LISTA 5

ANÁLISE DE SÉRIES TEMPORAIS

Tailine J. S. Nonato

July 14, 2024

Descrição da Atividade

Considere a série do consumo mensal de energia elétrica (`ConsumoEnergiaEAgua_New.xlsx`). Denotando X_t como o valor do consumo registrado no mês t e D_t como o número de dias de leitura, faça o que se pede a seguir.

Etapa 1

Calcule o consumo médio diário $Y_t = \frac{X_t}{D_t}$, e explique o porquê dessa transformação.

Etapa 2

Apresente o gráfico da evolução temporal de Y_t, e apresente sua descrição, contemplado elementos como o tamanho da série e periodicidade dos dados.

Etapa 3

Apresente os gráficos da função de autocorrelação (FAC) e da função de autocorrelação parcial (FACP) de Y_t, considerando um número apropriado de defasagens (lag), incluindo a banda de 95% de confiança sob a hipótese nula de não haver autocorrelação serial. Em um parágrafo, descreva as formas da FAC e da FAPC, explicando o que se pode diagnosticar/sugerir com base nelas.

Etapa 4

Aplique o teste aumentado de estacionariedade de Dickey-Fuller do pacote `aTSA` do R. Para a parte sazonal, faça a avaliação por meio de um modelo de regressão com funções harmônicas.

Etapa 5

Calcule a variação do consumo $Z_t = Y_t - Y_{t-1}$, e explique o papel/significado dessa transformação para a análise desses dados.

Etapa 6

Faça o gráfico da evolução temporal de Z_t, e descreva em um parágrafo o aspecto dessa figura, comparando-a com a forma observada no item 2.

Etapa 7

Repita os passos 3 e 4, comparando os novos resultados com os anteriores.

Etapa 8 a 14

Considere que $\hat{Y}_(t+h)$ representa a previsão no instante t+h obtida com base nas informações disponíveis até o tempo t; ou seja, $Y_1, ..., Y_t \rightarrow \hat{Y}_(t+h)$. Separe a massa de dados em duas partes, conforme esquema abaixo:

- Treinamento (modelagem): $Y_1, ..., Y_m$
- Validação: $Y_(m+1), ..., Y_n$

De 8 a 10 utilize os dados de treinamento para a modelagem, e de 11 a 14 utilize os dados de validação para a avaliação do modelo.

Etapa 8

Considerando o modelo SARIMA(p,d,q)×(P,D,Q)s para a série Y_t, defina um valor apropriado para a ordem sazonal s e as ordens de diferenciações d e D com base nos passos anteriores.

Etapa 9

Defina uma malha de valores para as ordens autorregressivas p e P e de médias móveis q e Q, e obtenha o valor do critério de informação bayesiano de Schwarz (BIC) para cada combinação (p,d,q)×(P,D,Q) por meio da função sarima do pacote astsa.

Etapa 10

Liste os modelos com os menores BIC. Certifique-se que o melhor modelo não possua uma ordem na extremidade da malha definida no item 9. Se houver, retorne para o passo 9, ampliando a malha.

Etapa 11

Inicie o diagnóstico com o modelo que apresenta o menor BIC:

1. Analise as estimativas dos parâmetros por meio da função sarima do pacote astsa.
2. Faça os gráficos da FAC e FACP residual, e aplique o teste de Ljung-Box.
3. Teste a normalidade residual.
4. Caso haja problemas em 11.1 e 11.2, repita a análise com os próximos modelos candidatos.
5. Caso não seja possível encontrar um modelo adequado, será preciso redefinir o modelo no passo 8. Se as ordens s, d, e D estiverem corretas, então é possível que o modelo SARIMA não seja apropriado.

Etapa 12

Como o método de estimação é recursivo, a obtenção dos erros de previsão um passo à frente na massa de validação pode ser realizada da seguinte forma:

1. Aplique o modelo sobre a base de dados completa, usando a função sarima do pacote astsa.
2. Obtenha os erros de previsão um passo à frente observados na parte da validação do modelo, ou seja, $\hat{e}_t = Y_t - \hat{Y}_t$, para $t = m+1, ..., n$.
3. Calcule um índice de desempenho preditivo. Por exemplo, obtenha o MAPE (mean absolute percentage error):

$$MAPE = \frac{100}{n-m} \cdot \sum_{t=m+1}^{n} \left(\frac{|Y_t - \hat{Y}_t|}{|Y_t|} \right)$$

4. Como referência, modelos com MAPE inferiores a 10% geralmente são considerados muito bons. Entre 10% e 20% são bons modelos preditivos, e entre 20% e 50% são modelos razoáveis/aceitáveis.

Etapa 13

Utilize a função sarima.for do pacote astsa para a obtenção de previsões para os próximos 12 meses (ou outro horizonte desejado) e a banda de previsão com 95% de cobertura. Discuta sobre as limitações dessas previsões, incluindo um insigh sobre como proceder se a hipótese de normalidade residual for descartada no passo 11.3.

Etapa 14

Redija um parágrafo concluindo o estudo (inclua uma recomendação sobre como o modelo deve ser atualizado à medida que novas informações estiverem disponíveis).

Respostas

Carregando os pacotes necessários

```
if (!require(pacman)) install.packages("pacman")
pacman::p_load(tidyverse,readxl, knitr, aTSA,forecast,astsa)
options(OutDec = ",")
```

Leitura e manipulação dos dados

```
energia <- read_excel("ConsumoEnergiaEAgua_New.xlsx")
energia <- energia[,c(1,3,4)]
energia <- na.omit(energia)
kable(tail(energia), align = "c",
    caption = "Últimos registros da base de dados")
```

Table 1: Últimos registros da base de dados

mes	Energia	Dias
2023-12-28	317	33
2024-01-28	367	28
2024-02-28	299	30
2024-03-30	419	33
2024-04-30	307	28
2024-05-31	296	30

Etapa 1

```
energia$consumo <- energia$Energia/energia$Dias
kable(mean(energia$consumo), align = "c",
    caption = "Consumo médio diário")
```

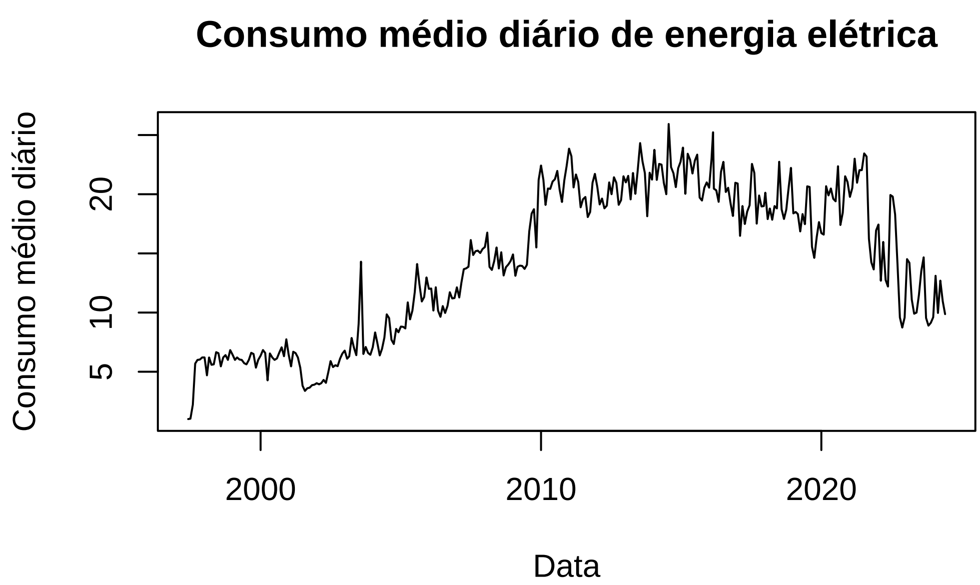
Table 2: Consumo médio diário

x
14,30075

Essa transformação é necessária para que possamos comparar o consumo de energia de diferentes meses, uma vez que o número de dias de leitura varia de um mês para o outro.

