Trabalho - Métodos Estatísticos de Previsão

2023-11-18

Isabelle Fernandes de Oliveira
Mariana Rodrigues Fontenelle 2021035454
Vitor Marques Rodrigues 2021035470

1 - Descrição dos Dados

1.1 - O Banco de Dados

A série escolhida consiste na temperatura mínima diária na cidade de Melbourne, Austrália de 01/01/1981 a 30/12/1990.

Resolvemos trabalhar com a frequência mensal ao invés da diária, para isso tomamos a temperatura média de cada mês. Logo, as características da série final são:

- **Resposta**: Temperatura Mínima Média (Grau Celsius)
- **Frequência**: Mensal
- **Período**: jan/1981 - dez/1990
- **Número de Observações**: 120
- **Fonte**: *https://www.kaggle.com/datasets/ingwangdk/minimum-daily-temperatures-in-melbourne-10-years/data?select=daily-minimum-temperatures.csv*

Carregando o banco de dados, convertendo para frequência mensal e dividindo em Train/Test

Separando as 12 últimas observações para o conjunto teste

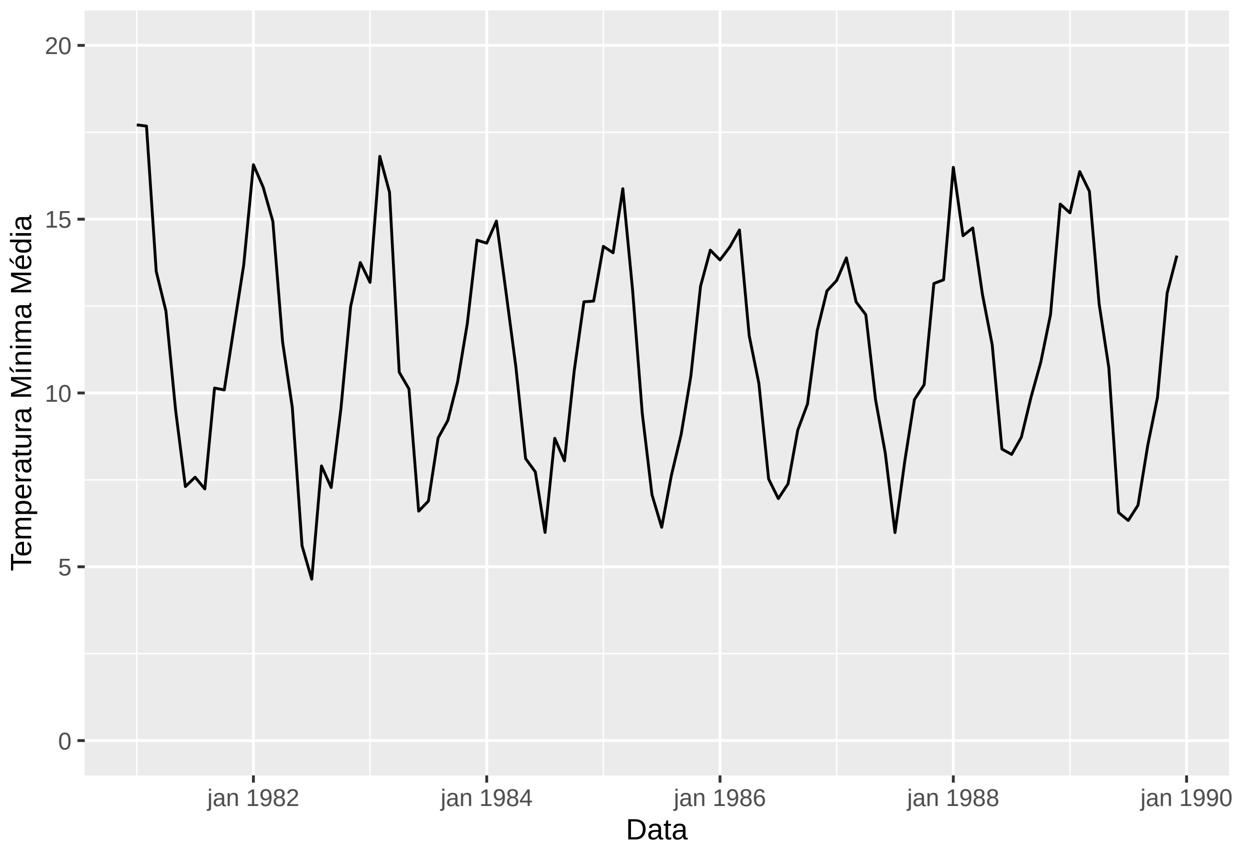
```r
library(tidyverse)
library(zoo)
library(lmtest)

# Série Diária (Temperatura Mínima Diária)
melbourne_daily <- read_csv("daily-min-temperatures.csv")
colnames(melbourne_daily) <- c("data", "temperatura")

# Série Mensal - (Temperatura Mínima Mensal Média)
melbourne_monthly <- melbourne_daily %>%
  mutate(data = as.yearmon(data)) %>%
  group_by(data) %>%
  summarize(temperatura = mean(temperatura))


n <- nrow(melbourne_monthly)

# Seprando em Train/Test, 12 últimas observações para teste

train <- melbourne_monthly[1:(n-12), ]
test <- melbourne_monthly[(n-11):n, ]
```

```r
head(train)
```

```
## # A tibble: 6 x 2
##   data      temperatura
##   <yearmon>       <dbl>
## 1 jan 1981        17.7
## 2 fev 1981        17.7
## 3 mar 1981        13.5
## 4 abr 1981        12.4
## 5 mai 1981         9.49
## 6 jun 1981         7.31
```

1.2 - Visualizando a Série

```r
train %>%
  ggplot(aes(x=data, y=temperatura))+
  geom_line()+
  labs(x='Data', y='Temperatura Mínima Média')+
  ylim(c(0, 20))
```



Por inspeção visual, podemos notar que:

- **Sazonalidade**: A série aparenta ter comportamente sazonal de 12 em 12 meses, o que é esperado uma vez que a temperatura é afetada pela translação da Terra em torno do Sol, que possui período de um ano.
- **Estacionariedade**: A série não aparenta ter tendência na parte simples nem sazonal, indicando possível

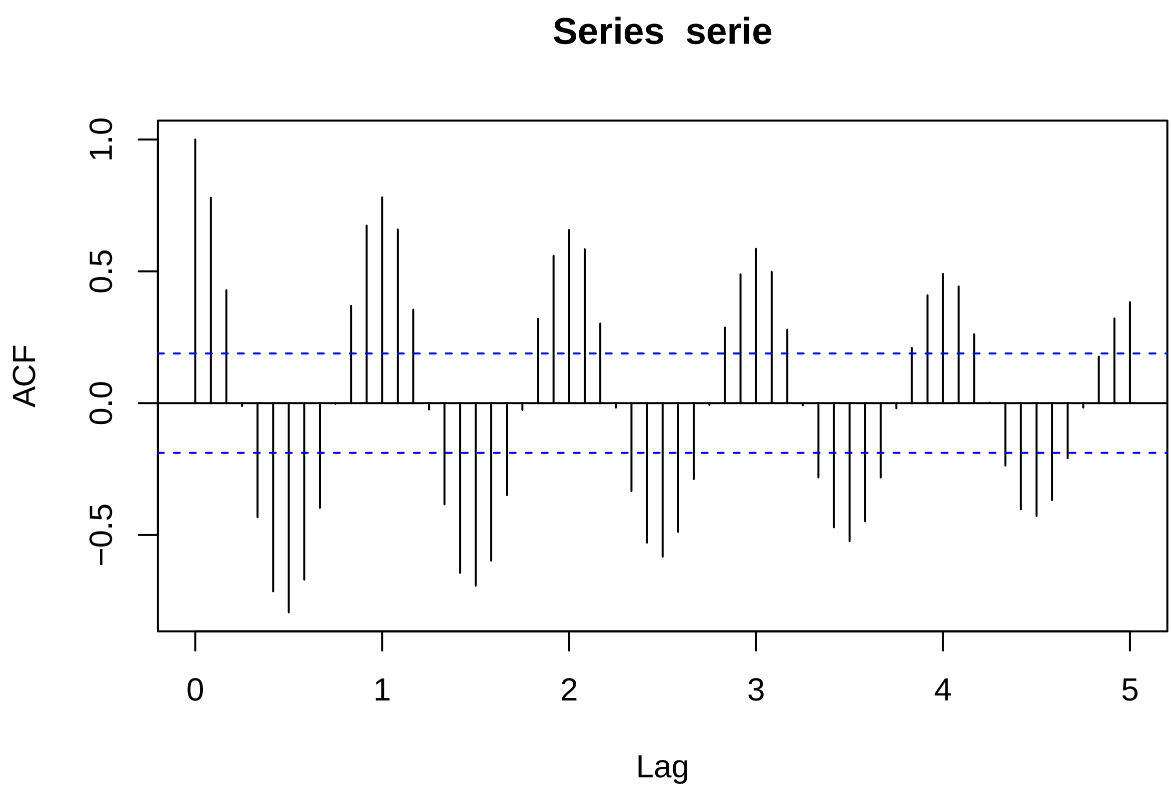
estacionariedade.

1.3 - ACF e PACF

```r
# Período de 12 meses
serie <- ts(train$temperatura, frequency=12, start = 1981)
```

ACF

```r
# Limitando a cinco períodos
acf(serie, lag.max=5*12)
```

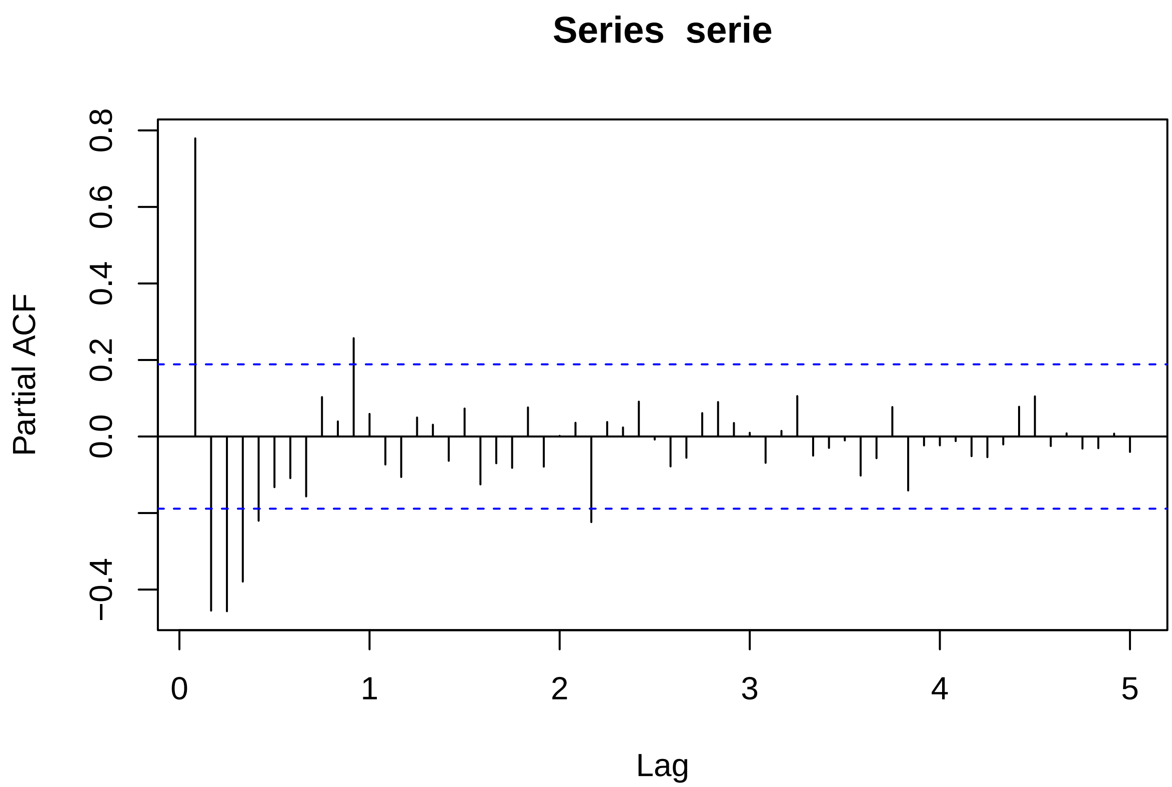


Podemos notar que:

