

UFPR: Análise de Séries Temporais - Lista 03

Luiz Paulo Tavares

2024-09-10

1) O que faz o Método Recursivo de J. Durbin publicado em 1960 no artigo *The fitting of time series models* na revista *Review of International Statistical Institute*? Detalhe o procedimento?

No artigo é apresentado um método recursivo para estimar os parâmetros de modelos autorregressivos de forma eficiente. O método recursivo de Durbin pode ser compreendido tomando algumas etapas dado um modelo AR(p):

$$Z_t = \phi_1 z_{t-1} + \phi_2 z_{t-2} + \dots + \phi_p z_{t-p} + \epsilon_t$$

Passo 1: seleção da ordem p. A seleção da ordem de defasagem p pode ser baseada em conhecimento prévio ou com base em algum critério de seleção como, por exemplo, Critério de Informação Akaike (AIC) ou o Critério de Informação Bayesiana (BIC).

Passo 2: ajustar o modelo para os primeiros p valores da série com um método tradicional (como o método de mínimos quadrados, por exemplo). A ideia é ter um ponto de partida para o processo recursivo.

Passo 3: calcular os resíduos. Para cada observação após p, ou seja, $t = p + 1, p + 2, \dots, n$:

$$\epsilon_t = z_t - (\phi_1 z_{t-1} + \phi_2 z_{t-2} + \dots + \phi_p z_{t-p})$$

A inovação do método de Durbin está na forma como os parâmetros $\phi_1, \phi_2, \dots, \phi_p$ são atualizados de maneira recursiva à medida que novos dados z_{t+1} são observados. O objetivo é minimizar os resíduos quadráticos ϵ_t^2 continuamente, sem a necessidade de recalculá-los para todos os dados anteriores.

A atualização dos parâmetros ϕ_1, \dots, ϕ_p ocorre de maneira recursiva à medida que novos dados são observados. O objetivo é minimizar iterativamente os resíduos quadráticos ϵ_t^2 , sem recalculá-los todo o modelo a cada nova observação.

Passo 4: Atualização

4.1. Geração do novo dado

Quando um novo valor y_{t+1} é observado, o modelo ajustado anteriormente é usado para prever \hat{y}_{t+1} com base nas observações passadas $y_t, y_{t-1}, \dots, y_{t-p+1}$.

4.2. Cálculo do resíduo atualizado

O resíduo ϵ_{t+1} é calculado como a diferença entre o valor observado y_{t+1} e o valor ajustado pelo modelo:

$$\epsilon_{t+1} = y_{t+1} - (\phi_1 y_t + \phi_2 y_{t-1} + \dots + \phi_p y_{t-p+1})$$

4.3. Correção dos parâmetros

A correção dos parâmetros ϕ_1, \dots, ϕ_p é feita utilizando o algoritmo de mínimos quadrados recursivo (Recursive Least Squares - RLS), que atualiza os parâmetros conforme novos resíduos são observados. A fórmula de atualização recursiva é dada por:

$$\phi_{t+1} = \phi_t + K_t \epsilon_{t+1}$$

onde: - $\phi_t = (\phi_1, \dots, \phi_p)$ representa o vetor dos parâmetros no tempo t , - K_t é a matriz de ganho que depende da covariância dos dados até o tempo t , - ϵ_{t+1} é o resíduo calculado no passo anterior.

A matriz de ganho K_t controla a magnitude da atualização dos parâmetros, permitindo que o modelo se ajuste gradualmente às novas observações sem se desviar excessivamente.

2) Escreva as equações que definem o Método Recursivo de Durbin

3) 10.15 - Dada a série temporal adiante que corresponde ao número de ações judiciais sofridas anualmente pela empresa X pede-se:

```
base::rm(list = ls())
pacman::p_load(tidyverse, DataExplorer, TSstudio, knitr,
               forecast, fable, fabletools)

# Valores fornecidos

n <- c(47, 29, 36, 39, 54, 69, 66, 35, 30, 25, 33, 25,
      30, 28, 43, 35, 38, 39, 51, 50, 50, 49, 46, 32,
      4, 9, 63, 59, 57, 66, 56, 68, 59, 84, 59, 77,
      65, 78, 82, 78, 67, 94, 88, 93, 87, 116, 126, 65,
      91, 119, 120, 114, 141, 110, 111, 137, 113, 116, 130, 71,
      82, 144, 122, 142, 117, 120, 140, 127, 124, 93, 98,
      69, 76, 138, 127, 126, 117, 122, 102)

# Criando o data frame

empresa <- data.frame(time = seq(as.Date("2004-01-01"), as.Date("2010-07-01"), by = "month"), n = n)

empresa_ts <- stats::ts(data = empresa$n, start = c(2004,1), end = c(2010,7), frequency = 12)
```