Etapa 2

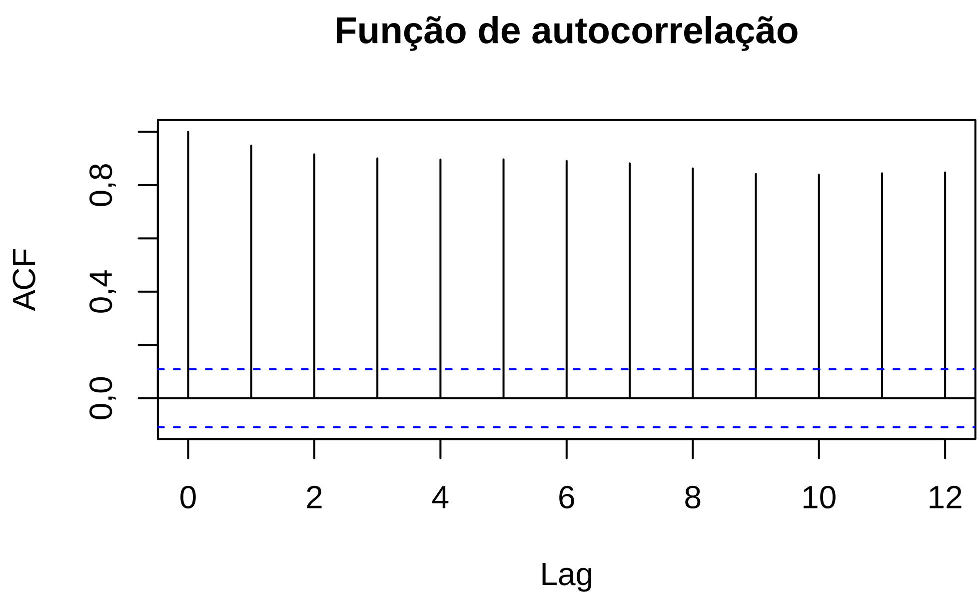
```
plot(energia$mes, energia$consumo, type = "l",
    main = "Consumo médio diário de energia elétrica",
    xlab = "Data",
    ylab = "Consumo médio diário")
```



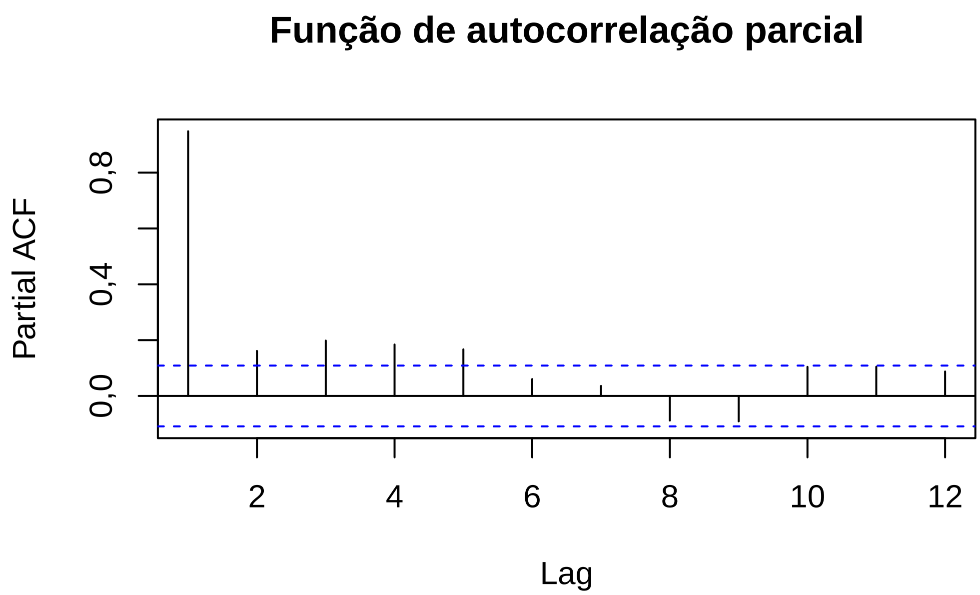
O gráfico apresenta o consumo médio diário de energia elétrica ao longo do tempo. A série é composta por 60 observações, com periodicidade mensal.

Etapa 3

```
acf(energia$consumo, lag.max = 12, main = "Função de autocorrelação")
```




```
pacf(energia$consumo, lag.max = 12, main = "Função de autocorrelação parcial")
```



Etapa 4

```
# Período sazonal (em meses)
s <- 12

adf_test <- aTSA::adf.test(energia$consumo)
```

```
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
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
Type 3: with drift and trend
     lag    ADF p.value
[1,]   0 -3,427  0,0493
[2,]   1 -2,667  0,2949
[3,]   2 -1,892  0,6216
[4,]   3 -1,153  0,9129
[5,]   4 -0,608  0,9766
[6,]   5 -0,292  0,9900
----
Note: in fact, p.value = 0.01 means p.value <= 0.01
```

```r
p_value_adf <- adf_test$type1[1, "p.value"]

if(p_value_adf > 0.05){
  d <- 1
  energia$consumo_diff <- c(NA,diff(energia$consumo, differences = d))
} else {
  d <- 0
  energia$consumo_diff <- energia$consumo
}

create_harmonics <- function(x, period, K){
  t <- 1:length(x)
  harmonics <- data.frame(
    tsin = sin(2 * pi * K * t / period),
    tcos = cos(2 * pi * K * t / period)
  )
  return(harmonics)
}

K <- 1
harmonics <- create_harmonics(energia$consumo, s, K)

modelo <- lm(energia$consumo ~ harmonics$tsin + harmonics$tcos)

summary(modelo)
```

```
Call:
lm(formula = energia$consumo ~ harmonics$tsin + harmonics$tcos)

Residuals:
     Min       1Q   Median       3Q      Max 
-13,4634  -6,5747   0,4974   5,8703  11,5266 

Coefficients:
               Estimate Std. Error t value Pr(>|t|)    
(Intercept)    14,30075    0,35438  40,354   <2e-16 ***
harmonics$tsin  0,02458    0,50117   0,049    0,961    
harmonics$tcos  0,17362    0,50117   0,346    0,729    
---
Signif. codes:  0 '***' 0,001 '**' 0,01 '*' 0,05 '.' 0,1 ' ' 1

Residual standard error: 6,379 on 321 degrees of freedom
Multiple R-squared:  0,0003812, Adjusted R-squared:  -0,005847 
F-statistic: 0,06121 on 2 and 321 DF,  p-value: 0,9406
```

```r
sazonaldiff <- diff(energia$consumo, lag = s)
adf_test_sazonal <- aTSA::adf.test(sazonaldiff)
```