- O módulo das correlações não estão próximos de 1, confirmando a hipótese de estacionariedade da série. **Não há necessidade de diferenciar a série, na parte simples nem sazonal**
- O módulo das correlações aparentam decrescer exponencialmente conforme o aumento do Lag, tanto na parte simples quanto na sazonal. **Improvável ter somente parte média móvel na parte simples e sazonal**

PACF

```r
# Limitando a cinco períodos
pacf(serie, lag.max=5*12)
```



Podemos notar que:

- Picos significativos nos primeiros lags (1,2,3,4 e 5). **Junto com a informação da ACF, pode sugerir um AR de ordem maior, ou um ARMA se interpretarmos esse fenômeno como um decrescimento exponencial**
- Dois picos significativos próximos da parte sazonal lags (11 e 14, período 1 e 2). **Junto com a informação da ACF, pode sugerir um AR(2) na parte sazonal**

2 - ARIMA

2.1 - Ajuste do modelo

```
#Sobrefixo AR(2) parte sazonal
M1 <- arima(serie, order = c(2, 0, 0), seasonal = list(order = c(2, 0, 0)))

M2 <-  arima(serie, order = c(1, 0, 1), seasonal = list(order = c(2, 0, 0)))

M3 <-  arima(serie, order = c(4, 0, 0), seasonal = list(order = c(2, 0, 0)))

M4 <-  arima(serie, order = c(0, 0, 2), seasonal = list(order = c(2, 0, 0)))


#Sobrefixo AR(1) parte sazonal
M5 <- arima(serie, order = c(2, 0, 0), seasonal = list(order = c(1, 0, 0)))

M6 <- arima(serie, order = c(1, 0, 1), seasonal = list(order = c(1, 0, 0)))
```

```r
M7 <- arima(serie, order = c(4, 0, 0), seasonal = list(order = c(1, 0, 0)))

M8 <- arima(serie, order = c(0, 0, 2), seasonal = list(order = c(1, 0, 0)))


#Sobrefixo ARRIMA(1,0,1) parte sazonal
M9 <- arima(serie, order = c(2, 0, 0), seasonal = list(order = c(1, 0, 1)))

M10 <- arima(serie, order = c(1, 0, 1), seasonal = list(order = c(1, 0, 1)))

M11 <- arima(serie, order = c(4, 0, 0), seasonal = list(order = c(1, 0, 1)))

M12 <- arima(serie, order = c(0, 0, 2), seasonal = list(order = c(1, 0, 1)))


AIC <- c(M1$aic,M2$aic,M3$aic,M4$aic,M5$aic,M6$aic,M7$aic,M8$aic,M9$aic,M10$aic,M11$aic,M12$aic)

Modelos <- rep(1:12)

resultado <- as.data.frame(cbind(Modelos, AIC)) %>%
  arrange(AIC)

resultado
```

```
##    Modelos      AIC
## 1       11 353.2398
## 2        9 354.1635
## 3       12 354.5645
## 4       10 356.0088
## 5        3 362.2953
## 6        7 364.0960
## 7        1 367.0374
## 8        4 368.1067
## 9        2 370.4584
## 10       5 380.2546
## 11       8 381.0633
## 12       6 384.8201
```

Pelo resultado, o modelo que possui menor AIC é o modelo 11

```r
coeftest(M11)
```

```
##
## z test of coefficients:
##
##             Estimate Std. Error  z value  Pr(>|z|)
## ar1        0.4315284  0.1338099   3.2249  0.001260 **
## ar2        0.3073368  0.1090463   2.8184  0.004826 **
## ar3       -0.1238241  0.1192560  -1.0383  0.299128
## ar4       -0.2583080  0.1375639  -1.8777  0.060418 .
## sar1       0.9996562  0.0020629 484.5927 < 2.2e-16 ***
## sma1      -0.9686252  0.0923167 -10.4924 < 2.2e-16 ***
## intercept 11.1613802  0.5599468  19.9329 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Contudo, este modelo apresenta parâmetros não significativos. O modelo com o segundo menor AIC é o modelo 9. Embora o modelo 12 tenha AIC próximo do modelo 9, ambos tem o mesmo grau de complexidade (parcimonioso). Então será analisado o modelo 9, e caso ele não tenha bom ajuste, analisaremos o modelo 12.

```
coeftest(M9)
```

```
##
## z test of coefficients:
##
##             Estimate Std. Error  z value  Pr(>|z|)
## ar1        0.3082407  0.1062156   2.9020  0.003708 **
## ar2        0.2037504  0.1009014   2.0193  0.043456 *
## sar1       0.9990276  0.0060436 165.3039 < 2.2e-16 ***
## sma1      -0.9039012  0.2925114  -3.0901  0.002001 **
## intercept 11.1938237  1.2908665   8.6716 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