a) Descrição da série estimando sua média e o seu desvio padrão

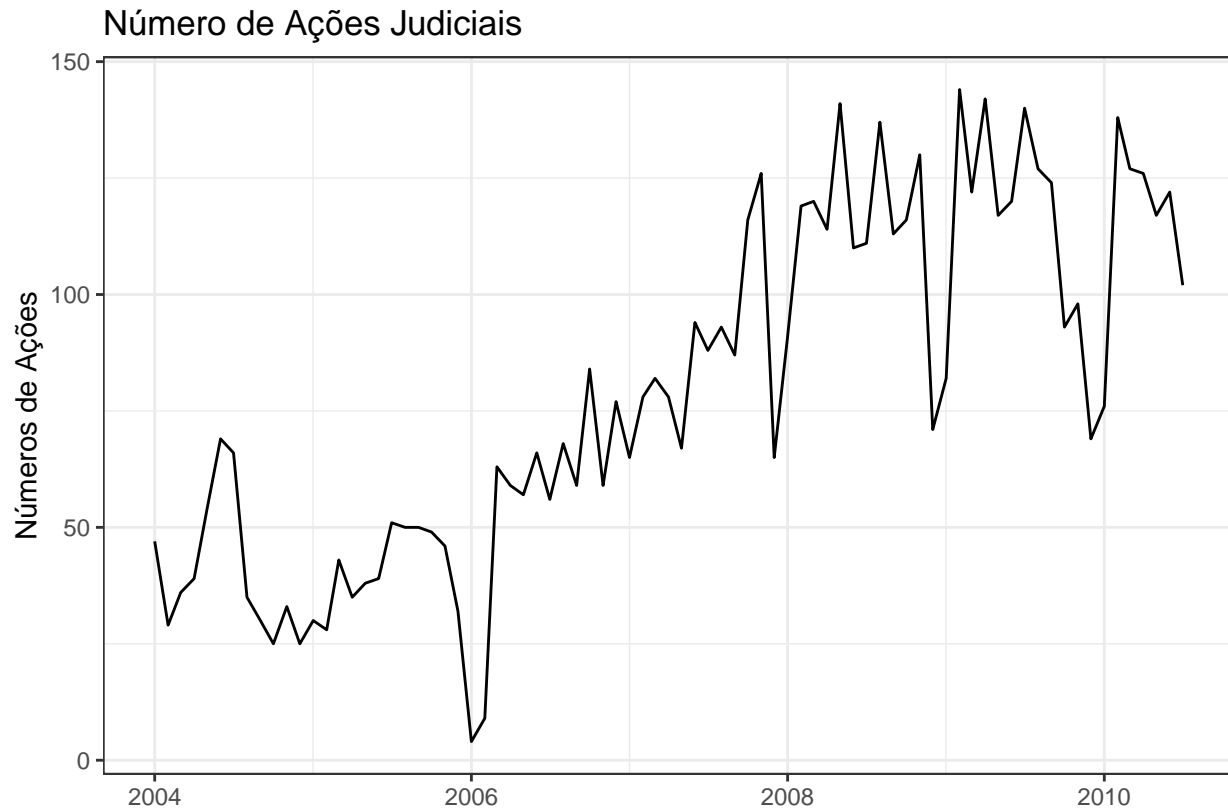
```
summario <- empresa %>%
  dplyr::group_by() %>%
  dplyr::summarise(média = mean(n),
                  desvio = sd(n))

knitr::kable(summario)
```

média	desvio
78.96203	37.39548

b) O gráfico horizontal da série.

