



DEPARTAMENTO DE ESTATÍSTICA

22 junho 2024

## **Lista 5**

Prof. Dr. Raul Yukihiro Matsushita

Aluno: Bruno Gondim Toledo

Matrícula: 15/0167636

Análise de séries temporais

1º/2024

## Questão 1

Calculando  $Y_t$

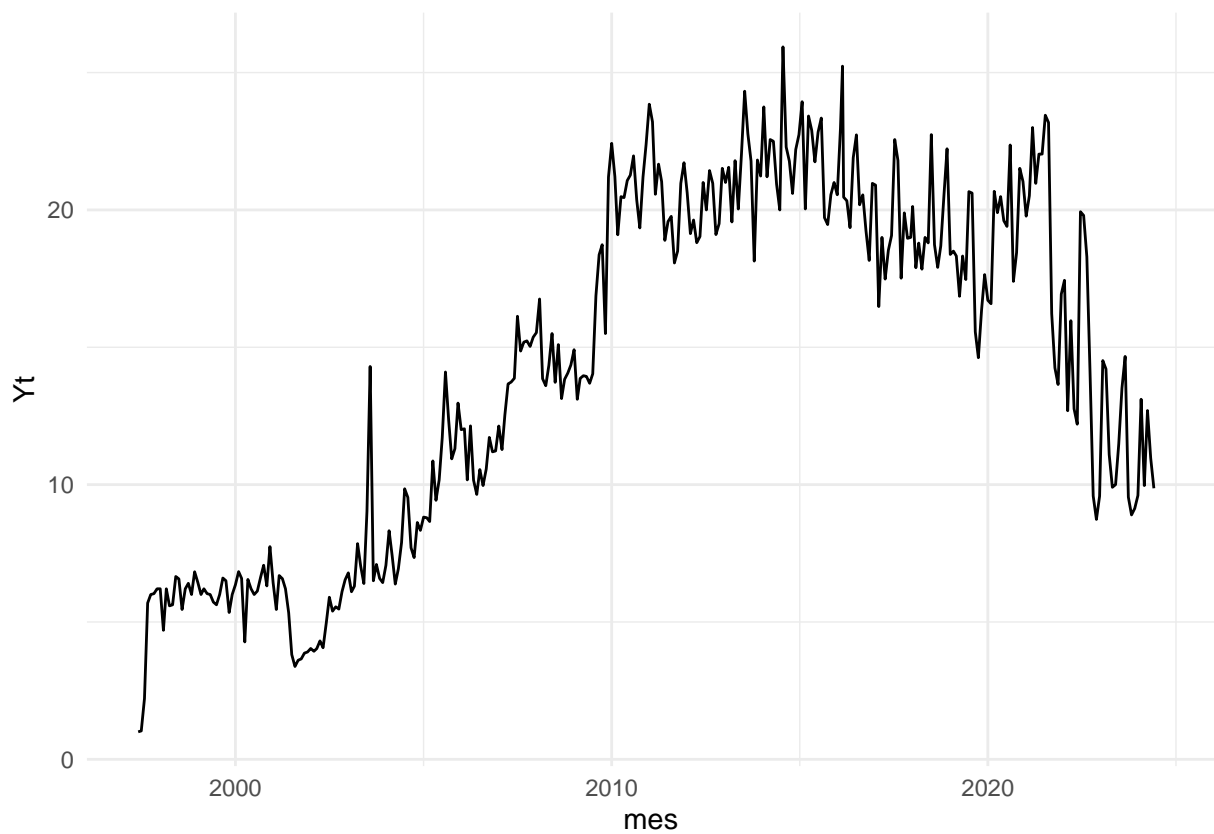
```
dados = df |> dplyr::select(mes, Energia, Dias) |>
  mutate(Yt = Energia/Dias)
kable(head(dados))
```

mes	Energia	Dias	Yt
1997-06-01	42	42	1.000000
1997-07-01	31	30	1.033333
1997-08-01	73	33	2.212121
1997-09-01	165	29	5.689655
1997-10-01	180	30	6.000000
1997-11-01	199	33	6.030303

## Questão 2

Visualizando a série

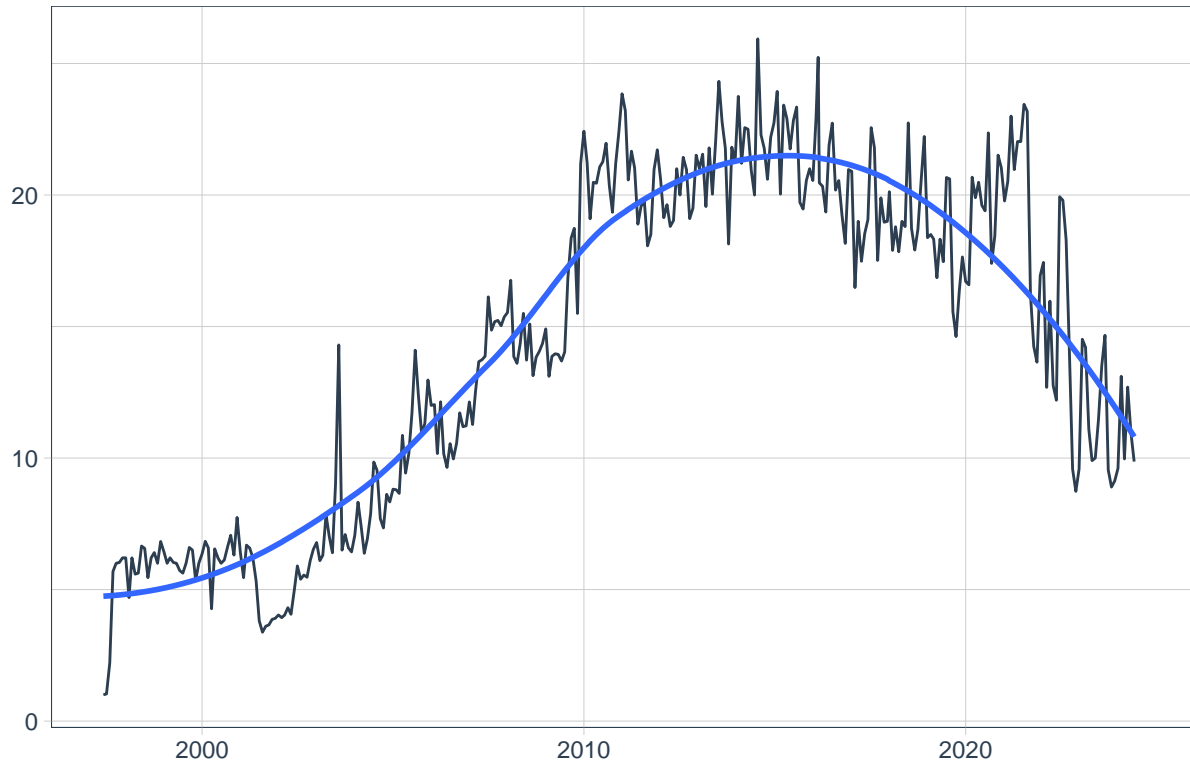
```
dados %>%
  ggplot() +
  geom_line(aes(x=mes,y=Yt)) + theme_minimal()
```



## Passando uma linha suavizada com auxílio do pacote *timetk* para verificação visual de tendência

```
dados %>%  
  plot_time_series(mes, Yt, .interactive = F, .smooth = T)
```

Time Series Plot



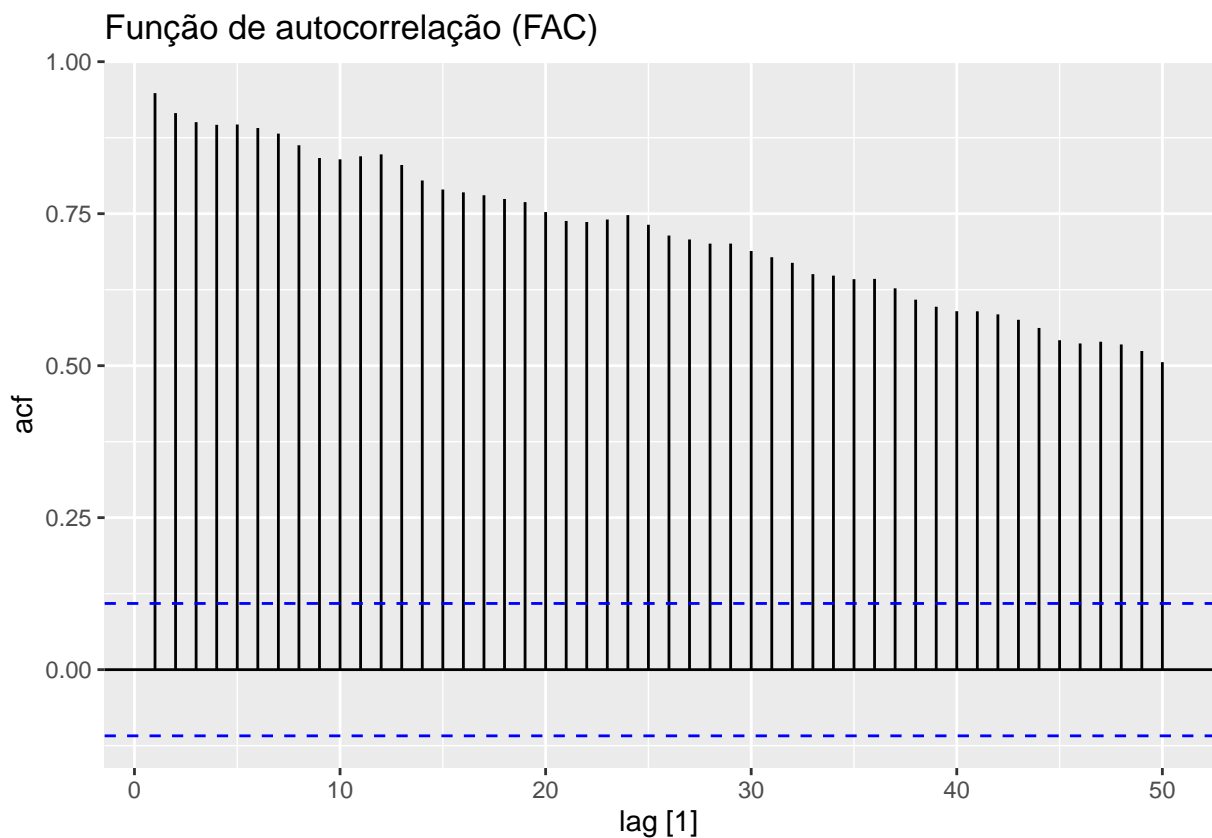
Com o auxílio visual dos gráficos, podemos tirar algumas conclusões exploratórias. Os dados se referem ao consumo médio diário registrado de julho de 1997 até setembro de 2024 de uma determinada residência. Existe uma tendência de aumento de consumo se observado os primeiros três quartos da série, seguido de uma tendência de queda após a mudança da derivada em torno do ano de 2015, e daí para a frente a série apresenta uma tendência de queda. Observada de forma completa, a série apresenta formato aproximadamente senoidal. Picos na série sugerem algum tipo de padrão sazonal, possivelmente relacionado a utilização de determinados eletrodomésticos, como ar condicionado, em determinadas estações do ano. Existe um pico bastante expressivo na série em torno do ano de 2004 em relação as vizinhanças próximas, que foi descrito em aula como sendo a utilização de uma betoneira neste ano.