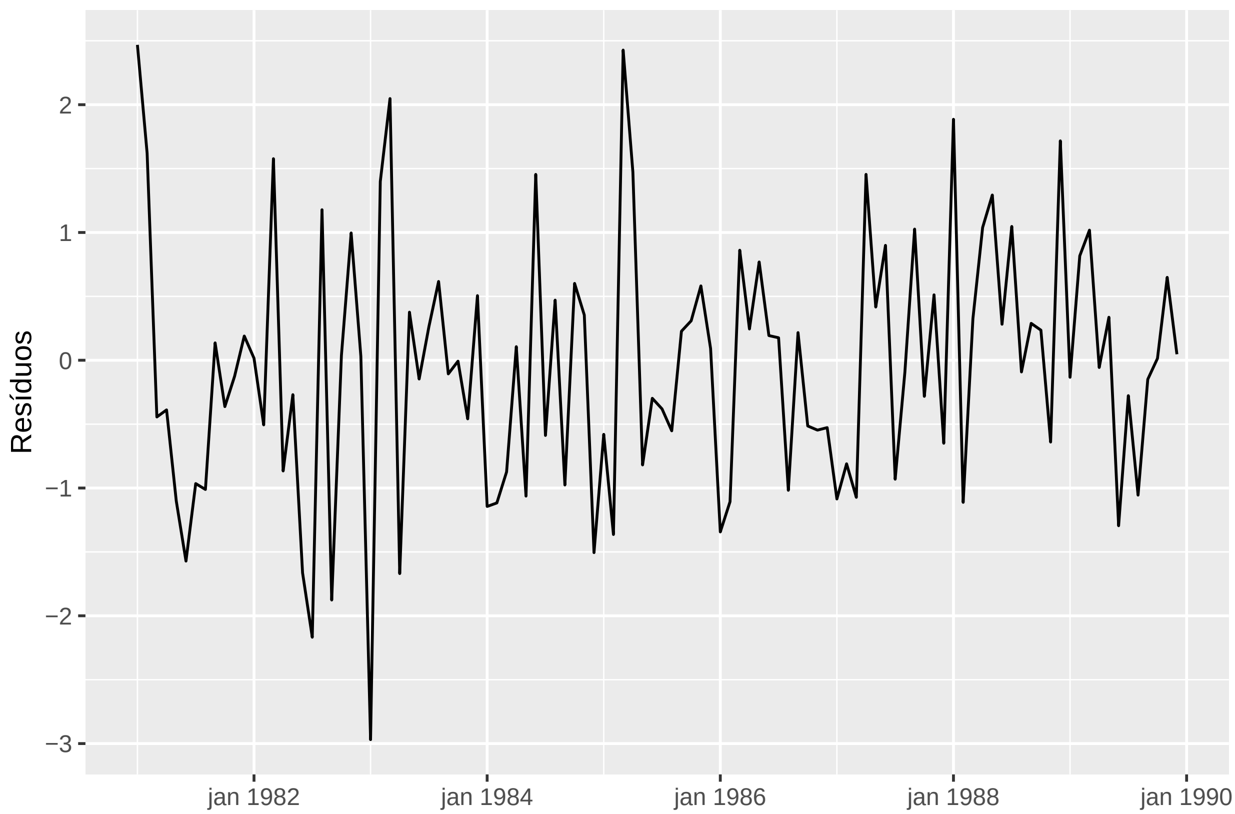
Os coeficientes deram significativos para o modelo 9.

2.2 - Análise de resíduos

Verificar se a variância é constante

```
train %>%
  mutate(Resíduos = M9$res) %>%
  ggplot(aes(x=data, y=Resíduos))+
  geom_line()+
  labs(x='', y='Resíduos')
```

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



Os resíduos parecem evoluir de forma aleatória em torno de zero e com variância constante.

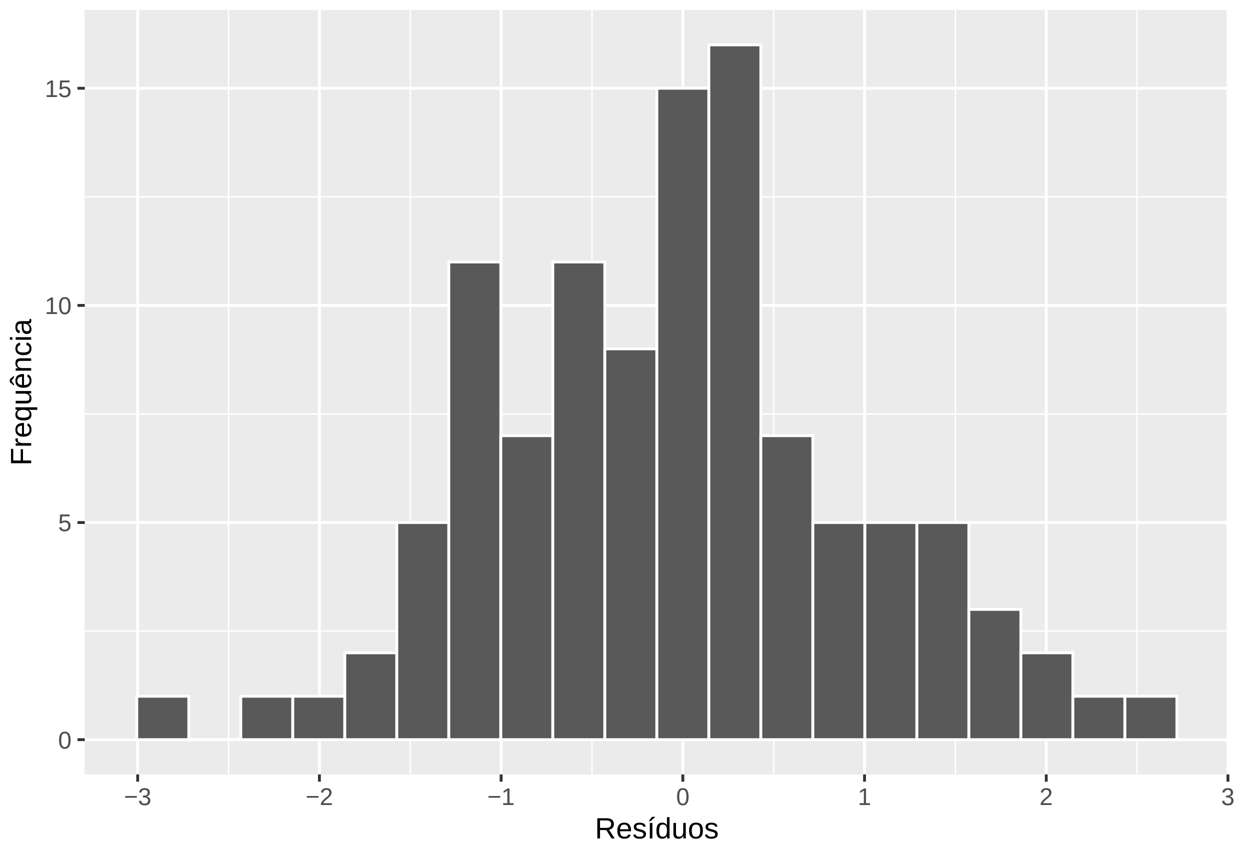
Verificar normalidade dos resíduos

```
shapiro.test(M12$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  M12$residuals
## W = 0.99055, p-value = 0.6589
```

```
train %>%
  mutate(Resíduos = M9$res) %>%
  ggplot(aes(x=Resíduos))+
  geom_histogram(color = "white",bins = 20) +
  labs(y='Frequência')
```