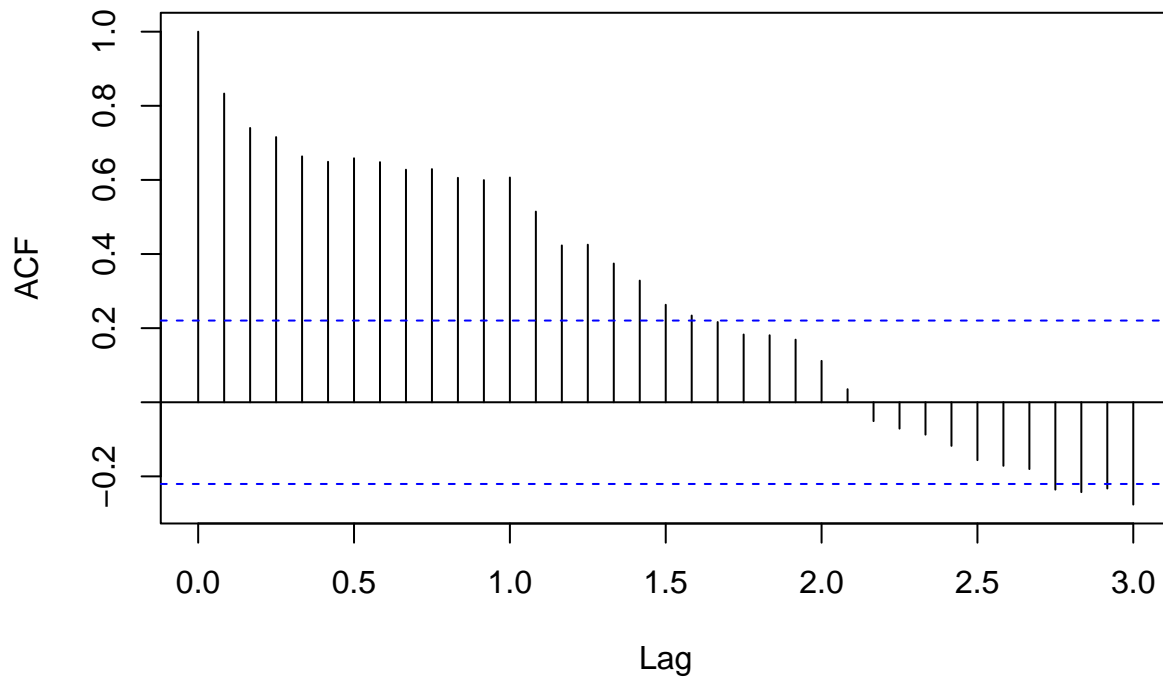
```
ggplot2::ggplot(data = empresa)+
  aes(y = n, x = time)+
  geom_line(col = "black")+
  labs(y = "Números de Ações",
       x = "",
       title = "Número de Ações Judiciais")+
  theme_bw()
```



c) O correlograma da FAC

```
stats::acf(empresa_ts, lag.max = 36, main = "Função de Autocorrelação (FAC)")
```

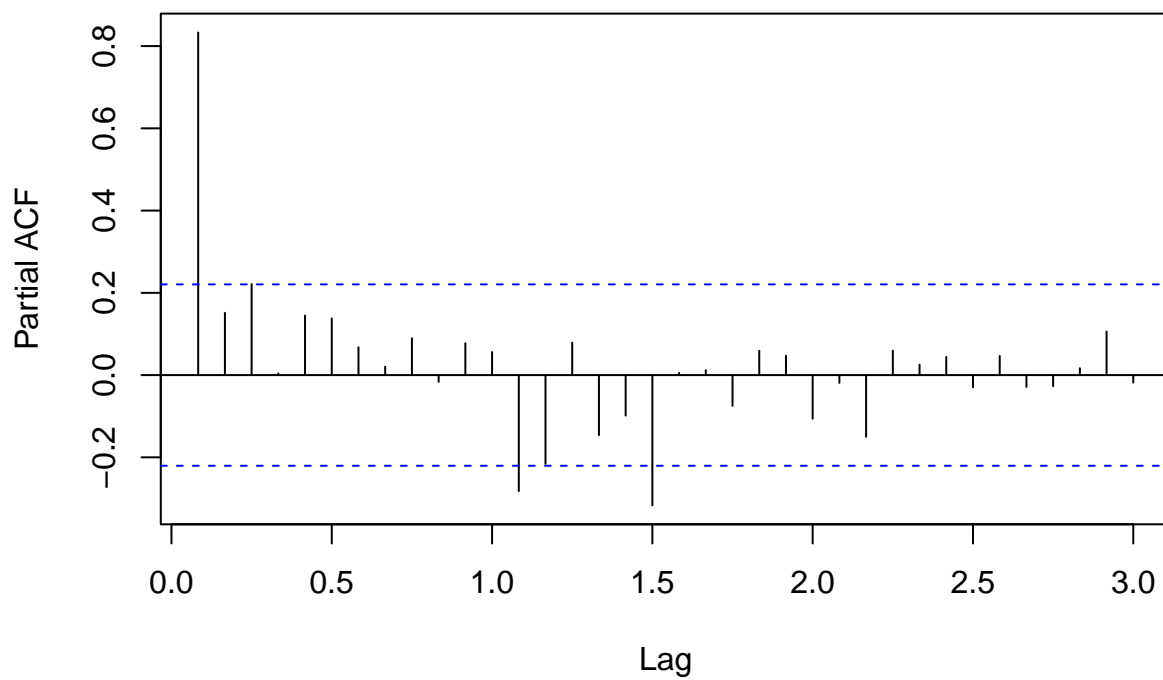
Função de Autocorrelação (FAC)