Conclusões exploratórias podem ser extraídas ao observar o comportamento da série, como possível aumento do número de residentes no domicílio e aumento do padrão de vida com mais equipamentos eletrodomésticos utilizando energia, seguido de possível diminuição do número de residentes, como filhos que se mudam, e avanço tecnológico de eletrodomésticos mais sustentáveis que utilizam menos energia elétrica.

## Questão 3

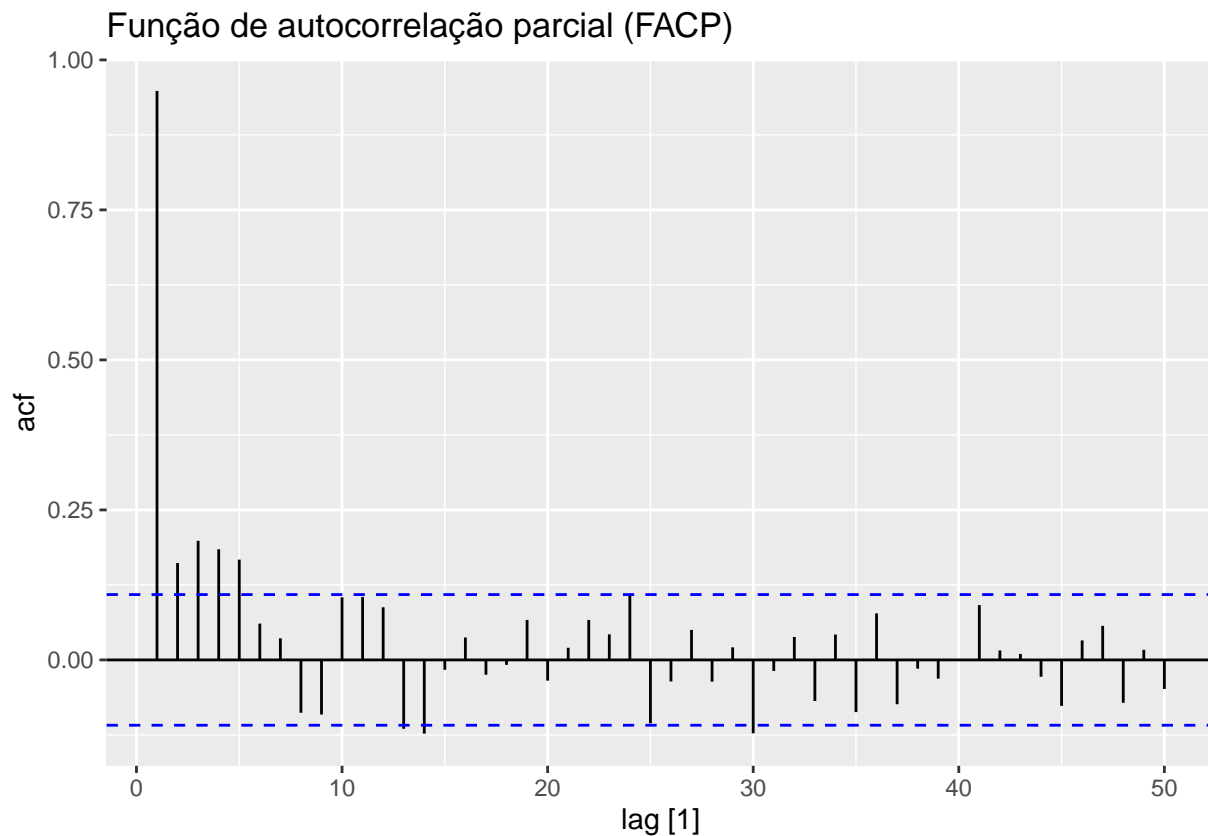
```
ts <- dados %>% dplyr::select(mes, Yt) %>% as_tibble() %>% na.omit()  
  
ts = as_tsibble(ts, index=mes, regular=F)  
  
ts %>%
```

```
ACF(var = Yt,lag_max = 50) %>%
autoplot() +
labs(title="Função de autocorrelação (FAC)")
```



Este gráfico da função de autocorrelação indica um lento decaimento das autocorrelações em função do crescimento da defasagem (lag). Vemos que todas as barras estão significativamente acima do limite superior da banda de credibilidade, indicando não estacionariedade da série. Neste caso, é indicado uma operação de diferenciação para tornar a série estacionária.

```
ts %>%
  ACF(var = Yt,lag_max = 50,type = "partial") %>%
  autoplot() +
  labs(title="Função de autocorrelação parcial (FACP)")
```



Este gráfico da função de autocorrelação parcial indica que a autocorrelação parcial da série é significativa até a defasagem 1, seguido de autocorrelações acima da banda de confiança de 95% por mais 4 defasagens, e, após isso, as autocorrelações não ultrapassam mais significativamente as bandas de confiança de 95%, indicando que a série pode ser modelada por um modelo AR(1) aou até mesmo um modelo AR(2).

## Questão 4

```
st <- ts(ts$Yt, start = c(1997, 6), frequency = 12)
aTSA::adf.test(st, nlag = 10)
```

```
## Augmented Dickey-Fuller Test
## alternative: stationary
##
## Type 1: no drift no trend
##      lag    ADF p.value
## [1,]  0 -0.995  0.323
## [2,]  1 -0.759  0.407
## [3,]  2 -0.527  0.490
## [4,]  3 -0.372  0.537
## [5,]  4 -0.224  0.580
## [6,]  5 -0.168  0.596
## [7,]  6 -0.142  0.603
## [8,]  7 -0.213  0.583
## [9,]  8 -0.274  0.565
## [10,] 9 -0.217  0.581
## Type 2: with drift no trend
##      lag    ADF p.value
## [1,]  0 -3.07  0.0311
## [2,]  1 -2.67  0.0846
```

```
## [3,] 2 -2.27 0.2197
## [4,] 3 -1.85 0.3849
## [5,] 4 -1.66 0.4617
## [6,] 5 -1.54 0.5077
## [7,] 6 -1.45 0.5414
## [8,] 7 -1.41 0.5530
## [9,] 8 -1.51 0.5174
## [10,] 9 -1.45 0.5399
## Type 3: with drift and trend
## lag ADF p.value
## [1,] 0 -3.4272 0.0493
## [2,] 1 -2.6670 0.2949
## [3,] 2 -1.8920 0.6216
## [4,] 3 -1.1532 0.9129
## [5,] 4 -0.6080 0.9766
## [6,] 5 -0.2919 0.9900
## [7,] 6 -0.0666 0.9900
## [8,] 7 -0.2939 0.9900
## [9,] 8 -0.6252 0.9758
## [10,] 9 -0.3712 0.9873
## ----
## Note: in fact, p.value = 0.01 means p.value <= 0.01
```

Sob a hipótese nula da série ser não estacionária, não rejeitamos a hipótese nula.

Ou seja, concluímos não haver evidências de estacionariedade da série pelo teste aumentado de Dickey-Fuller.

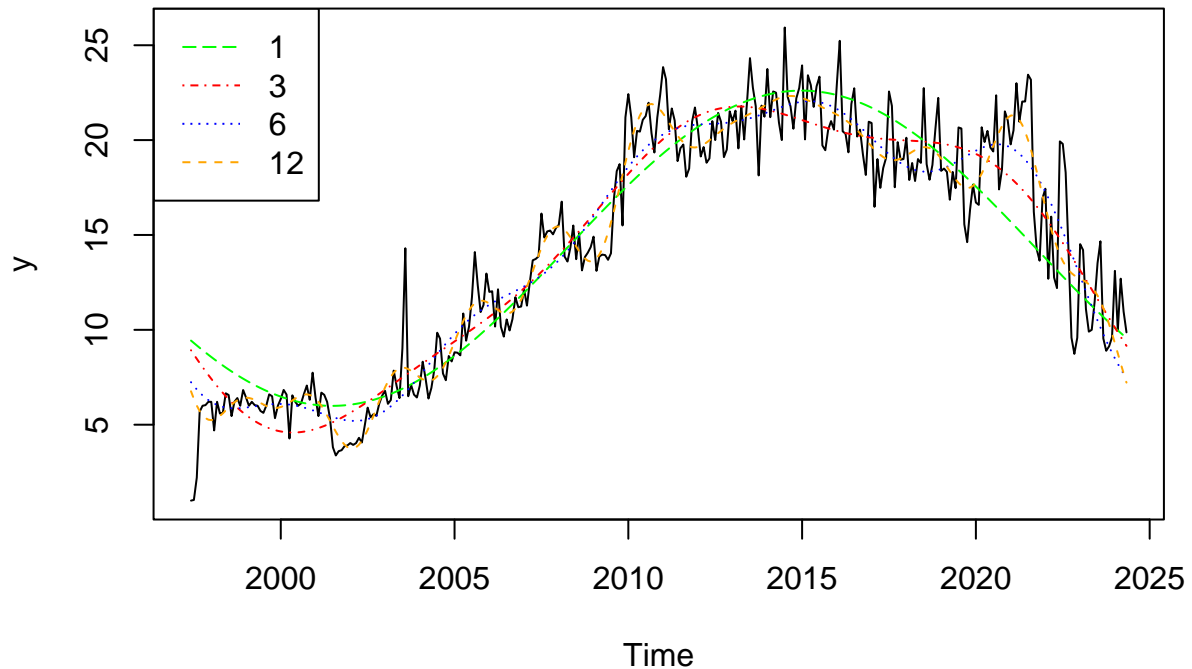
```
y <- st

fit1 <- haRmonics(y = y,
                  numFreq = 1,
                  delta = 0.1)
fit3 <- haRmonics(y = y,
                  numFreq = 3,
                  delta = 0.1)
fit6 <- haRmonics(y = y,
                  numFreq = 6,
                  delta = 0.1)
fit12 <- haRmonics(y = y,
                  numFreq = 12,
                  delta = 0.1)

x = ts(fit1$fitted, start = c(1997, 6), frequency = 12)
w = ts(fit3$fitted, start = c(1997, 6), frequency = 12)
z = ts(fit6$fitted, start = c(1997, 6), frequency = 12)
v = ts(fit12$fitted, start = c(1997, 6), frequency = 12)

plot(y, pch = 16, main = "Previsoes de uma modelagem Harmonica")
lines(x, lty = 5, col = "green")
lines(w, lty = 4, col = "red")
lines(z, lty = 3, col = "blue")
lines(v, lty = 2, col = "orange")
legend("topleft", legend = c("1", "3", "6", "12"),
       col = c("green", "red", "blue", "orange"), lty = c(5, 4, 3, 2))
```