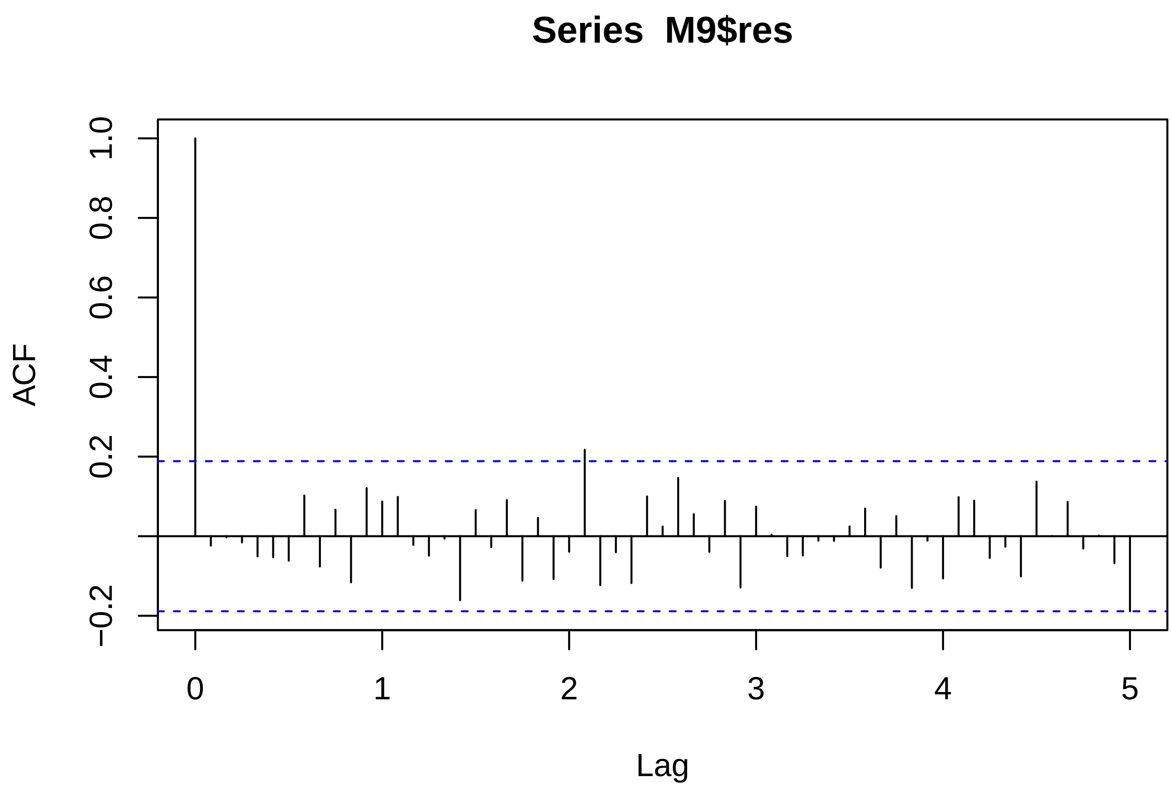
```
## Don't know how to automatically pick scale for object of type <ts>. Defaulting
## to continuous.
```



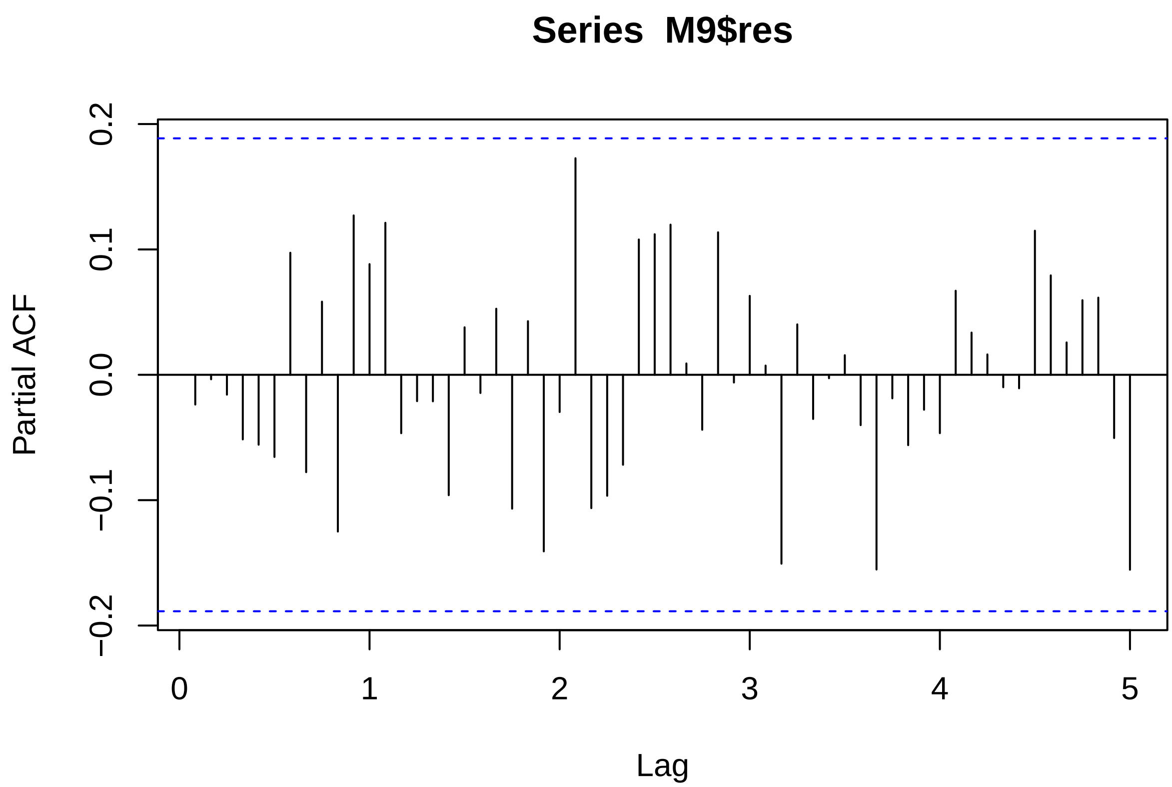
Os resíduos apresentam distribuição normal, como pode ser visto pelo valor-p no teste de Shapiro-Wilk e histograma.