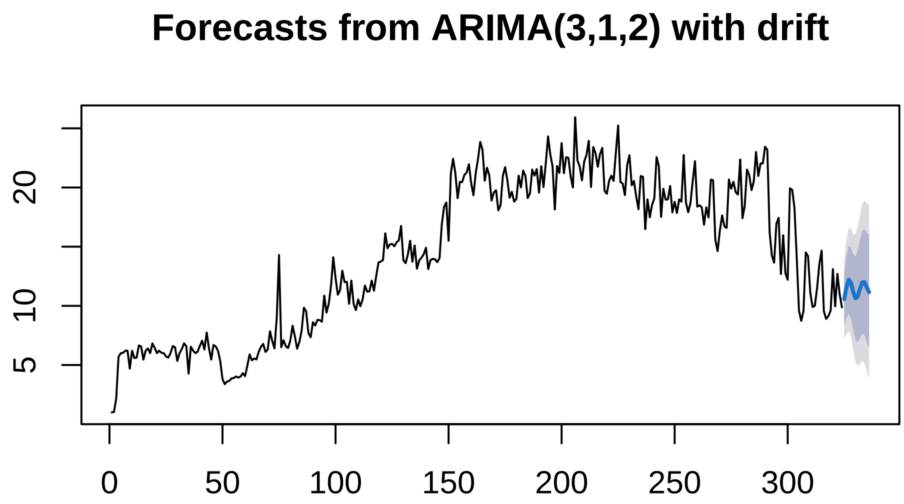
```
Augmented Dickey-Fuller Test 
alternative: stationary 
 
Type 1: no drift no trend 
     lag   ADF p.value
[1,]   0 -7,74    0,01
[2,]   1 -5,73    0,01
[3,]   2 -4,52    0,01
[4,]   3 -4,17    0,01
[5,]   4 -3,88    0,01
[6,]   5 -3,76    0,01
Type 2: with drift no trend 
     lag   ADF p.value
[1,]   0 -7,75    0,01
[2,]   1 -5,73    0,01
[3,]   2 -4,52    0,01
[4,]   3 -4,17    0,01
[5,]   4 -3,88    0,01
```

```
[6,]   5 -3,76    0,01
Type 3: with drift and trend
     lag   ADF p.value
[1,]   0 -8,12    0,01
[2,]   1 -5,96    0,01
[3,]   2 -4,69    0,01
[4,]   3 -4,39    0,01
[5,]   4 -4,12    0,01
[6,]   5 -4,03    0,01
----
Note: in fact, p.value = 0.01 means p.value <= 0.01
```

```r
p_value_adf_sazonal <- adf_test_sazonal$type1[1, "p.value"]

if(p_value_adf_sazonal > 0.05){D <- 1} else {D <- 0}

fit <- auto.arima(energia$consumo, seasonal = TRUE, D = D, d = d)

summary(fit)
```

```
Series: energia$consumo
ARIMA(3,1,2) with drift

Coefficients:
         ar1      ar2      ar3      ma1     ma2   drift
      0,7574  -0,6083  -0,2194  -1,1151  0,7591  0,0308
s.e.  0,0905   0,0695   0,0673   0,0764  0,0693  0,0570

sigma^2 = 2,941:  log likelihood = -629,94
AIC=1273,89   AICc=1274,25   BIC=1300,33

Training set error measures:
                       ME     RMSE      MAE        MPE     MAPE      MASE
Training set 0,0004634691 1,696347 1,213003 -0,7426017 9,649822 0,9109127
                    ACF1
Training set -0,02725481
```

```r
forecasts <- forecast(fit, h = 12)
plot(forecasts)
```



A série é estacionária após a primeira diferenciação.

Etapa 5

```
energia$variacao <- c(NA, diff(energia$consumo))
kable(mean(energia$variacao, na.rm=TRUE), align = "c",
    caption = "Variação do consumo")
```

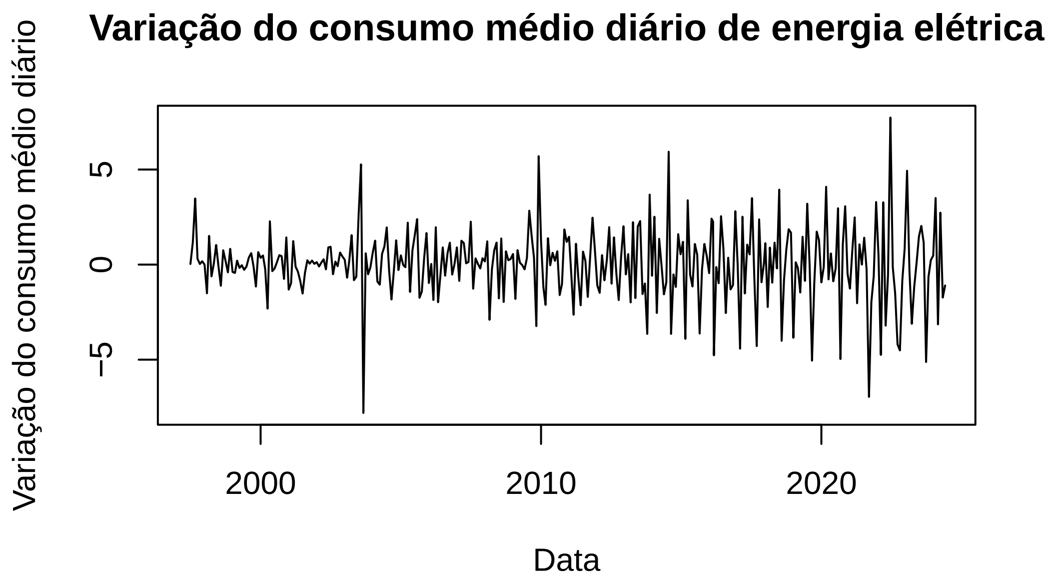
Table 3: Variação do consumo

x
0,027451

Essa transformação é necessária para que possamos analisar a variação do consumo de energia de um mês para o outro. E assim, identificar possíveis padrões de comportamento.