## Previsões de uma modelagem Harmonica



Pela Figura, notamos que a regressão harmônica não parece conseguir capturar as sazonalidades da série.

## Questão 5

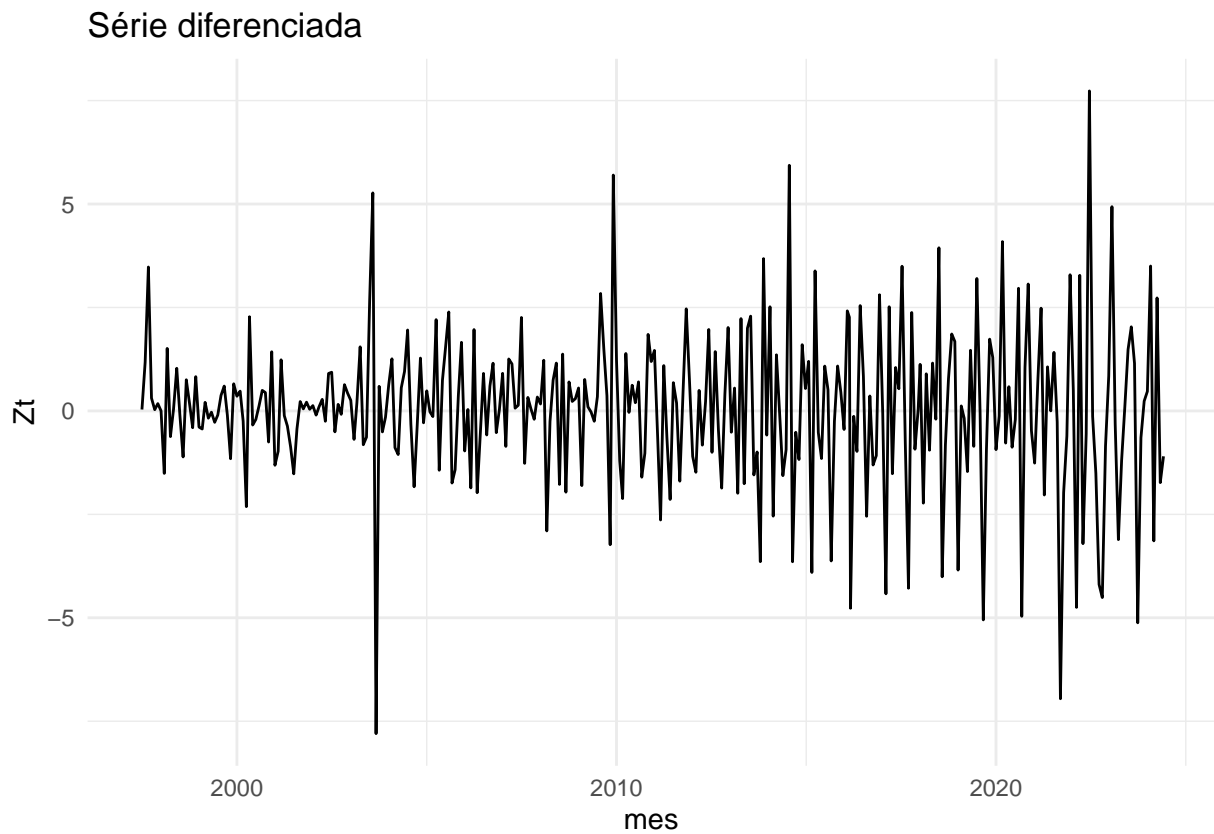
```
# Calcular  $Z_t = Y_t - Y_{t-1}$ 
ts = ts %>% mutate(Zt = c(NA, diff(Yt)))
kable(head(ts))
```

mes	Yt	Zt
1997-06-01	1.000000	NA
1997-07-01	1.033333	0.0333333
1997-08-01	2.212121	1.1787879
1997-09-01	5.689655	3.4775340
1997-10-01	6.000000	0.3103448
1997-11-01	6.030303	0.0303030

O cálculo da diferenciação da série é crucial para analisar uma série temporal não estacionária. Podemos modelar agora uma série estacionária utilizando um modelo ARIMA, por exemplo.

## Questão 6

```
ts %>%
  ggplot() +
  geom_line(aes(x=mes, y=Zt)) + theme_minimal() + ggtitle("Série diferenciada")
```



A série diferenciada parece ser estacionária, com média aparentemente constante centrada em zero, e variância aparentemente constante. A série parece ser um ruído branco, salvo alguns padrões sem tendência empírica. Agora parece ser possível uma modelagem mais confiável, com modelos robustos.

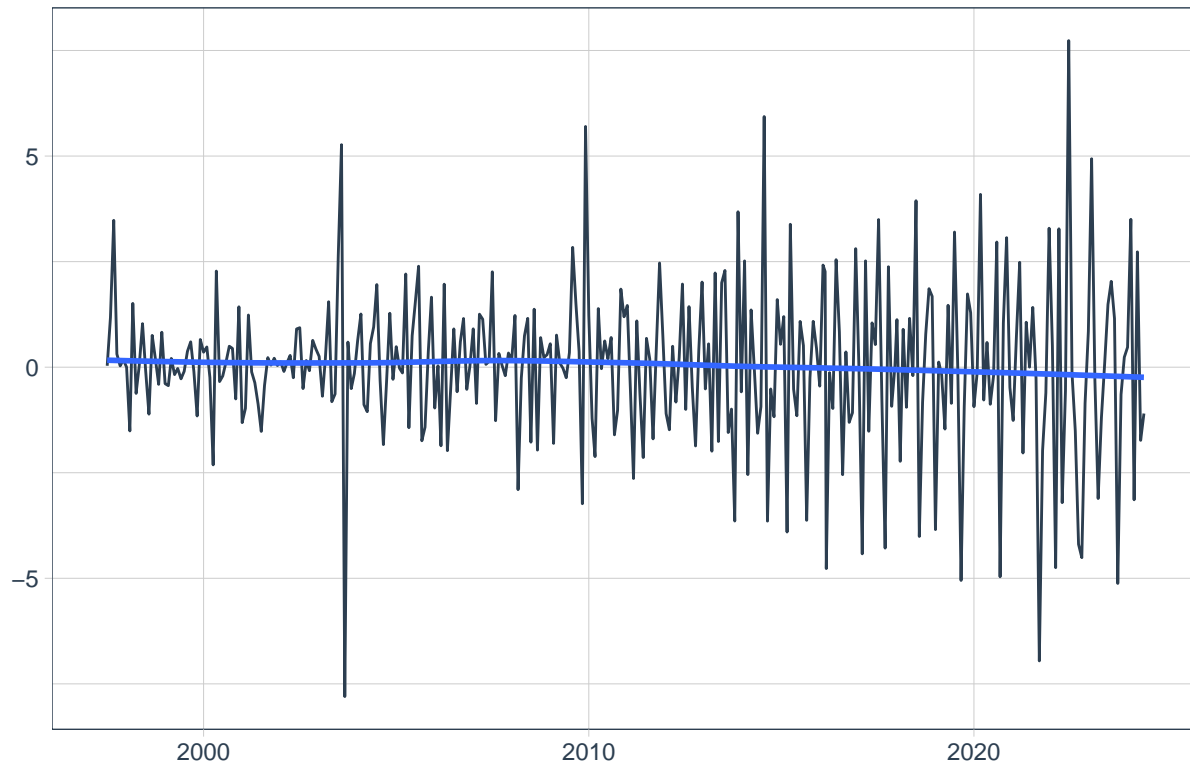
## Questão 7

Passando uma linha suavizada com auxílio do pacote *timetk* para verificação visual de tendência

```
ts %>%  
  plot_time_series(mes, Zt, .interactive = F, .smooth = T)
```