Verificar independência dos resíduos

```
acf(M9$res, lag.max = 5*12)
```




```r
pacf(M9$res, lag.max = 5*12)
```




```
Box.test(M9$res, lag=1, type = c("Box-Pierce", "Ljung-Box"), fitdf = 0)
```

```
##
##  Box-Pierce test
##
## data:  M9$res
## X-squared = 0.060819, df = 1, p-value = 0.8052
```

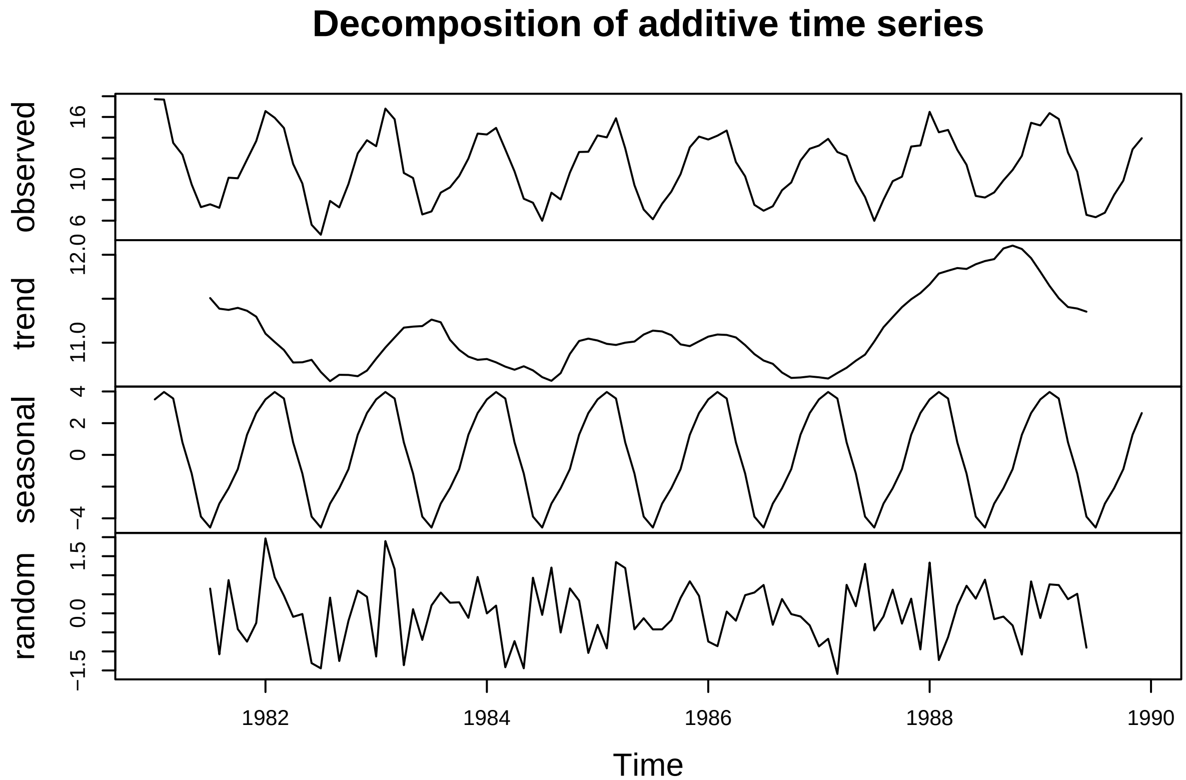
A ACF e PACF dos resíduos apresentam um comportamento de ruído branco (o pico no lag 25 ACF pode ser apenas ruído e não é motivo de preocupação).O teste de Ljung e Box também confirma a hipótese de ruído branco, pois o valor-p foi maior que 0,05.

O ajuste do modelo 9, parte simples (2,0,0) e parte sazonal (1,0,1), atendeu aos requisitos de significância dos coefientes e análise dos resíduos. Portanto, foi o modelo ARIMA escolhido.