Etapa 6

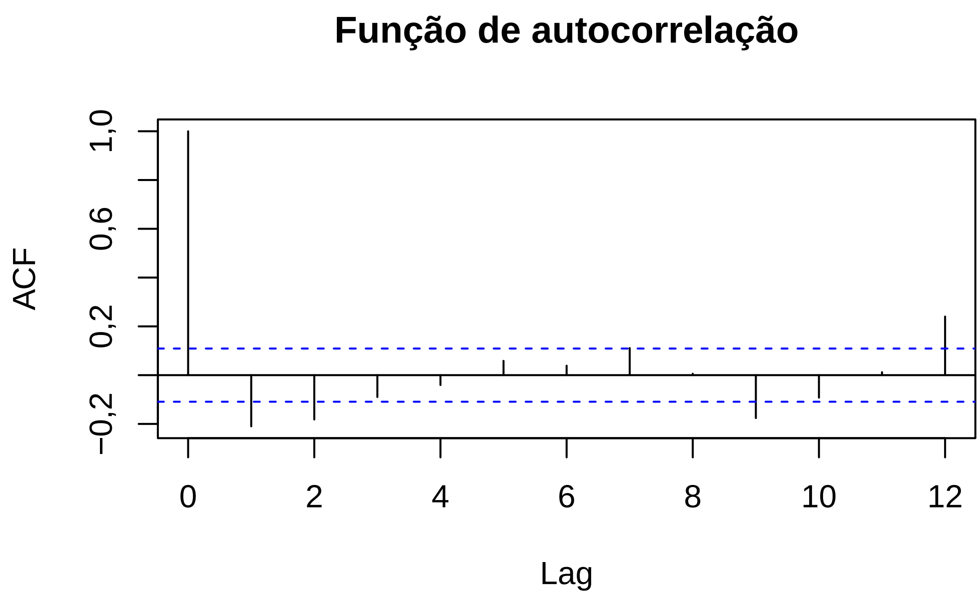
```
plot(energia$mes, energia$variacao, type = "l",
    main = "Variação do consumo médio diário de energia elétrica",
    xlab = "Data",
    ylab = "Variação do consumo médio diário")
```



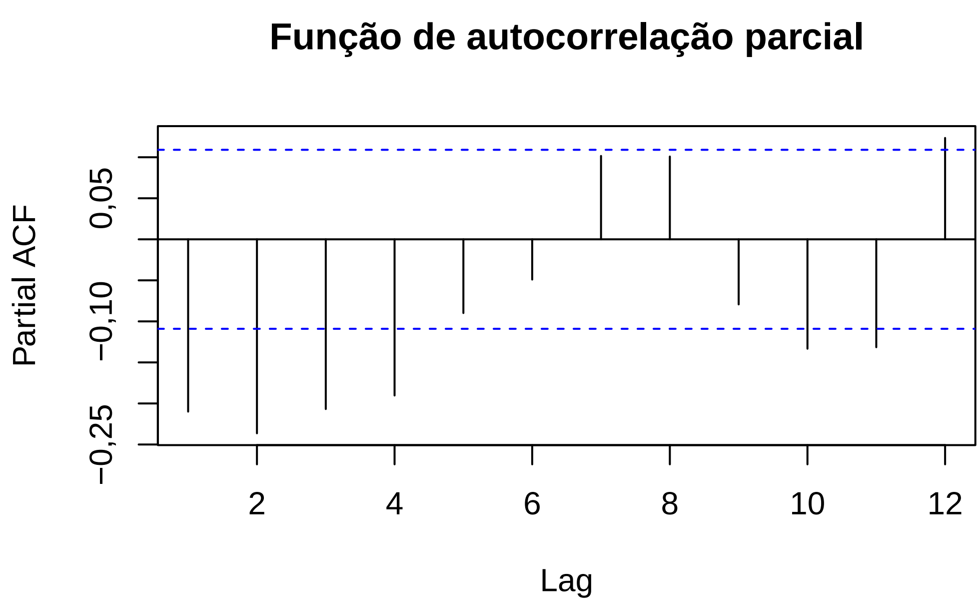
O gráfico apresenta a variação do consumo médio diário de energia elétrica ao longo do tempo. A série é composta por 60 observações, com periodicidade mensal.

Etapa 7

```
variacao <- na.omit(energia$variacao)

acf(variacao, lag.max = 12, main = "Função de autocorrelação")
```




```
pacf(variacao, lag.max = 12, main = "Função de autocorrelação parcial")
```



```r
adf.test(variacao)
```

```
Augmented Dickey-Fuller Test
alternative: stationary

Type 1: no drift no trend
     lag   ADF p.value
[1,]   0 -22,2    0,01
[2,]   1 -17,7    0,01
[3,]   2 -15,6    0,01
[4,]   3 -14,3    0,01
[5,]   4 -12,3    0,01
[6,]   5 -10,8    0,01
Type 2: with drift no trend
     lag   ADF p.value
[1,]   0 -22,1    0,01
[2,]   1 -17,6    0,01
[3,]   2 -15,6    0,01
[4,]   3 -14,3    0,01
[5,]   4 -12,3    0,01
[6,]   5 -10,7    0,01
Type 3: with drift and trend
     lag   ADF p.value
[1,]   0 -22,2    0,01
[2,]   1 -17,7    0,01
[3,]   2 -15,7    0,01
[4,]   3 -14,5    0,01
[5,]   4 -12,5    0,01
[6,]   5 -11,0    0,01
----
Note: in fact, p.value = 0.01 means p.value <= 0.01
```

Amostras de treinamento e validação

```r
m <- round(nrow(energia)*0.8,0)
n <- nrow(energia)

treinamento <- energia[1:m,]
validacao <- energia[(m+1):n,]
```

Etapa 8

Nesse caso, temos dados mensais, logo assume-se que a ordem sazonal é 12 ($s = 12$).

```
adf_test <- aTSA::adf.test(treinamento$consumo)
```

```
Augmented Dickey-Fuller Test
alternative: stationary

Type 1: no drift no trend
     lag    ADF p.value
[1,]   0 -0,257   0,570
[2,]   1  0,157   0,689
[3,]   2  0,484   0,783
[4,]   3  0,789   0,871
[5,]   4  0,885   0,899
Type 2: with drift no trend
     lag   ADF p.value
[1,]   0 -2,26   0,225
[2,]   1 -1,87   0,379
[3,]   2 -1,61   0,481
[4,]   3 -1,27   0,600
[5,]   4 -1,13   0,652
Type 3: with drift and trend
     lag   ADF p.value
[1,]   0 -5,05  0,0100
[2,]   1 -3,80  0,0194
[3,]   2 -2,87  0,2078
[4,]   3 -2,07  0,5460
[5,]   4 -1,85  0,6372
----
Note: in fact, p.value = 0.01 means p.value <= 0.01
```

```
p_value_adf <- adf_test$type1[1, "p.value"]

if(p_value_adf > 0.05){
  d <- 1
  treinamento$consumo_diff <- c(NA,diff(treinamento$consumo, differences = d))
} else {
  d <- 0
  treinamento$consumo_diff <- treinamento$consumo
```

```r
}

d
```

```
[1] 1
```

```r
sazonal_diff <- diff(treinamento$consumo, lag = s)
adf_test_sazonal <- aTSA::adf.test(sazonal_diff)
```

```
Augmented Dickey-Fuller Test
alternative: stationary

Type 1: no drift no trend
     lag   ADF p.value
[1,]   0 -7,27    0,01
[2,]   1 -5,20    0,01
[3,]   2 -4,15    0,01
[4,]   3 -3,43    0,01
[5,]   4 -3,49    0,01
Type 2: with drift no trend
     lag   ADF p.value
[1,]   0 -7,57    0,01
[2,]   1 -5,41    0,01
[3,]   2 -4,32    0,01
[4,]   3 -3,60    0,01
[5,]   4 -3,70    0,01
Type 3: with drift and trend
     lag   ADF p.value
[1,]   0 -7,57  0,0100
[2,]   1 -5,38  0,0100
[3,]   2 -4,28  0,0100
[4,]   3 -3,57  0,0357
[5,]   4 -3,70  0,0246
----
Note: in fact, p.value = 0.01 means p.value <= 0.01
```

```r
p_value_adf_sazonal <- adf_test_sazonal$type1[1, "p.value"]

if(p_value_adf_sazonal > 0.05){
  D <- 1
```

```r
} else {
  D <- 0
}

D
```

```
[1] 0
```