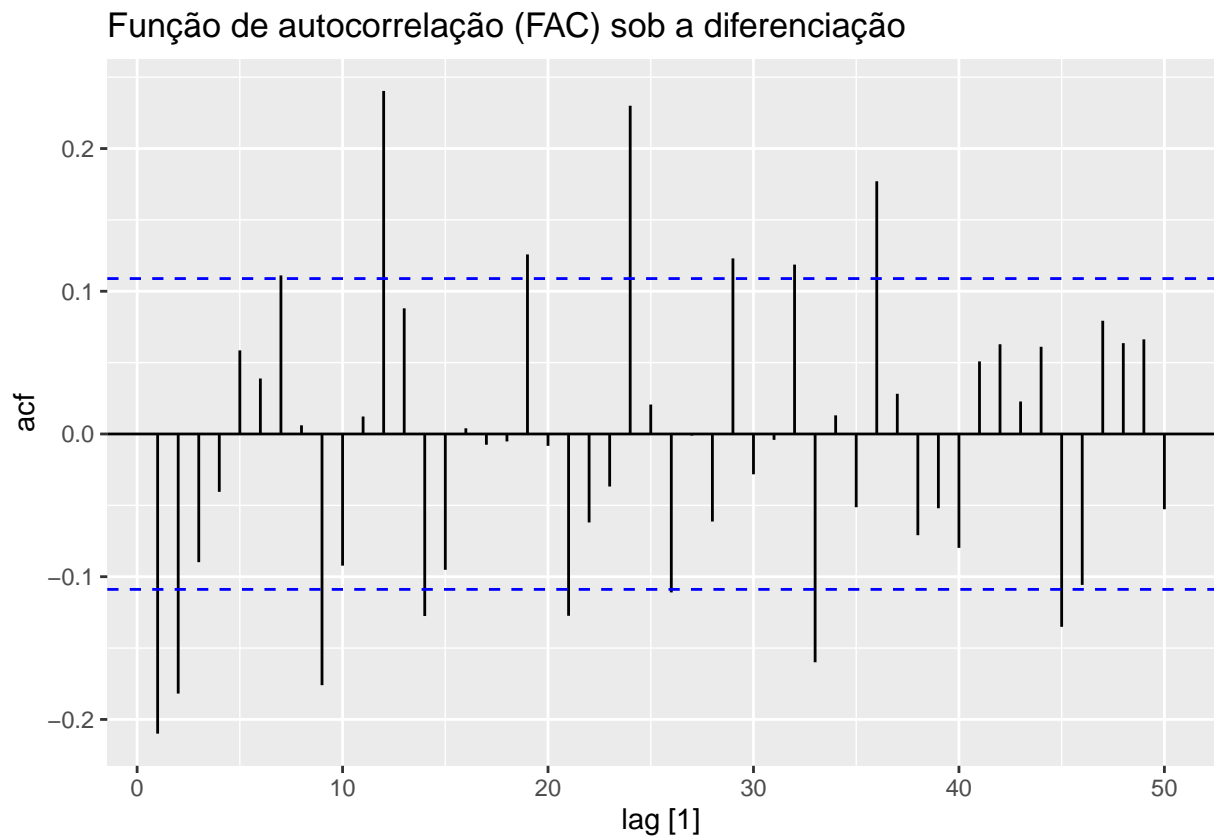


Time Series Plot



## Visualizando a FAC

```
ts %>%  
  ACF(var = Zt, lag_max = 50) %>%  
  autoplot() +  
  labs(title="Função de autocorrelação (FAC) sob a diferenciação")
```

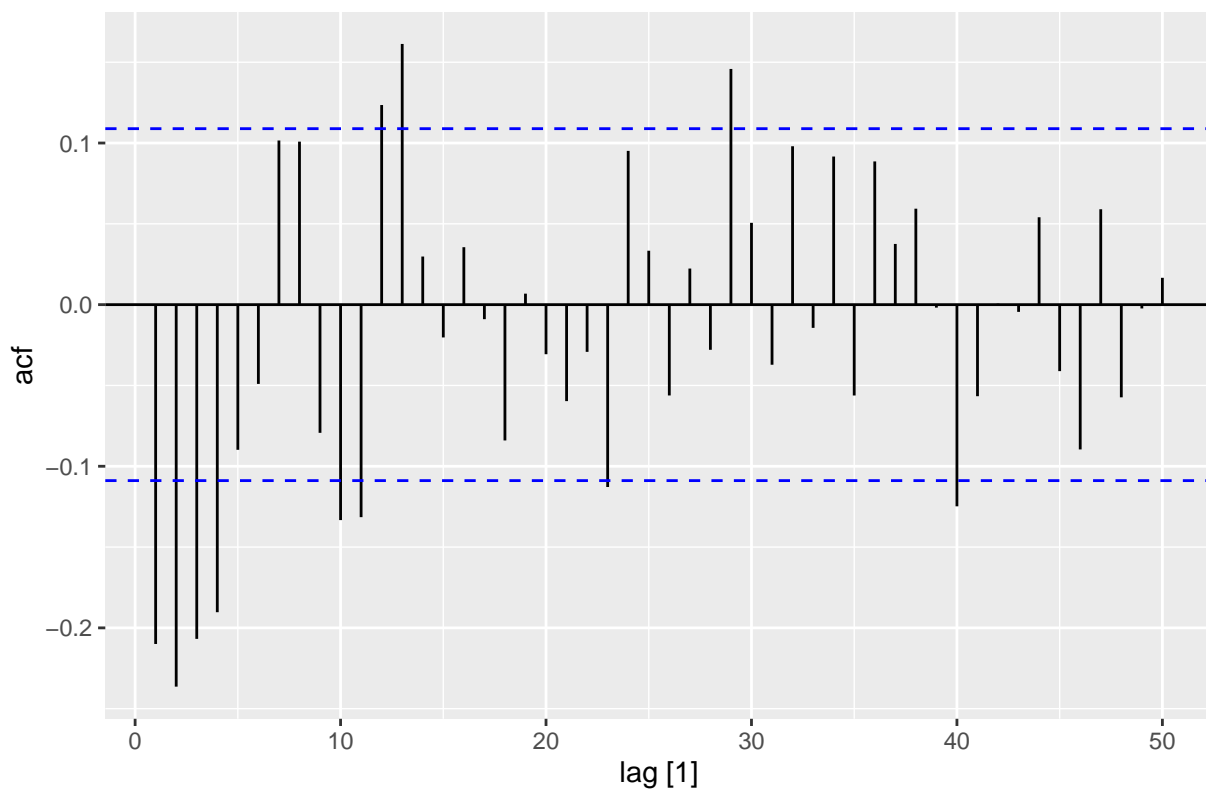


A presença de um grande número de valores contido no intervalo da banda de credibilidade indica estacionariedade desta série.

## Visualizando a FACP

```
ts %>%  
  ACF(var = Zt, lag_max = 50, type = "partial") %>%  
  autoplot() +  
  labs(title = "Função de autocorrelação parcial (FACP) sob a diferenciação")
```

### Função de autocorrelação parcial (FACP) sob a diferenciação



Novamente, os lags para a FACP indicam que a série perdeu suas tendências originais e apresenta comportamento agora aleatório, sugerindo estacionariedade.

```
st2 <- ts(ts$Zt, start = c(1997, 6), frequency = 12)
aTSA::adf.test(st2, nlag = 10)
```

```
## Augmented Dickey-Fuller Test
## alternative: stationary
##
## Type 1: no drift no trend
##      lag      ADF p.value
## [1,]  0 -22.16    0.01
## [2,]  1 -17.67    0.01
## [3,]  2 -15.61    0.01
## [4,]  3 -14.34    0.01
## [5,]  4 -12.32    0.01
## [6,]  5 -10.75    0.01
## [7,]  6  -8.37    0.01
## [8,]  7  -6.73    0.01
## [9,]  8  -6.82    0.01
## [10,] 9  -7.18    0.01
## Type 2: with drift no trend
##      lag      ADF p.value
## [1,]  0 -22.13    0.01
## [2,]  1 -17.65    0.01
## [3,]  2 -15.60    0.01
## [4,]  3 -14.33    0.01
## [5,]  4 -12.31    0.01
## [6,]  5 -10.75    0.01
## [7,]  6  -8.37    0.01
## [8,]  7  -6.72    0.01
```

```
## [9,] 8 -6.82 0.01
## [10,] 9 -7.19 0.01
## Type 3: with drift and trend
## lag ADF p.value
## [1,] 0 -22.17 0.01
## [2,] 1 -17.72 0.01
## [3,] 2 -15.69 0.01
## [4,] 3 -14.47 0.01
## [5,] 4 -12.50 0.01
## [6,] 5 -10.96 0.01
## [7,] 6 -8.54 0.01
## [8,] 7 -6.89 0.01
## [9,] 8 -7.01 0.01
## [10,] 9 -7.47 0.01
## ----
## Note: in fact, p.value = 0.01 means p.value <= 0.01
```

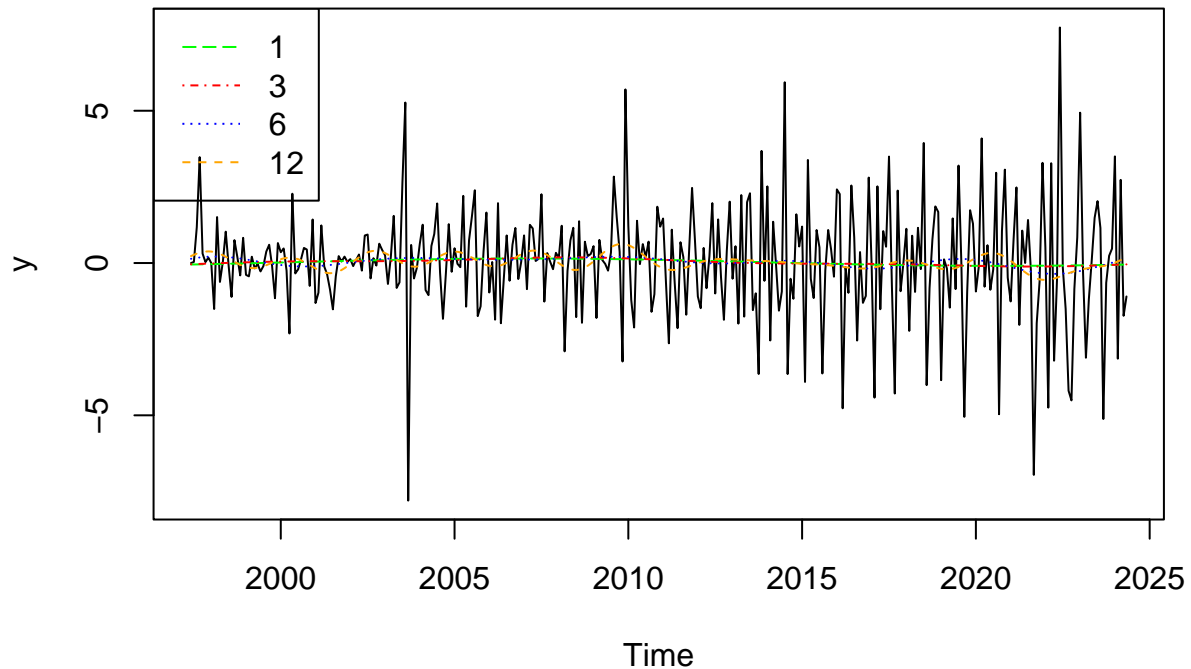
O teste aumentado de Dickey-Fuller sob a hipótese nula de não estacionariedade da série, rejeita a hipótese nula com p-valor < 0,01. Ou seja, temos fortes indícios apontando para a estacionariedade da série.

```
st2[1] = 0
y <- st2
fit1 <- haRmonics(y = y,
                  numFreq = 1,
                  delta = 0.1)
fit3 <- haRmonics(y = y,
                  numFreq = 3,
                  delta = 0.1)
fit6 <- haRmonics(y = y,
                  numFreq = 6,
                  delta = 0.1)
fit12 <- haRmonics(y = y,
                  numFreq = 12,
                  delta = 0.1)

x = ts(fit1$fitted, start = c(1997, 6), frequency = 12)
w = ts(fit3$fitted, start = c(1997, 6), frequency = 12)
z = ts(fit6$fitted, start = c(1997, 6), frequency = 12)
v = ts(fit12$fitted, start = c(1997, 6), frequency = 12)

plot(y, pch = 16, main = "Previsoes de uma modelagem Harmonica")
lines(x, lty = 5, col = "green")
lines(w, lty = 4, col = "red")
lines(z, lty = 3, col = "blue")
lines(v, lty = 2, col = "orange")
legend("topleft", legend = c("1", "3", "6", "12"),
      col = c("green", "red", "blue", "orange"), lty = c(5, 4, 3, 2))
```