d) O correlograma da FACP.

```
stats::pacf(empresa_ts, lag.max = 36, main = "Função de Autocorrelação Parcial (FACP)")
```

Função de Autocorrelação Parcial (FACP)



e) Ajuste um modelo à série aplicando a Seleção Automática do Modelo e compare-o com outros possíveis modelos usando o Critério de Informação de Akaike -AIC

```
# Passar por um rápida análise exploratória antes da modelagem
```

```
# TSstudio::ts_decompose(empresa_ts, type = "both")  
# TSstudio::ts_seasonal(empresa_ts)  
# TSstudio::ts_heatmap(empresa_ts, padding = FALSE)  
# TSstudio::ts_lags(empresa_ts, lags = 1:12)
```

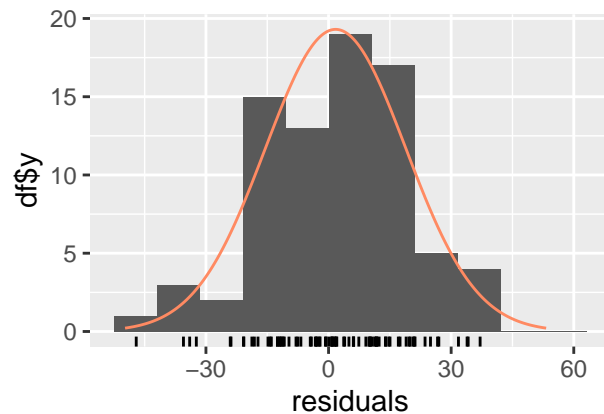
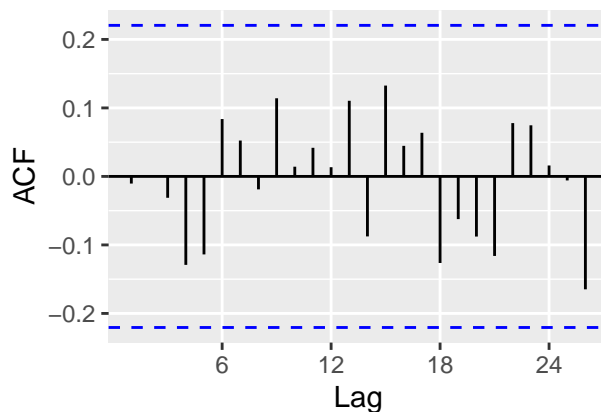
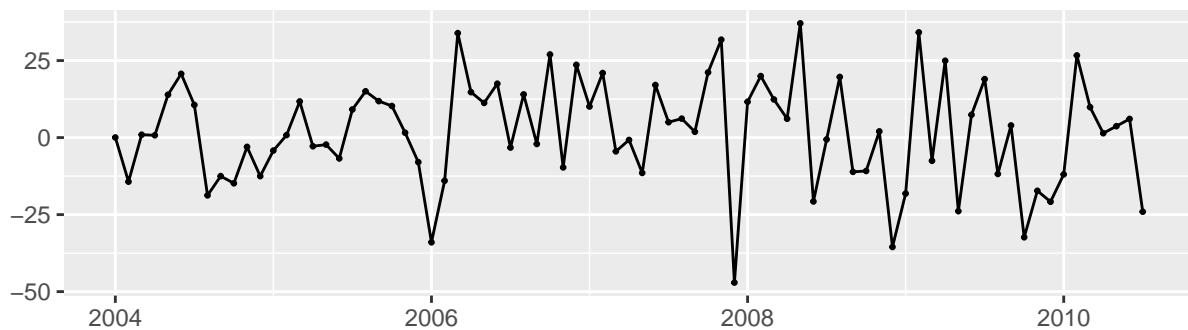
```
# Auto-ARIMA \*
```

```
# auto_arima = forecast::auto.arima(empresa$n)  
# checkresiduals(auto_arima)
```

```
# Auto-SARIMA \*
```

```
auto_sarima = forecast::auto.arima(empresa_ts)  
forecast::checkresiduals(auto_sarima)
```