Etapa 9

```r
p_values <- 0:2
q_values <- 0:2
P_values <- 0:1
Q_values <- 0:1

calculate_bic <- function(x, p, d, q, P, D, Q, s) {
  model <- tryCatch({
    sarima(x, p, d, q, P, D, Q, s, details = FALSE)
  }, error = function(e) {
    return(NA)
  })
  return(model$BIC)
}

resultados <- list()

for (p in p_values) {
  for (q in q_values) {
    for (P in P_values) {
      for (Q in Q_values) {
        modelo <- Arima(treinamento$consumo,
        order = c(p, d, q), seasonal = c(P, D, Q, s))
        bic_value <- BIC(modelo)
        resultados[[paste(p, q, P, Q, sep = "_")]] <- bic_value
      }
    }
  }
}

resultados_df <- do.call(rbind, lapply(names(resultados), function(x) {
```

```r
  parts <- unlist(strsplit(x, "_"))
  data.frame(p = as.numeric(parts[1]), q = as.numeric(parts[2]),
             P = as.numeric(parts[3]), Q = as.numeric(parts[4]),
             BIC = resultados[[x]])
}))

kable(resultados_df, align = "c",
    caption = "Head - Valores do critério de informação bayesiano (BIC)")
```

Table 4: Head - Valores do critério de informação bayesiano (BIC)

p	q	P	Q	BIC
0	0	0	0	996,0140
0	0	0	1	996,0140
0	0	1	0	996,0140
0	0	1	1	996,0140
0	1	0	0	955,0395
0	1	0	1	955,0395
0	1	1	0	955,0395
0	1	1	1	955,0395
0	2	0	0	954,1273
0	2	0	1	954,1273
0	2	1	0	954,1273
0	2	1	1	954,1273
1	0	0	0	978,1264
1	0	0	1	978,1264
1	0	1	0	978,1264
1	0	1	1	978,1264
1	1	0	0	955,3682
1	1	0	1	955,3682
1	1	1	0	955,3682
1	1	1	1	955,3682
1	2	0	0	959,6475
1	2	0	1	959,6475
1	2	1	0	959,6475
1	2	1	1	959,6475
2	0	0	0	966,7430
2	0	0	1	966,7430
2	0	1	0	966,7430
2	0	1	1	966,7430
2	1	0	0	958,1620

p	q	P	Q	BIC
2	1	0	1	958,1620
2	1	1	0	958,1620
2	1	1	1	958,1620
2	2	0	0	960,1000
2	2	0	1	960,1000
2	2	1	0	960,1000
2	2	1	1	960,1000

Etapa 10

```
melhor_modelo <- resultados_df[resultados_df$BIC == min(resultados_df$BIC),]

kable(melhor_modelo, align = "c",
    caption = "Melhor modelo")
```

Table 5: Melhor modelo

	p	q	P	Q	BIC
9	0	2	0	0	954,1273
10	0	2	0	1	954,1273
11	0	2	1	0	954,1273
12	0	2	1	1	954,1273

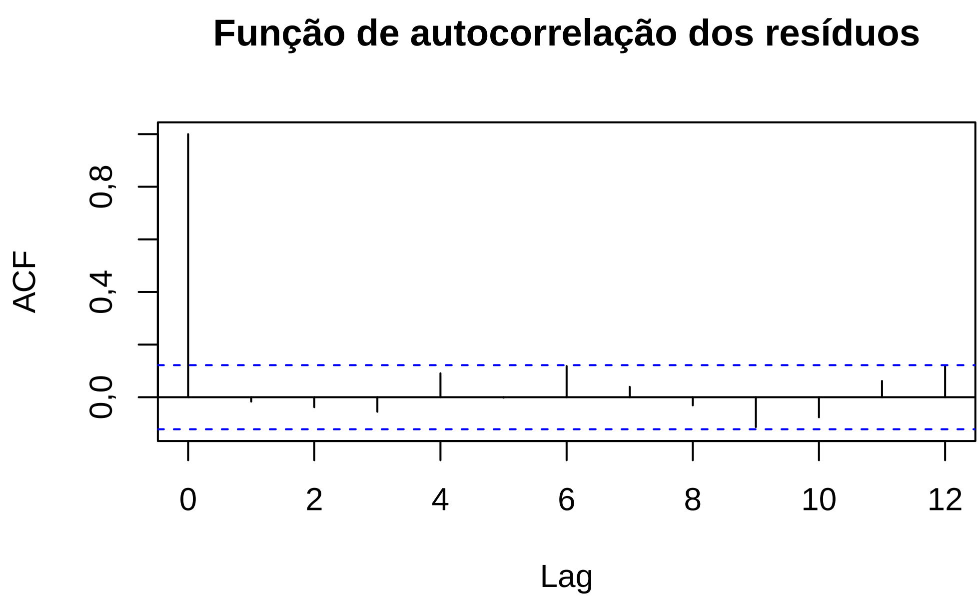
Etapa 11

```
modelo_final <- Arima(treinamento$consumo,
order = c(melhor_modelo$p[1], d, melhor_modelo$q[1]),
seasonal = c(melhor_modelo$P[1], D, melhor_modelo$Q[1], s))

summary(modelo_final)
```

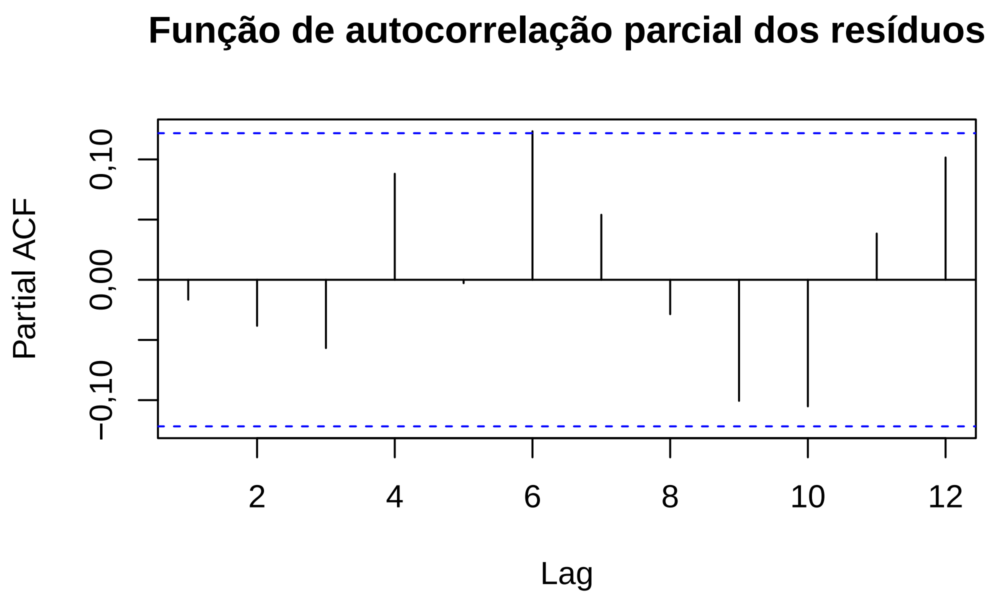
```
Series: treinamento$consumo
ARIMA(0,1,2)

Coefficients:
          ma1      ma2
```

```
      -0,4476  -0,1510
s.e.   0,0614   0,0579

sigma^2 = 2,23:  log likelihood = -468,73
AIC=943,47   AICc=943,56   BIC=954,13

Training set error measures:
                    ME     RMSE      MAE       MPE   MAPE      MASE        ACF1
Training set 0,1848532 1,48469 1,067609 0,9884692 8,8736 0,9101472 -0,01653426
```

```r
residuos <- residuals(modelo_final)

acf(residuos, lag.max = 12, main = "Função de autocorrelação dos resíduos")
```




```r
pacf(residuos, lag.max = 12, main = "Função de autocorrelação parcial dos resíduos")
```




```r
Box.test(residuos, lag = 12, type = "Ljung-Box")
```

```
    Box-Ljung test

data:  residuos
X-squared = 17,862, df = 12, p-value = 0,12
```

```r
shapiro.test(residuos)
```