## Previsões de uma modelagem Harmonica



Desta vez, o modelo de regressão harmônico capturou as estruturas sazonais da série, retornando apenas ruído branco.

Portanto, a operação de diferenciação foi essencial para definir estacionariedade da série, e prepará-la para modelagem preditiva.

### Questão 8

Com base nos passos anteriores e no fato de se tratar de uma série de consumo energético mensal (da qual espera-se padrões relacionados aos meses e estações do ano), iremos considerar a ordem sazonal  $s = 12$ .

Para a ordem de diferenciação, vimos que  $d = 1$  foi suficiente para atingir a estacionariedade da série, bem como uma diferenciação  $D = 1$  para a sazonalidade deve ser suficiente para eliminar a sazonalidade, se confirmado o pressuposto do parágrafo anterior.

O restante dos parâmetros  $(p, P, q, Q)$  devem ser obtidos via métodos iterativos, observando o AIC e BIC dos modelos para escolher o que melhor se ajusta aos dados.

Para iniciar o método iterativo, devemos “chutar” um valor inicial, e esperar pela convergência em torno de um valor ótimo partindo do chute inicial.

### Questão 9

Definindo uma malha de valores, e testando todas as combinações entre eles por meio de 4 loops *for*. OBS: Ver questão 10 para explicação dos valores aqui contidos.

```
modelos <- data.frame(p = integer(),
                      P = integer(),
                      q = integer(),
                      Q = integer(),
                      AIC = numeric(),
                      BIC = numeric())
```

```

tic()
for(p in 0:2){
  for(P in 0:2){
    for(q in 0:3){
      for(Q in 0:3){
        tryCatch({
          fit = astsa::sarima(st, details = FALSE, Model = FALSE,
                             p = p, d = 1, q = q, P = P, D = 1, Q = Q, S = 12)
          AIC = fit$ICs[1]
          BIC = fit$ICs[3]
          mod = c(p, P, q, Q, AIC, BIC)
          modelos = rbind(modelos, mod)
        }, error = function(e) {
        })
      }
    }
  }
}
toc()

colnames(modelos) = c("p", "P", "q", "Q", "AIC", "BIC")

```

## Questão 10

Foi testada inicialmente os valores 0, 1 e 2 e as combinações entre eles na primeira rodada da malha da questão anterior. Como haviam modelos candidatos com  $q$  ou  $Q = 2$ , resolvi expandir a malha desses dois parâmetros. Portanto, os valores testados foram  $p = [0,1,2]$ ,  $P = [0,1,2]$ ,  $q = [0,1,2,3]$ ,  $Q = [0,1,2,3]$ . Desta vez, nenhum dos modelos candidatos tinham o parâmetro na margem, portanto parece suficiente estes conjuntos de valores para os 4 parâmetros do modelo, enquanto fixo os parâmetros  $S = 12$  e  $d = 1$ .

```

modelos %>%
  arrange(BIC, AIC) %>% head() %>% kable()

```

p	P	q	Q	AIC	BIC
0	0	1	1	3.907011	3.943087
0	0	2	1	3.898798	3.946898
1	0	1	1	3.900741	3.948842
0	1	1	1	3.913114	3.961214
0	0	1	2	3.913143	3.961243
1	0	2	1	3.904225	3.964350

## Questão 11

Pelo observado anteriormente, o modelo que apresentou menor valor  $BIC$  foi o modelo com parâmetros  $p = 0$ ,  $P = 0$ ,  $q = 1$ ,  $Q = 1$ . Portanto, utilizarei este.

```

fit = astsa::sarima(st, p = 0, d = 1, q = 1, P = 0, D = 1, Q = 1, S = 12)

```

```

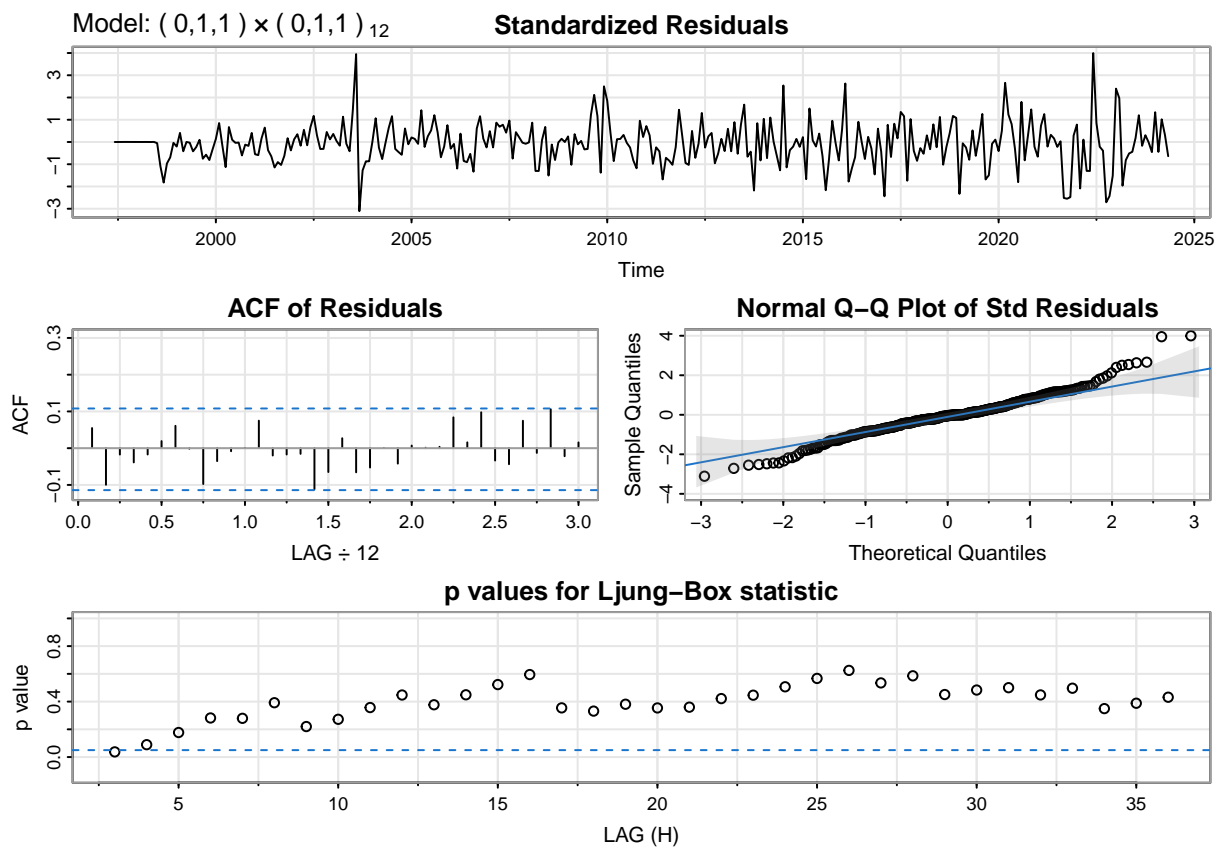
## initial value 0.839633
## iter 2 value 0.571359

```

```

## iter    3 value 0.549661
## iter    4 value 0.545316
## iter    5 value 0.534964
## iter    6 value 0.532353
## iter    7 value 0.531989
## iter    8 value 0.531659
## iter    9 value 0.531655
## iter   10 value 0.531655
## iter   10 value 0.531655
## iter   10 value 0.531655
## final   value 0.531655
## converged
## initial  value 0.526939
## iter    2 value 0.525109
## iter    3 value 0.524927
## iter    4 value 0.524921
## iter    5 value 0.524921
## iter    5 value 0.524921
## iter    5 value 0.524921
## final   value 0.524921
## converged
## <><><><><><><><><><><><><><>
##
## Coefficients:
##      Estimate      SE t.value p.value
## ma1   -0.5035 0.0556  -9.0584      0
## sma1  -0.7695 0.0456 -16.8671      0
##
## sigma^2 estimated as 2.757438 on 309 degrees of freedom
##
## AIC = 3.907012  AICc = 3.907137  BIC = 3.943087
##