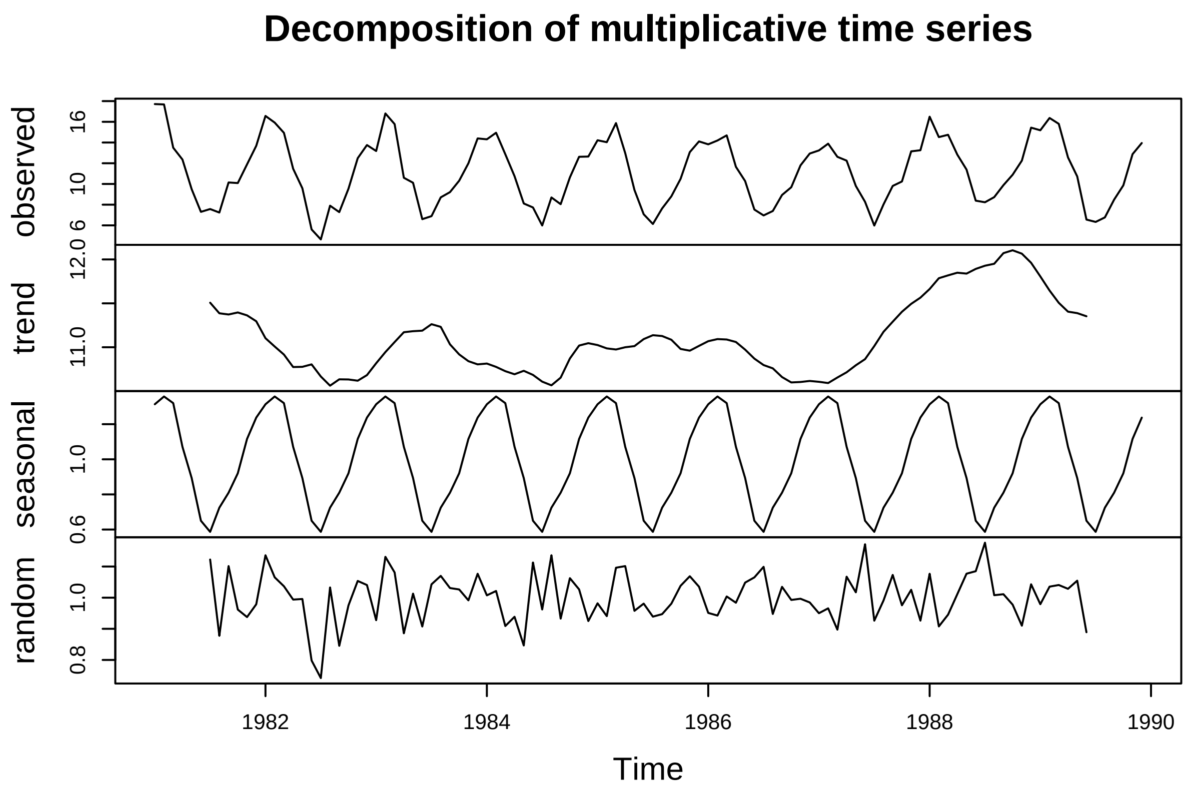
3 - Alisamento Exponencial

3.1 - Seleção do Método

```
plot(decompose(serie, type = c("additive")))
```




```r
plot(decompose(serie, type = c("multiplicative")))
```



Como a série apresenta forte sazonalidade, como observado no gráfico da série, usaremos o Alisamento Exponencial de Holt-Winters. Faremos a Comparação dos modelos Aditivo e Multiplicativo para descobrir qual se adequa melhor à série.

3.2 AEHW Aditivo

```
aehw.aditivo <- HoltWinters(serie, alpha = NULL, beta = NULL,
                            gamma = NULL, seasonal = c("additive"))
aehw.aditivo
```

```
## Holt-Winters exponential smoothing with trend and additive seasonal component.
##
## Call:
## HoltWinters(x = serie, alpha = NULL, beta = NULL, gamma = NULL,     seasonal = c("additive"))
##
## Smoothing parameters:
##  alpha: 0.2100695
##  beta : 0.0252934
##  gamma: 0.4271307
##
## Coefficients:
##            [,1]
## a   10.71496243
## b   -0.01299015
## s1   3.88822890
## s2   4.08506296
```

```
## s3   3.78330922
## s4   1.33163512
## s5  -0.56383797
## s6  -3.66324099
## s7  -4.21889552
## s8  -3.28026746
## s9  -1.78091873
## s10 -0.63171778
## s11  1.75808298
## s12  3.14269808
```

3.3 - AEHW Multiplicativo

```
aehw.multiplicativo <- HoltWinters(serie, alpha=NULL, beta=NULL,
                                   gamma=NULL, seasonal = c("multiplicative"))
aehw.multiplicativo
```

```
## Holt-Winters exponential smoothing with trend and multiplicative seasonal component.
##
## Call:
## HoltWinters(x = serie, alpha = NULL, beta = NULL, gamma = NULL,     seasonal = c("multiplicative"))
##
## Smoothing parameters:
##  alpha: 0.186035
##  beta : 0.01309093
##  gamma: 0.377079
##
## Coefficients:
##            [,1]
## a   10.65750852
## b   -0.02674285
## s1   1.34754131
## s2   1.37028771
## s3   1.34644758
## s4   1.12513131
## s5   0.94786152
## s6   0.66694562
## s7   0.61005826
## s8   0.69736547
## s9   0.83121676
## s10  0.93709158
## s11  1.15650917
## s12  1.28131898
```

3.4 - Comparação dos modelos

Apesar de a variabilidade da série sazonal aparentemente permanecer constante ao longo do tempo, usaremos aqui um método mais objetivo para eleger qual dos dois modelos apresenta um melhor ajuste para a série.Para compará-los, vamos usar a Soma de Quadrados dos Erros de Previsão (SQEP). O modelo com o menor SQEP será o melhor modelo.

```
aehw.aditivo$SSE
```

```
## [1] 115.6369
```