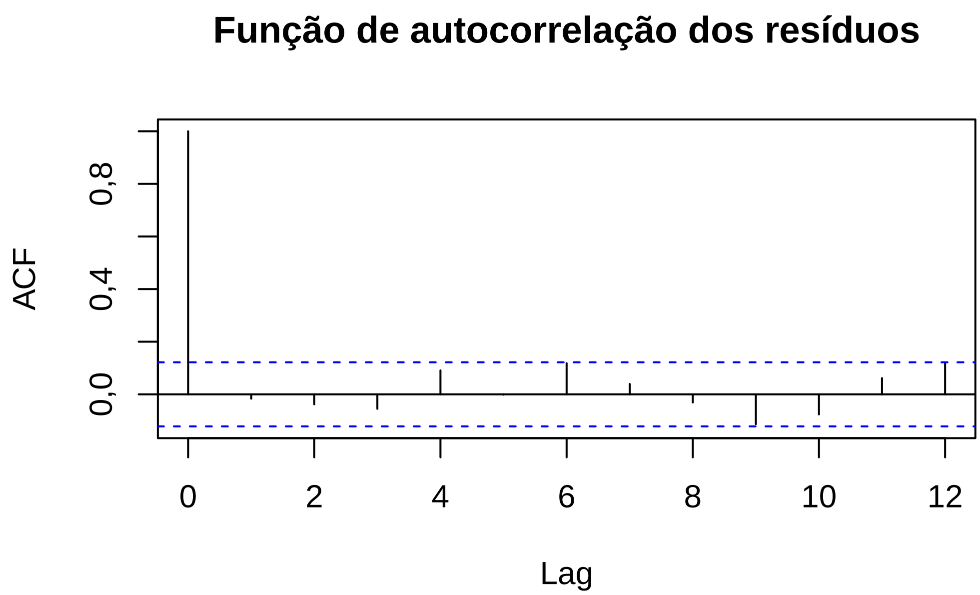
```
    Shapiro-Wilk normality test

data:  residuos
W = 0,96985, p-value = 2,761e-05
```

Não sendo um bom modelo, testa-se o segundo:

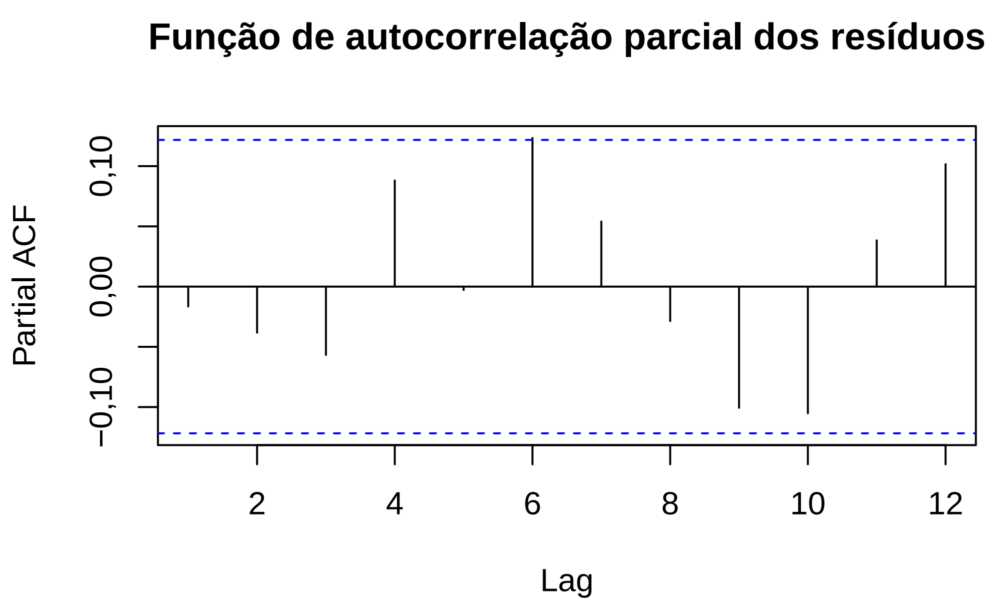
```r
modelo_final <- Arima(treinamento$consumo,
order = c(melhor_modelo$p[2], d, melhor_modelo$q[2]),
seasonal = c(melhor_modelo$P[2], D, melhor_modelo$Q[2], s))

summary(modelo_final)
```

```
Series: treinamento$consumo
ARIMA(0,1,2)

Coefficients:
          ma1      ma2
      -0,4476  -0,1510
s.e.   0,0614   0,0579

sigma^2 = 2,23:  log likelihood = -468,73
AIC=943,47   AICc=943,56   BIC=954,13

Training set error measures:
                    ME    RMSE      MAE       MPE   MAPE      MASE        ACF1
Training set 0,1848532 1,48469 1,067609 0,9884692 8,8736 0,9101472 -0,01653426
```

```r
residuos <- residuals(modelo_final)

acf(residuos, lag.max = 12, main = "Função de autocorrelação dos resíduos")
```




```r
pacf(residuos, lag.max = 12, main = "Função de autocorrelação parcial dos resíduos")
```



```r
Box.test(residuos, lag = 12, type = "Ljung-Box")
```

```
	Box-Ljung test

data:  residuos
X-squared = 17,862, df = 12, p-value = 0,12
```

```r
shapiro.test(residuos)
```