```



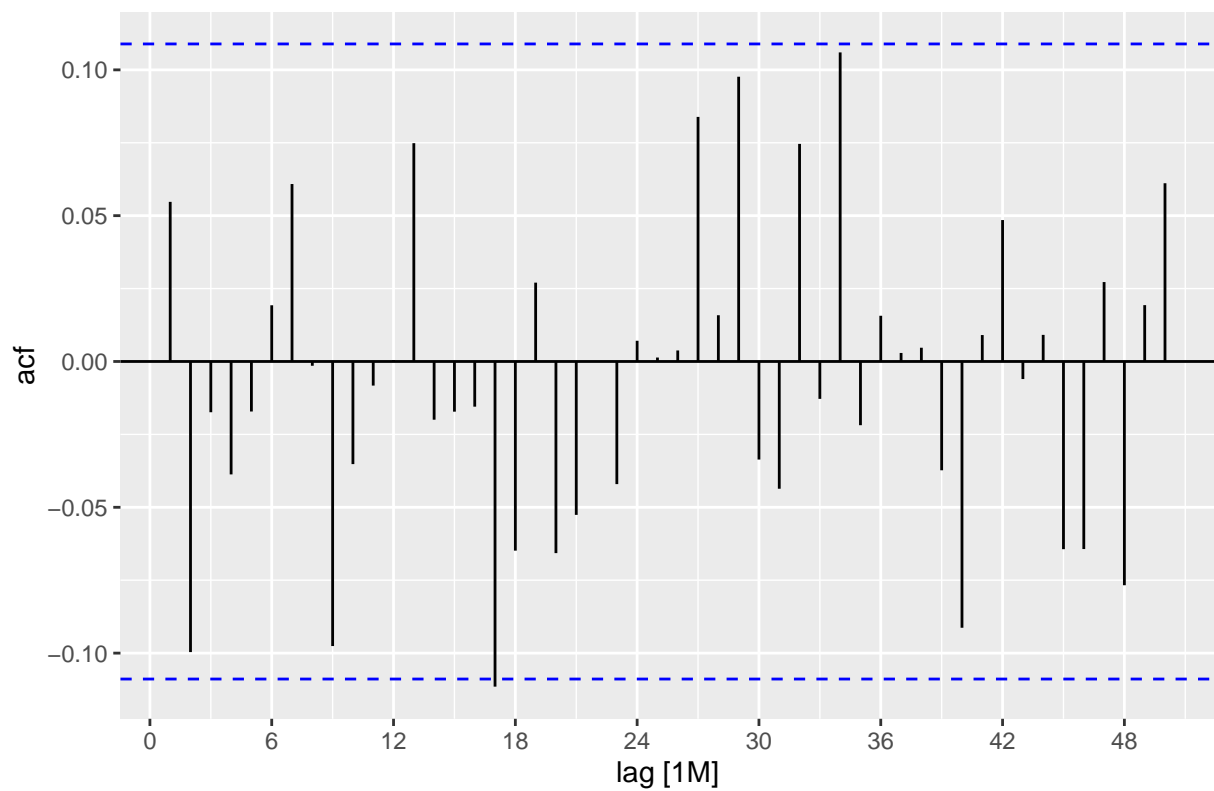
## 11.2 FAC e FACP

```
res = as_tsibble(fit[["fit"]][["residuals"]])

res %>%
  ACF(value, lag_max = 50) %>%
  autoplot() +
  labs(title="Função de autocorrelação (FAC) sob os resíduos do modelo")
```

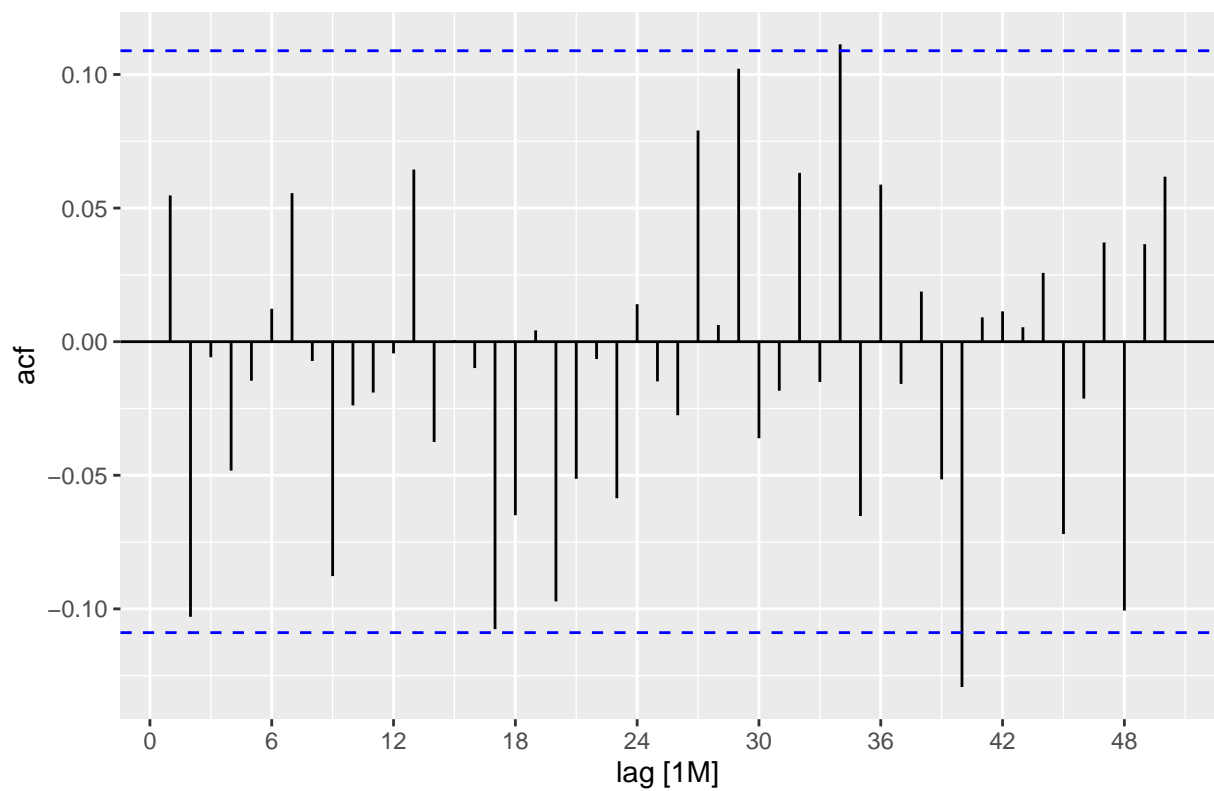


Função de autocorrelação (FAC) sob os resíduos do modelo



```
res %>%
  ACF(value,lag_max = 50,type = "partial") %>%
  autoplot() +
  labs(title="Função de autocorrelação parcial (FACP) sob os resíduos do modelo")
```

Função de autocorrelação parcial (FACP) sob os resíduos do modelo



```
Box.test(fit[["fit"]][["residuals"]], lag = 50, type = "Ljung")
```

```
##  
## Box-Ljung test  
##  
## data: fit[["fit"]][["residuals"]]  
## X-squared = 46.605, df = 50, p-value = 0.6104
```

Sob a hipótese nula de independência, o teste de Ljung-Box não rejeita a hipótese nula. Portanto, existem indícios de independência na série dos resíduos.

Avaliando as formas da FAC, FACP e resultado do teste de Ljung-Box, não aparenta haver autocorrelação significativa com lag = 50 para os resíduos, sugerindo que os resíduos se tratam de ruído branco. Logo, este modelo aparenta ter ajustado quanto a estrutura de dados, sendo um modelo adequado para previsões.

### 11.3 Testes de normalidade dos resíduos:

```
# Teste de Shapiro-Wilk  
shapiro.test(fit[["fit"]][["residuals"]])
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: fit[["fit"]][["residuals"]]  
## W = 0.97127, p-value = 0.000004725
```

```
# Teste de Lilliefors  
lillie.test(fit[["fit"]][["residuals"]])
```

```
##  
## Lilliefors (Kolmogorov-Smirnov) normality test  
##  
## data: fit[["fit"]][["residuals"]]  
## D = 0.065286, p-value = 0.002024
```

Sob a hipótese nula de normalidade dos resíduos, os testes de Shapiro-Wilk e Lilliefors rejeitam a normalidade dos resíduos a 5% de confiança. Entretanto, observando o gráfico Q-Q dos resíduos produzido pela função, notamos que esta rejeição aparenta está relacionada a alguns valores discrepantes na cauda.

### Subitens 11.4 e 11.5:

Não necessário, visto que o modelo aparenta ser adequado.