Residuals from ARIMA(0,1,2)(1,0,0)[12]



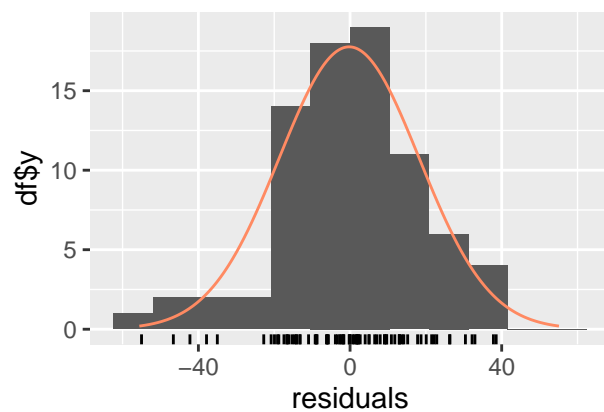
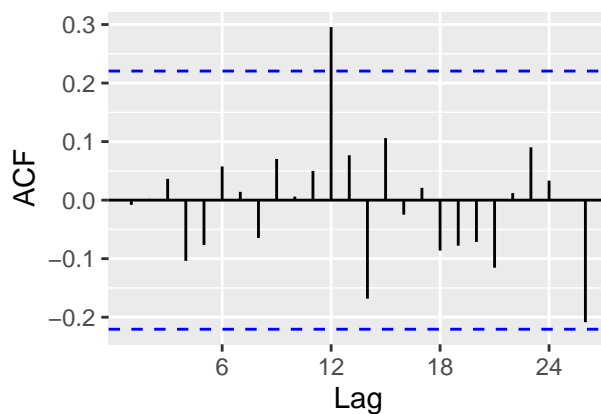
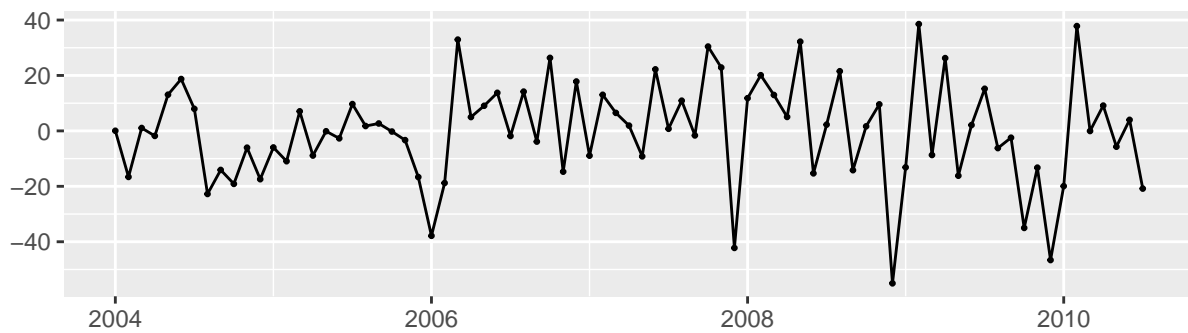
```
##  
## Ljung-Box test  
##  
## data: Residuals from ARIMA(0,1,2)(1,0,0)[12]  
## Q* = 8.8139, df = 13, p-value = 0.7868  
##  
## Model df: 3. Total lags used: 16
```

```
# Manual-ARIMA \*
```

```
manual_arima = forecast::Arima(empresa_ts,  
                               order = c(0, 1, 2), include.drift = TRUE)
```

```
forecast::checkresiduals(manual_arima)
```