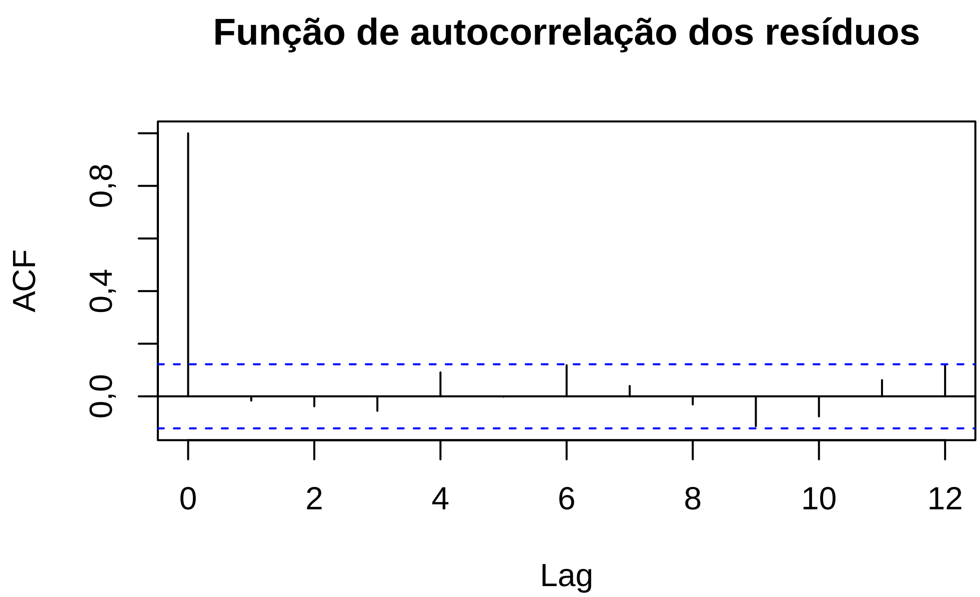
```
	Shapiro-Wilk normality test

data:  residuos
W = 0,96985, p-value = 2,761e-05
```

Ainda não é um bom modelo, testa-se o terceiro:

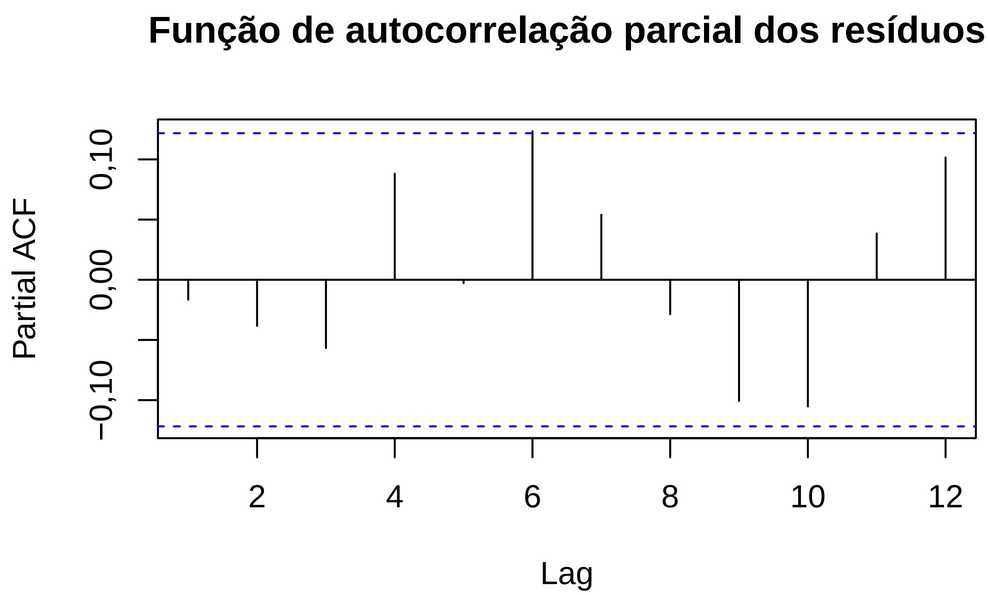
```r
modelo_final <- Arima(treinamento$consumo,
order = c(melhor_modelo$p[3], d, melhor_modelo$q[3]),
seasonal = c(melhor_modelo$P[3], D, melhor_modelo$Q[3], s))

summary(modelo_final)
```

```
Series: treinamento$consumo
ARIMA(0,1,2)

Coefficients:
          ma1      ma2
      -0,4476  -0,1510
s.e.   0,0614   0,0579

sigma^2 = 2,23:  log likelihood = -468,73
AIC=943,47   AICc=943,56   BIC=954,13

Training set error measures:
                    ME    RMSE      MAE       MPE   MAPE      MASE        ACF1
Training set 0,1848532 1,48469 1,067609 0,9884692 8,8736 0,9101472 -0,01653426
```

```r
residuos <- residuals(modelo_final)

acf(residuos, lag.max = 12, main = "Função de autocorrelação dos resíduos")
```




```r
pacf(residuos, lag.max = 12, main = "Função de autocorrelação parcial dos resíduos")
```




```
Box.test(residuos, lag = 12, type = "Ljung-Box")
```

```

	Box-Ljung test

data:  residuos
X-squared = 17,862, df = 12, p-value = 0,12
```

```
shapiro.test(residuos)
```

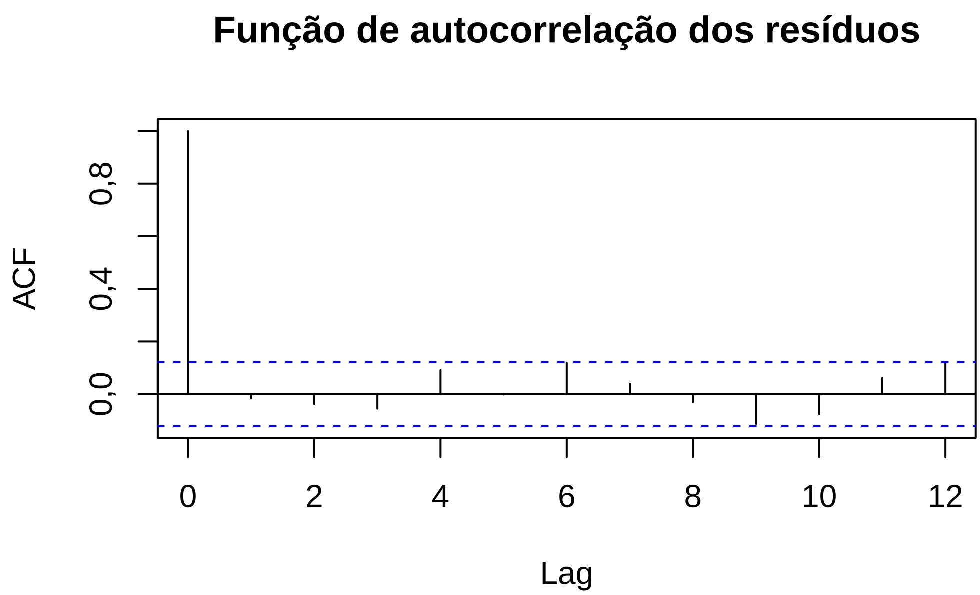
```

	Shapiro-Wilk normality test

data:  residuos
W = 0,96985, p-value = 2,761e-05
```

Ainda não é um bom modelo, testa-se o quarto:

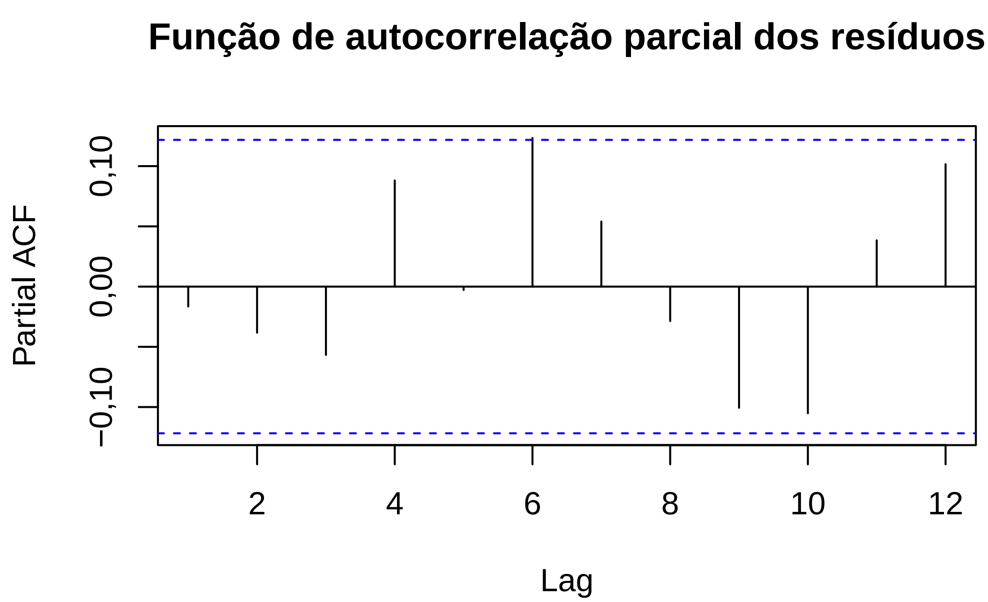
```r
modelo_final <- Arima(treinamento$consumo,
order = c(melhor_modelo$p[4], d, melhor_modelo$q[4]),
seasonal = c(melhor_modelo$P[4], D, melhor_modelo$Q[4], s))

summary(modelo_final)
```

```
Series: treinamento$consumo
ARIMA(0,1,2)

Coefficients:
          ma1      ma2
      -0,4476  -0,1510
s.e.   0,0614   0,0579

sigma^2 = 2,23:  log likelihood = -468,73
AIC=943,47   AICc=943,56   BIC=954,13

Training set error measures:
                    ME    RMSE      MAE       MPE   MAPE      MASE        ACF1
Training set 0,1848532 1,48469 1,067609 0,9884692 8,8736 0,9101472 -0,01653426
```

```r
residuos <- residuals(modelo_final)

acf(residuos, lag.max = 12, main = "Função de autocorrelação dos resíduos")
```




```r
pacf(residuos, lag.max = 12, main = "Função de autocorrelação parcial dos resíduos")
```



```r
Box.test(residuos, lag = 12, type = "Ljung-Box")
```

```
	Box-Ljung test

data:  residuos
X-squared = 17,862, df = 12, p-value = 0,12
```

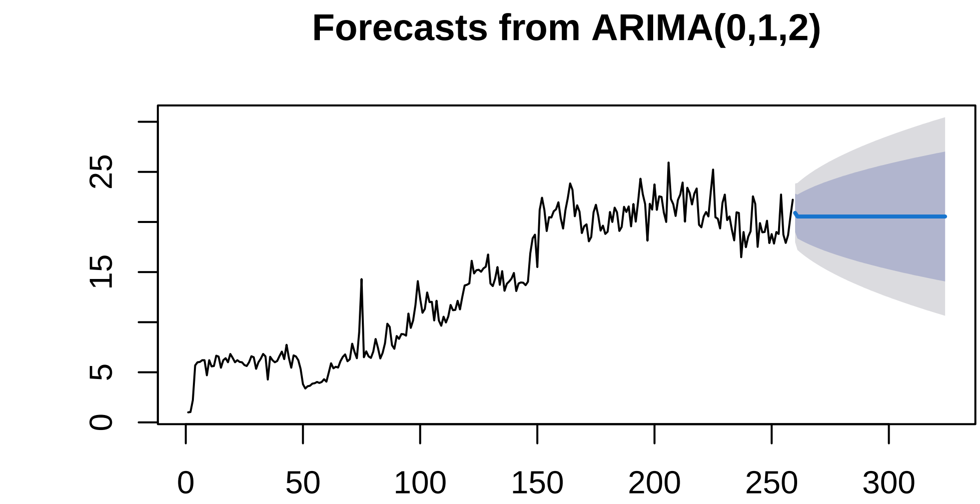
```r
shapiro.test(residuos)
```

```
	Shapiro-Wilk normality test

data:  residuos
W = 0,96985, p-value = 2,761e-05
```

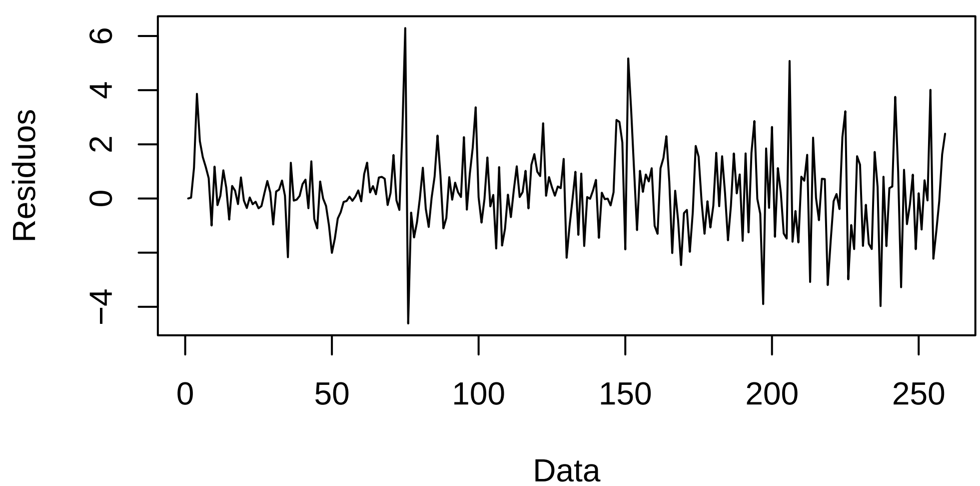
Nenhum dos modelos testados é adequado. Considerando que as ordens s, d e D estão corretas, é possível que o modelo SARIMA não seja apropriado.

Etapa 12

```r
previsao <- forecast(modelo_final, h = nrow(validacao))
plot(previsao)
```




```
erros <- resid(modelo_final)
plot(erros, type = 'l', main = "", xlab = "Data", ylab = "Residuos")
```



```
mape <- mean(abs((validacao$consumo - previsao$mean)/validacao$consumo))
mape
```

```
[1] 0,4053458
```

Logo, utilizando a referência de MAPE, o modelo é considerado razoável/aceitável, pois se apresenta em 40%.

Etapa 13

```
#previsao <- sarima.for(treinamento$consumo, n.ahead = 12, model = modelo_final, d=1, D=0,
#plot(previsao)
```

As previsões são limitadas pela hipótese de normalidade residual. Caso a hipótese seja descartada, é possível que o modelo não seja adequado, como foi observado durante esse estudo.

Logo, também não é possível realizar a previsão para os próximos 12 meses.