## Questão 12

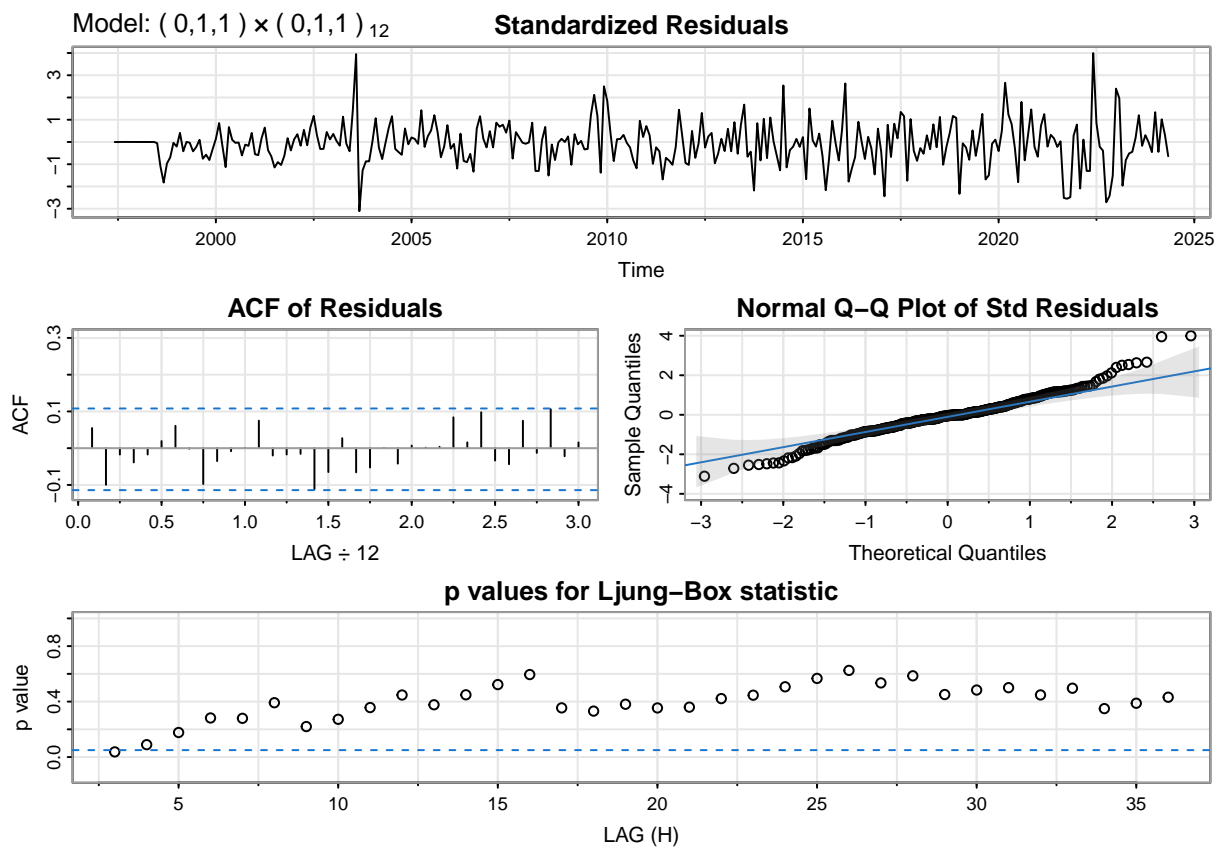
### 12.1

```
fit = astsa::sarima(st,p = 0, d = 1, q = 1, P = 0, D = 1, Q = 1, S = 12)
```

```

## initial value 0.839633
## iter 2 value 0.571359
## iter 3 value 0.549661
## iter 4 value 0.545316
## iter 5 value 0.534964
## iter 6 value 0.532353
## iter 7 value 0.531989
## iter 8 value 0.531659
## iter 9 value 0.531655
## iter 10 value 0.531655
## iter 10 value 0.531655
## iter 10 value 0.531655
## final value 0.531655
## converged
## initial value 0.526939
## iter 2 value 0.525109
## iter 3 value 0.524927
## iter 4 value 0.524921
## iter 5 value 0.524921
## iter 5 value 0.524921
## iter 5 value 0.524921
## final value 0.524921
## converged
## <><><><><><><><><><><><><><>
##
## Coefficients:
##      Estimate      SE t.value p.value
## ma1  -0.5035 0.0556  -9.0584      0
## sma1 -0.7695 0.0456 -16.8671      0
##
## sigma^2 estimated as 2.757438 on 309 degrees of freedom
##
## AIC = 3.907012 AICc = 3.907137 BIC = 3.943087
##

```

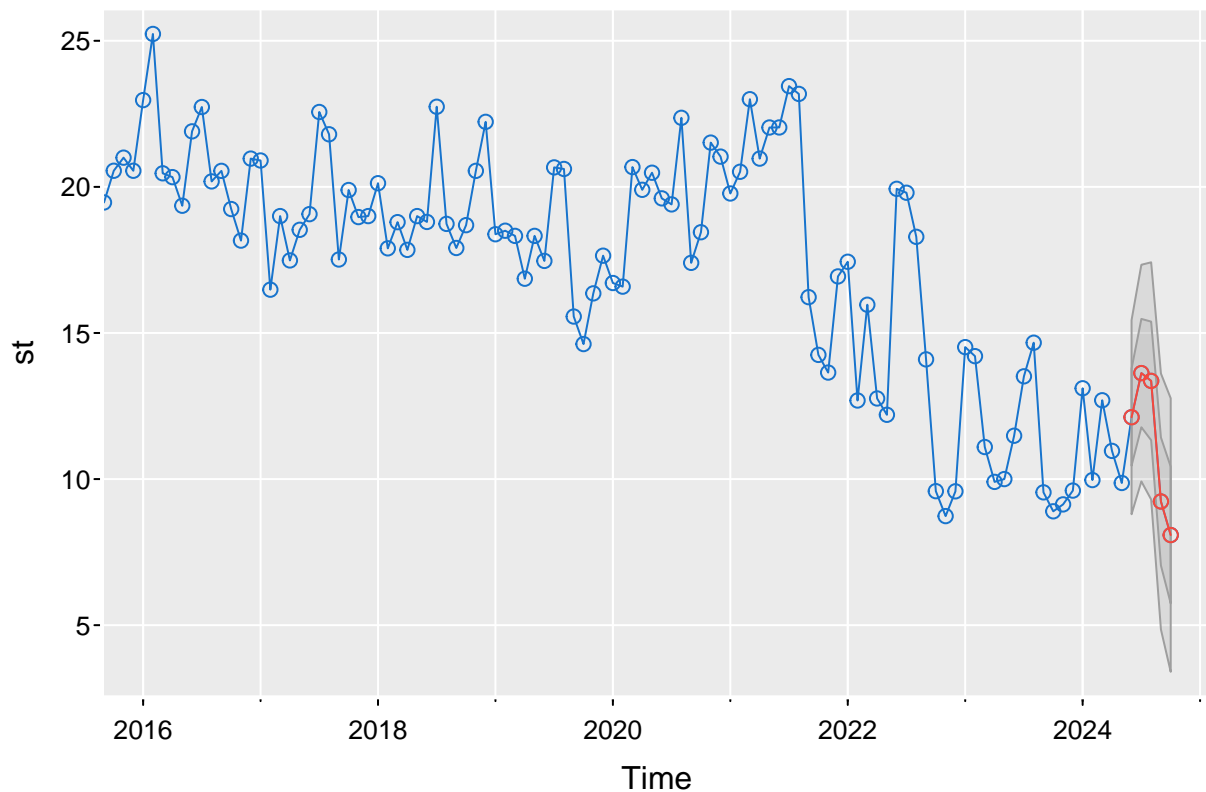


## 12.2

Previsão 5 passos a frente utilizando a base completa:

```
prev = sarima.for(st,p = 0, d = 1, q = 1, P = 0, D = 1, Q = 1, S = 12,
  n.ahead = 5, gg=TRUE, col=4,main="Previsão do modelo completo p/ 5 prox. passos")
```

## Previsão do modelo completo p/ 5 prox. passos



Previsões do modelo:

prev

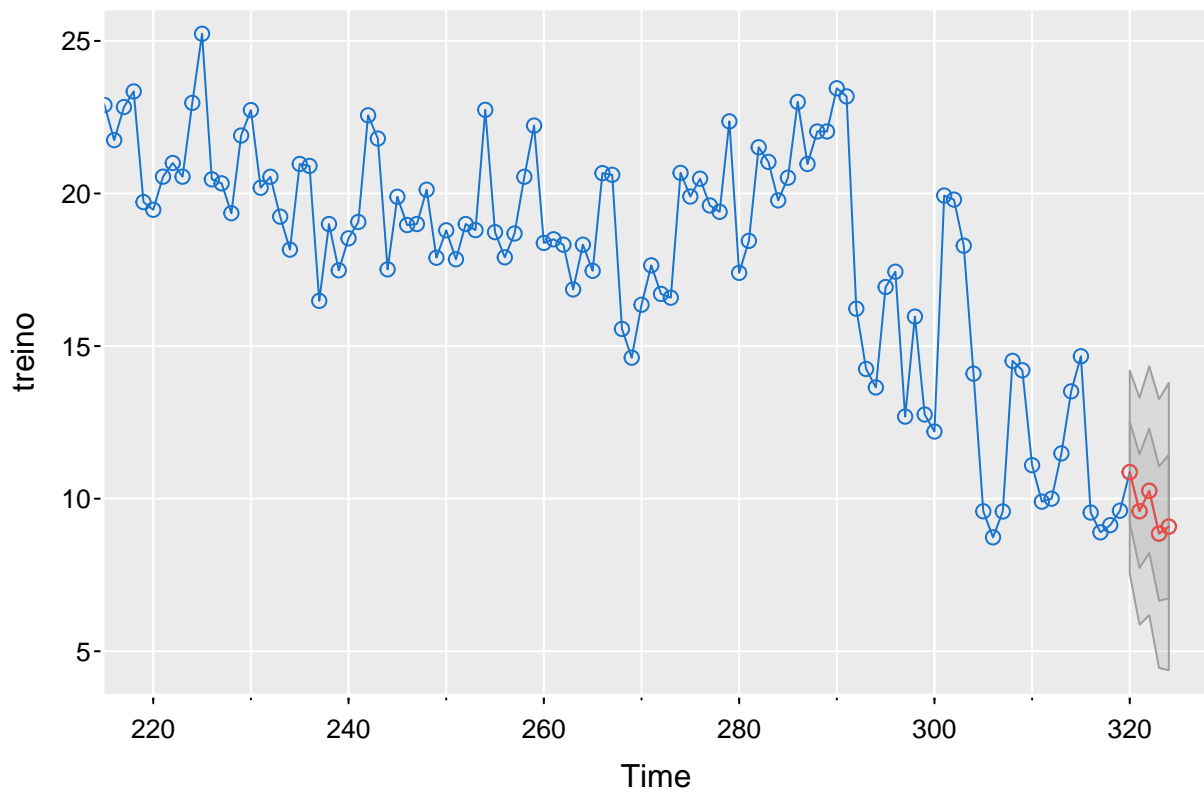
```
## $pred
##           Jun           Jul           Aug           Sep           Oct
## 2024 12.117672 13.626804 13.360397  9.232916  8.084329
##
## $se
##           Jun           Jul           Aug           Sep           Oct
## 2024 1.660554 1.853971 2.029034 2.190148 2.340196
```

Fazendo um novo modelo para poder comparar o poder deste: Mantido todos os parâmetros fixo, vamos modelar para as n-5 (319) observações da série, prever os 5 passos seguintes, e depois comparar com o real valor para validação do modelo

```
treino = ts(st[1:319])
teste = ts(st[320:324])
```

```
prev = sarima.for(treino,p = 0, d = 1, q = 1, P = 0, D = 1, Q = 1, S = 12,
                  n.ahead = 5, gg=TRUE, col=4,main="Modelo com n-5 observações, e 5 previsões")
```