Residuals from ARIMA(0,1,2) with drift



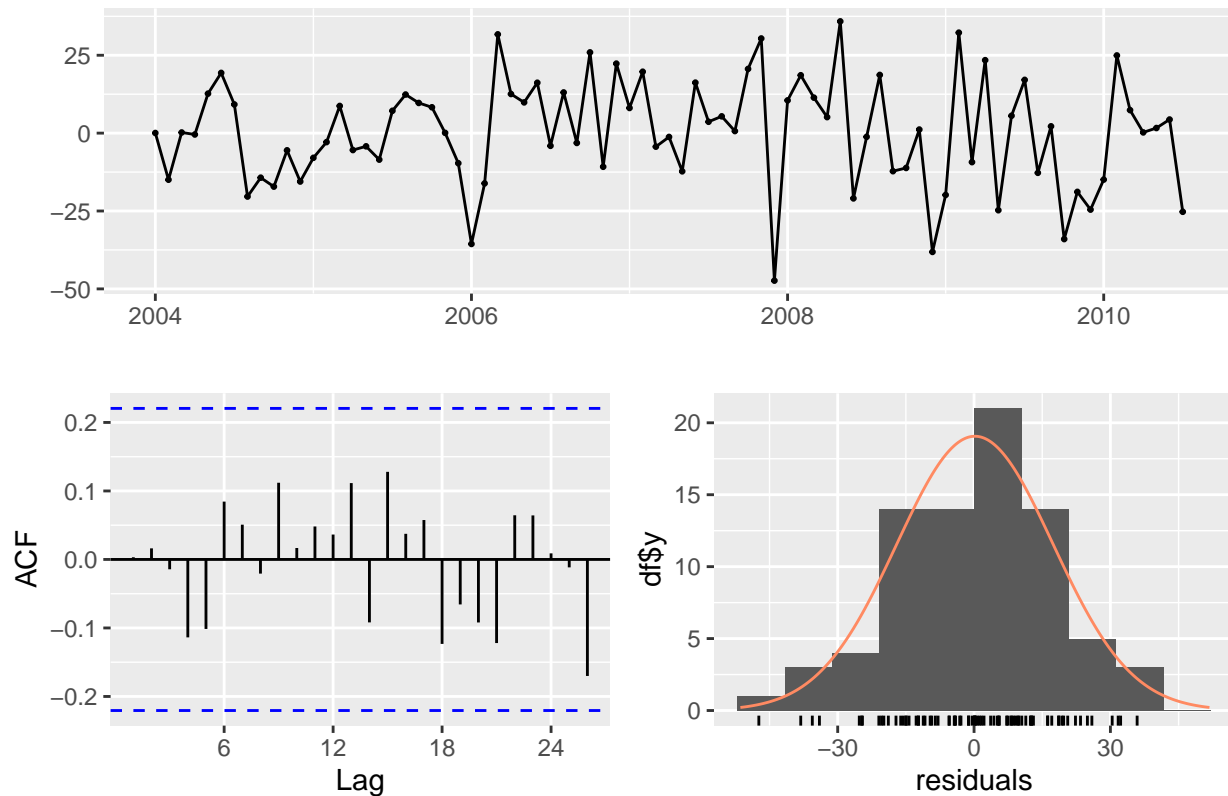
```
##  
## Ljung-Box test  
##  
## data: Residuals from ARIMA(0,1,2) with drift  
## Q* = 15.809, df = 14, p-value = 0.3252  
##  
## Model df: 2. Total lags used: 16
```

```
# Manual: SARIMA com drift \*
```

```
manual_sarima_drift <- forecast::Arima(empresa_ts,  
                                       order = c(0, 1, 2),  
                                       seasonal = list(order = c(1, 0, 0), period = 12),  
                                       include.drift = TRUE)
```

```
forecast::checkresiduals(manual_sarima_drift)
```

Residuals from ARIMA(0,1,2)(1,0,0)[12] with drift



```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(0,1,2)(1,0,0)[12] with drift
## Q* = 8.2566, df = 13, p-value = 0.8265
##
## Model df: 3.   Total lags used: 16
```

```
# Métricas dos modelos \*
```

```
models = data.frame(model = c(
  "SARIMA Automático",
  "ARIMA Manual",
  "SARIMA Manual"),
  AIC = c(
    auto_sarima[["aic"]],
    manual_arima[["aic"]],
    manual_sarima_drift[["aic"]]),
  AICc = c(
    auto_sarima[["aicc"]], NA,
    manual_sarima_drift[["aicc"]]),
  BIC = c(
    auto_sarima[["bic"]], NA,
    manual_sarima_drift[["bic"]]))
```

