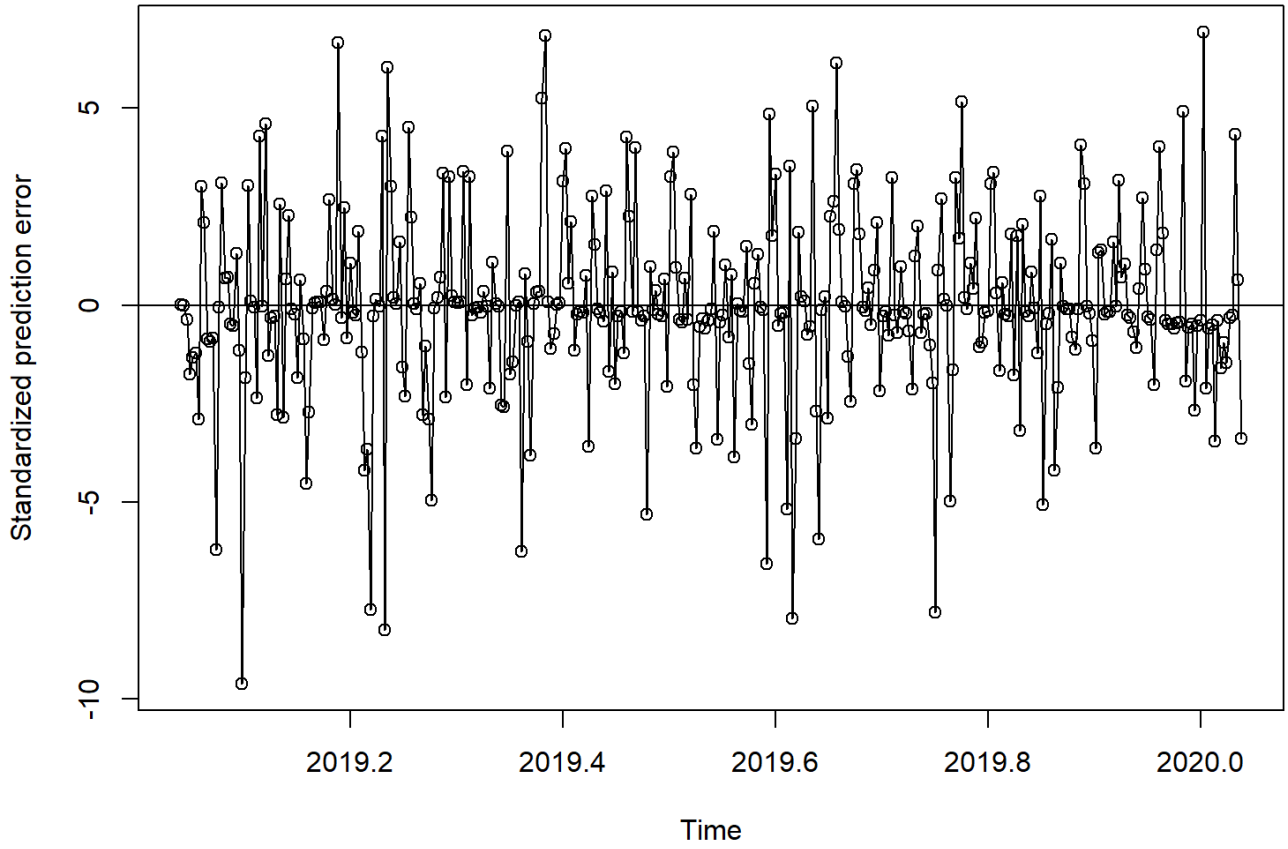
Modelo de Espaço de estados (XI)

```
build = function(theta) {
  dlmModPoly(order = 2, dV = theta[1], dW = theta[2:3]) +
    dlmModARMA(ar = theta[4:5], ma = theta[6:7], sigma2 = theta[8]) +
    dlmModReg(X = cbind(petr4, brkm3))
}

fit = dlmMLE(ibov, parm = c(100, 2, 2, .1, .1, .1, .1, 1.5), build,
             lower = c(rep(1e-4, 3), rep(-Inf, 4), 1e-4))
```

Modelo de Espaço de estados (XII)

```
filtered <- dlmFilter(ibov, model)
plot(residuals(filtered, sd = FALSE), type = "o",
     ylab = "Standardized prediction error")
abline(h = 0)
```



Modelo de Espaço de estados (XIII)

```
build = function(theta) {
  dlmModPoly(order = 1, dV = theta[1], dW = theta[2]) %+%
  dlmModPoly(order = 1, dV = theta[1], dW = theta[2])
}

fit = dlmMLE(cbind(petr4, brkm3), parm = c(100, 2, 100, 2), build,
             lower = rep(1e-4, 4))
```

Modelo de Espaço de estados (X)

```
fit
```

```
## $par
## [1]   0.0001000   0.3406173 100.0000000   2.0000000
##
## $value
## [1] -11.71285
##
## $counts
## function gradient
##       28       28
##
## $convergence
## [1] 0
##
## $message
## [1] "CONVERGENCE: REL_REDUCTION_OF_F <= FACTR*EPSMCH"
```

Próximos passos

- Como incluir séries de maior frequência como exógena?
- Como modelar séries multivariadas correlacionadas?
- Aplicação aos dados econométricos.