## Modelo com n=5 observações, e 5 predições



### 12.2

```
prev$pred
```

```
## Time Series:
## Start = 320
## End = 324
## Frequency = 1
## [1] 10.870019  9.590050 10.256936  8.856744  9.084641
```

### 12.3 MAPE

```
MAPE = MAPE(prev$pred, as.vector(teste))
MAPE
```

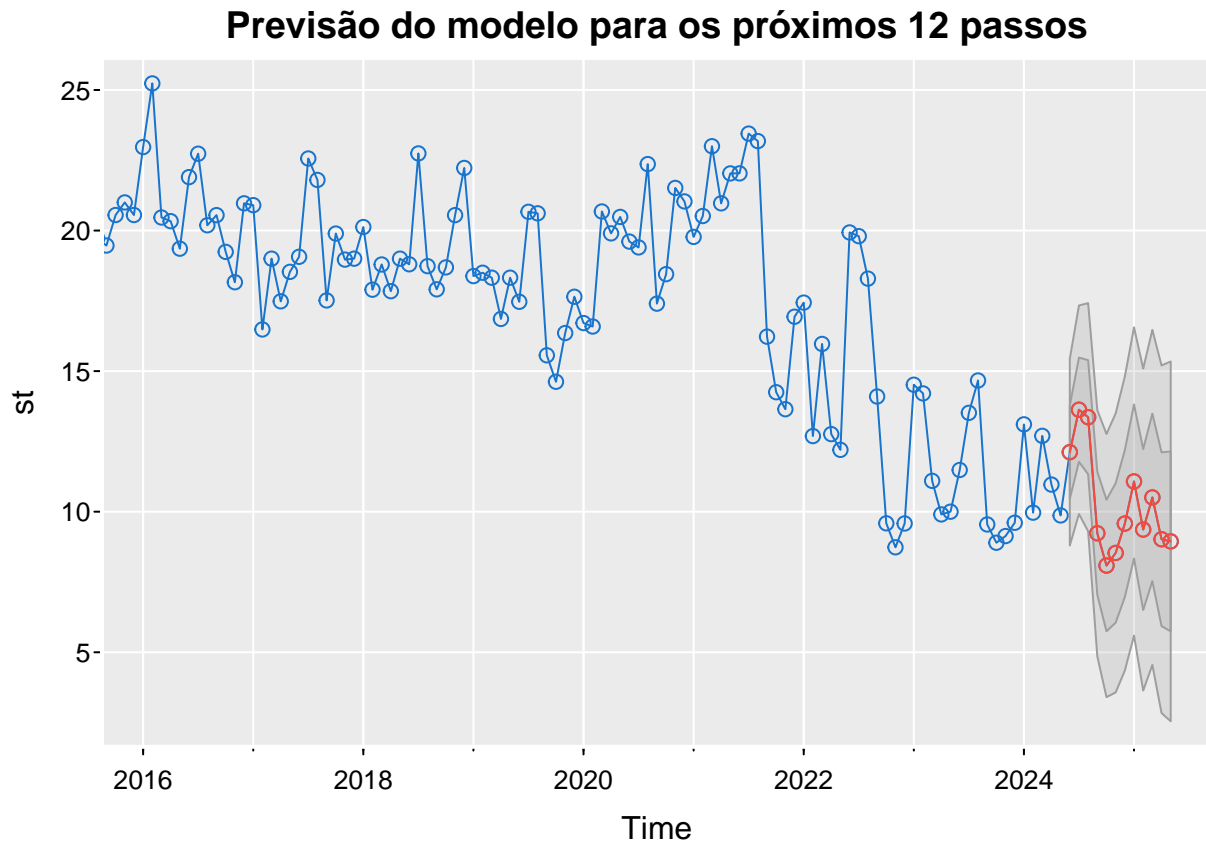
```
## [1] 0.134424
```

### 12.4

Com um  $MAPE = 0.134424$ , ou seja, estando no intervalo 10%-20%, podemos dizer que se trata de um bom modelo preditivo.

## Questão 13

```
prev = sarima.for(st,p = 0, d = 1, q = 1, P = 0, D = 1, Q = 1, S = 12,  
  n.ahead = 12, gg=TRUE, col=4,main="Previsão do modelo para os próximos 12 passos")
```



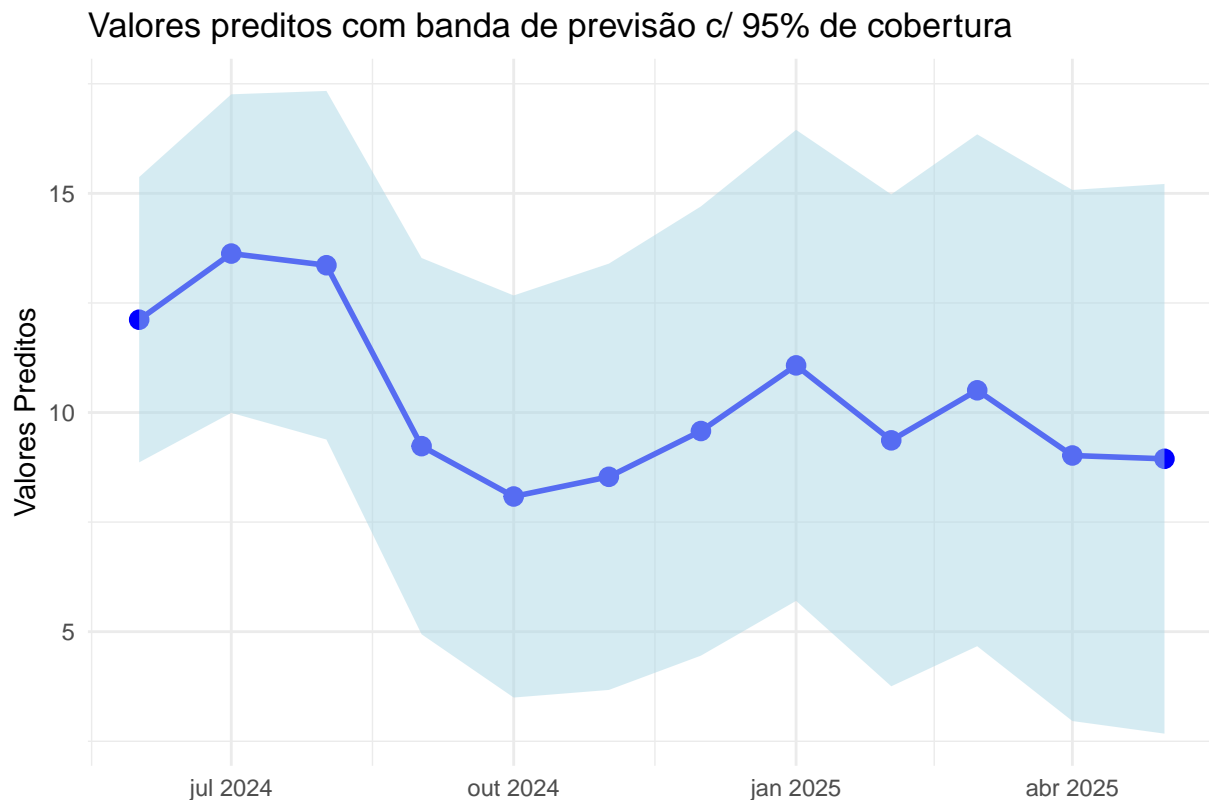
```
datas <- seq.Date(from = as.Date("2024-06-01"), to = as.Date("2025-05-31"), by = "months")
```

```
previsoes = data.frame(datas,prev$pred,prev$se)  
colnames(previsoes) = c("data","Valores_preditos","Erro_padrao")
```

```
previsoes = previsoes %>%  
  mutate(Limite_superior = Valores_preditos + (1.96 * Erro_padrao),  
         Limite_inferior = Valores_preditos - (1.96 * Erro_padrao))
```

```
ggplot(previsoes, aes(x = data)) +  
  geom_line(aes(y = Valores_preditos), color = "blue", size = 1) +  
  geom_point(aes(y = Valores_preditos), color = "blue", size = 3) +  
  geom_ribbon(aes(ymin = Limite_inferior, ymax = Limite_superior), fill = "lightblue", alpha = 0.5)  
labs(title = "Valores preditos com banda de previsão c/ 95% de cobertura",  
     x = "",  
     y = "Valores Preditos") +  
theme_minimal()
```

```
## Don't know how to automatically pick scale for object of type <ts>. Defaulting  
## to continuous.
```



Modelos de séries temporais em geral não são muito bons para gerar previsões muito a frente dos dados. Talvez as 4 primeiras previsões estejam próximas do esperado (como verificado anteriormente com os modelos incompletos e previsões com dados do conjunto). Logo, utilizar este modelo para prever com alta credibilidade o último ponto (12) pode ser um tanto inverossímil.

A hipótese de normalidade de fato foi descartada no passo 11.3. Entretanto, pelo gráfico Q-Q, podemos não levar com tanta certeza esta hipótese. Para além disso, utilizar um valor maior que 1,96 para gerar a banda de confiança pode ser suficiente, quantificando uma distribuição aproximadamente normal porém de cauda mais pesada. Talvez esta fosse uma boa estratégia para tornar a análise mais crível.

## Questao 14

Este foi um estudo bem completo para modelagem preditiva. Estudamos a série original, estacionariedade, FAC e FACP. Fiz diagnóstico de necessidade de diferenciação para estacionariedade, definição de sazonalidade, bem como iteração de parâmetros para um modelo ARIMA sazonal. Fizemos diagnóstico dos resíduos, testes de normalidade, e previsões para 5 e 12 passos a frente. Portanto, isso traz credibilidade para as previsões apresentadas ao final, com a ressalva do modelo não ser eficiente para previsões muito a frente do escopo. Novas informações de medição de eletricidade podem entretanto ser re-inseridas no modelo. Ou seja, se a cada mês for re-inserido o dado de eletricidade lido, pode-se ter um modelo útil para entender o comportamento da energia no trimestre subjacente, por exemplo. Uma boa ressalva é quanto a não normalidade dos resíduos, ou seja, a banda de credibilidade pode ser insuficiente para capturar valores atípicos.