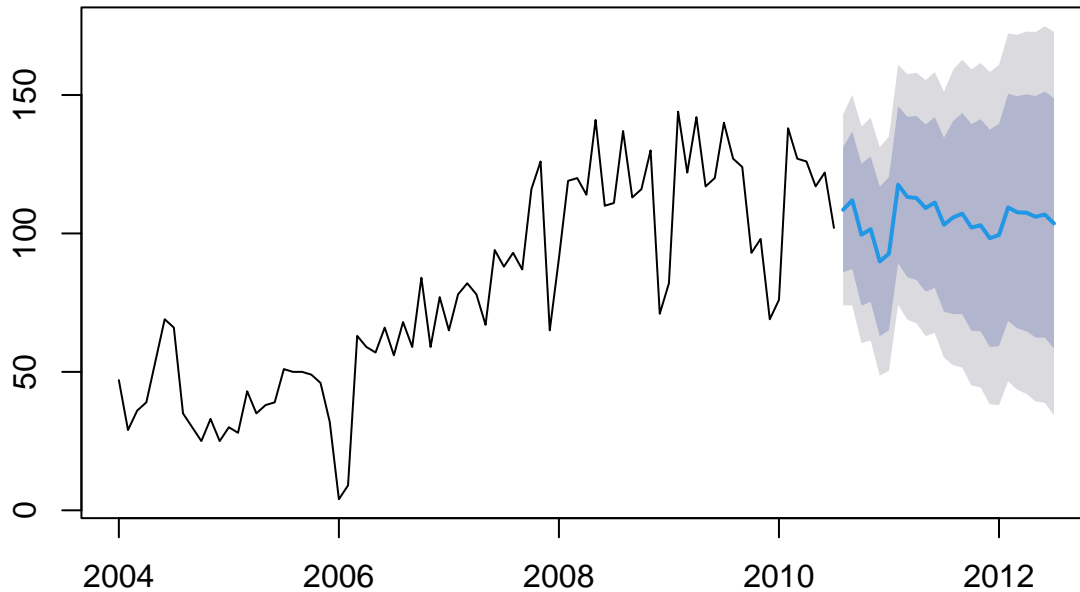
```
knitr::kable(models)
```

model	AIC	AICc	BIC
SARIMA Automático	676.4214	676.9694	685.8483
ARIMA Manual	684.8611	NA	NA
SARIMA Manual	677.6715	678.5048	689.4550

```
# model_select <- models %>% dplyr::filter(AIC == min(models['AIC'])) %>% print()
```

```
pred = forecast::forecast(auto_sarima, h = 24)
plot(pred)
```

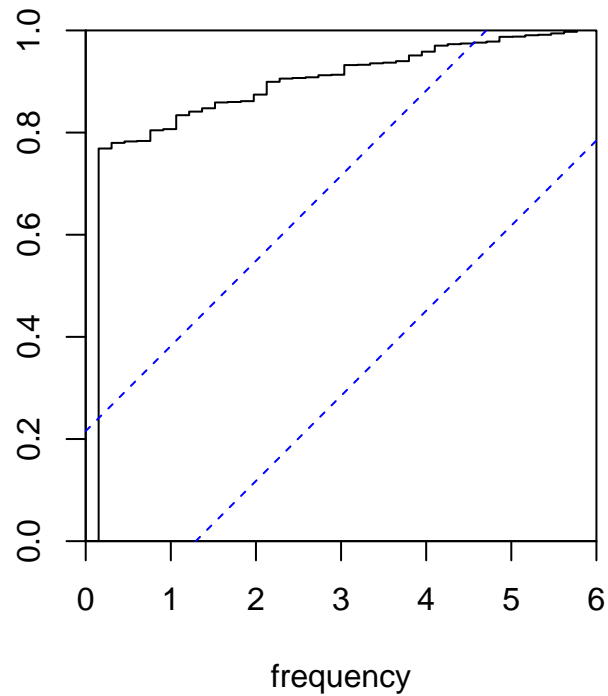
Forecasts from ARIMA(0,1,2)(1,0,0)[12]



f) O gráfico do Periodograma Integrado (Acumulado)

```
stats::cpgram(empresa_ts, main = "Cumulative periodogram of residuals")
```


Cumulative periodogram of residuals



4) 10.16 - O arquivo de séries temporais IBOVESPA (em EXCEL) foi postado nos arquivos TEAMS, espaço da disciplina. Então:

- a) Verifique se a série temporal IBOV tem parte sistemática para ser estimada aplicando o Gráfico do Periodograma Integrado;
- b) Faça o gráfico horizontal da série;