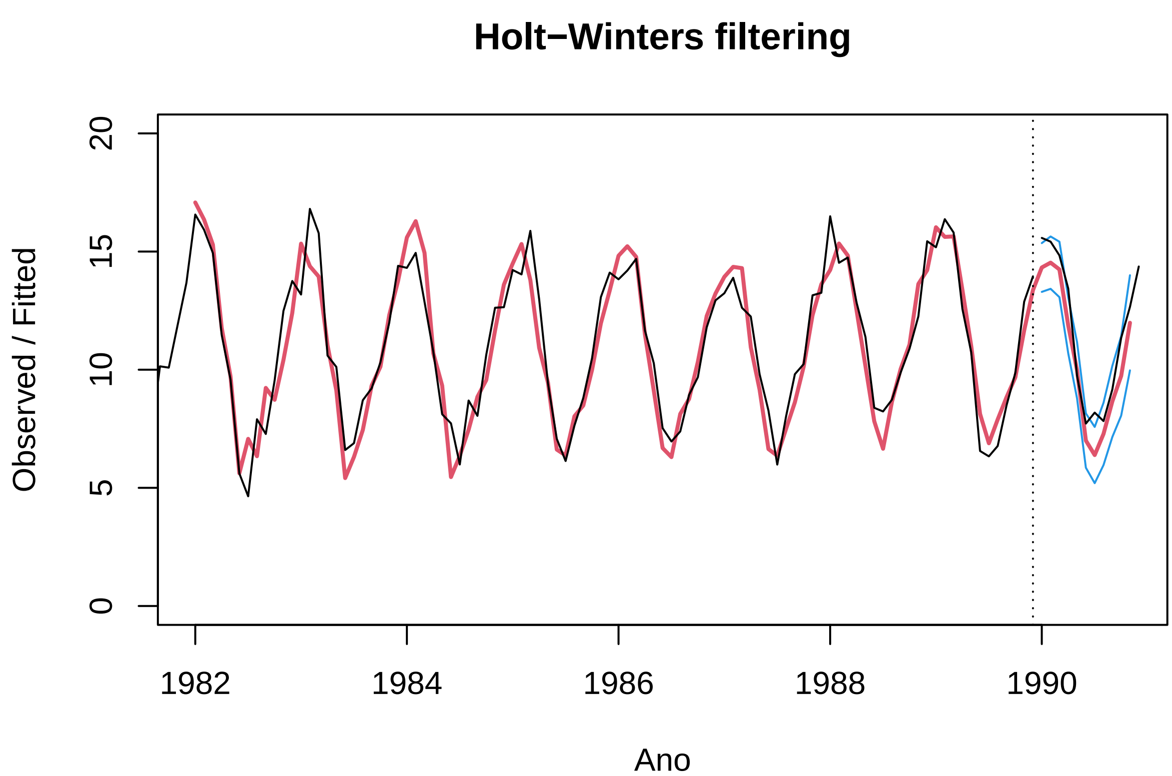
```r
aehw.multiplicativo$SSE
```

```
## [1] 109.882
```

Como o SQEP do modelo multiplicativo é menor, usaremos ele para fazer as previsões.

3.5 - Previsões usando AEHW multiplicativo

```r
previsao <- predict(aehw.multiplicativo, n.ahead = 11, prediction.interval = T,
                    level = 0.95, interval = "prediction")
a <- ts(test$temperatura, frequency = 12, start = 1990)
plot(aehw.multiplicativo, previsao, lwd =2, col = "black", xlab = "Ano", ylim = c(0,20))
lines(a, col = "black", lwd = 1)
```



Em preto no gráfico estão os valores reais observados na série, sendo aqueles que estão antes da linha vertical rachurada os dados de treino e os que estão depois os de teste. Em vermelho, as previsões realizadas pelo AEWH multiplicativo e em azul os limites do intervalo de confiança para as previsões dos dados de teste.

Como a série real de teste está inserida no intervalo de confiança da previsão, podemos considerar que o modelo AEWH multiplicativo prevê razoavelmente bem a série original.

4 - Comparação dos Modelos

4.1 - Previsões um passo à frente

```r
dados <- train$temperatura
previsoes <- c()
```

```r
for (i in 1:(length(test$temperatura))){
  # Adicionando a nova observação
  if (i != 1){
      dados <- c(dados, test[i-1,]$temperatura)
  }

  # Série com nova observação
  serie_temp <- ts(dados, frequency=12, start = 1981)

  # Melhores modelos
  marima <- arima(serie_temp, order = c(2, 0, 0), seasonal = list(order = c(1, 0, 1)))
  aehw.multiplicativo <- HoltWinters(serie_temp, alpha=NULL, beta=NULL,
                                     gamma=NULL, seasonal = c("multiplicative"))
  # Previsão um passo a frente
  previsao_aehw <- predict(marima, n.ahead=1, prediction.interval = TRUE, level = 0.95,        interval=
  previsao_arima <- predict(aehw.multiplicativo, n.ahed=1, prediction.interval = TRUE, level = 0.95, int

  # Previsão AEHW
  aehw_xhat <- unname(previsao_aehw$pred)
  aehw_lwr <- unname(aehw_xhat - qnorm(0.025, lower.tail=FALSE)*previsao_aehw$se)
  aehw_upp <- unname(aehw_xhat + qnorm(0.025, lower.tail=FALSE)*previsao_aehw$se)

  previsoes_temp <- c(aehw_xhat, aehw_lwr, aehw_upp)

  # Previsão ARIMA
  previsoes_temp <- c(previsoes_temp, unname(previsao_arima[1,]), test$temperatura[i])

  previsoes <- rbind(previsoes, previsoes_temp)

}
rownames(previsoes) <- NULL
colnames(previsoes) <- c('AEHW_xhat', 'AEHW_lwr', 'AEHW_upp',
                         'ARIMA_xhat', 'ARIMA_upp', 'ARIMA_lwr', 'Y_t')
previsoes <- as.data.frame(previsoes)

head(previsoes)
```

```
##   AEHW_xhat  AEHW_lwr  AEHW_upp ARIMA_xhat ARIMA_upp ARIMA_lwr       Y_t
## 1 14.965277 12.987733 16.942820  14.325396 15.357513 13.293279 15.577419
## 2 15.479808 13.514004 17.445611  14.821395 15.785022 13.857769 15.417857
## 3 14.667689 12.714983 16.620395  14.644921 15.575491 13.714350 14.835484
## 4 12.097550 10.154006 14.041093  12.216598 13.137000 11.296197 13.433333
## 5 10.565639  8.593369 12.537909  10.433331 11.360376  9.506286  9.748387
## 6  7.600516  5.643970  9.557062   7.268705  8.167636  6.369774  7.720000
```

4.2 - Comparando o Erro Quadrático Médio de Previsão dos dois modelos

```r
AEHW_EQMP <- mean((previsoes$AEHW_xhat - previsoes$Y_t)^2)
ARIMA_EQMP <- mean((previsoes$ARIMA_xhat - previsoes$Y_t)^2)

AEHW_EQMP
```

```
## [1] 0.5572061
```

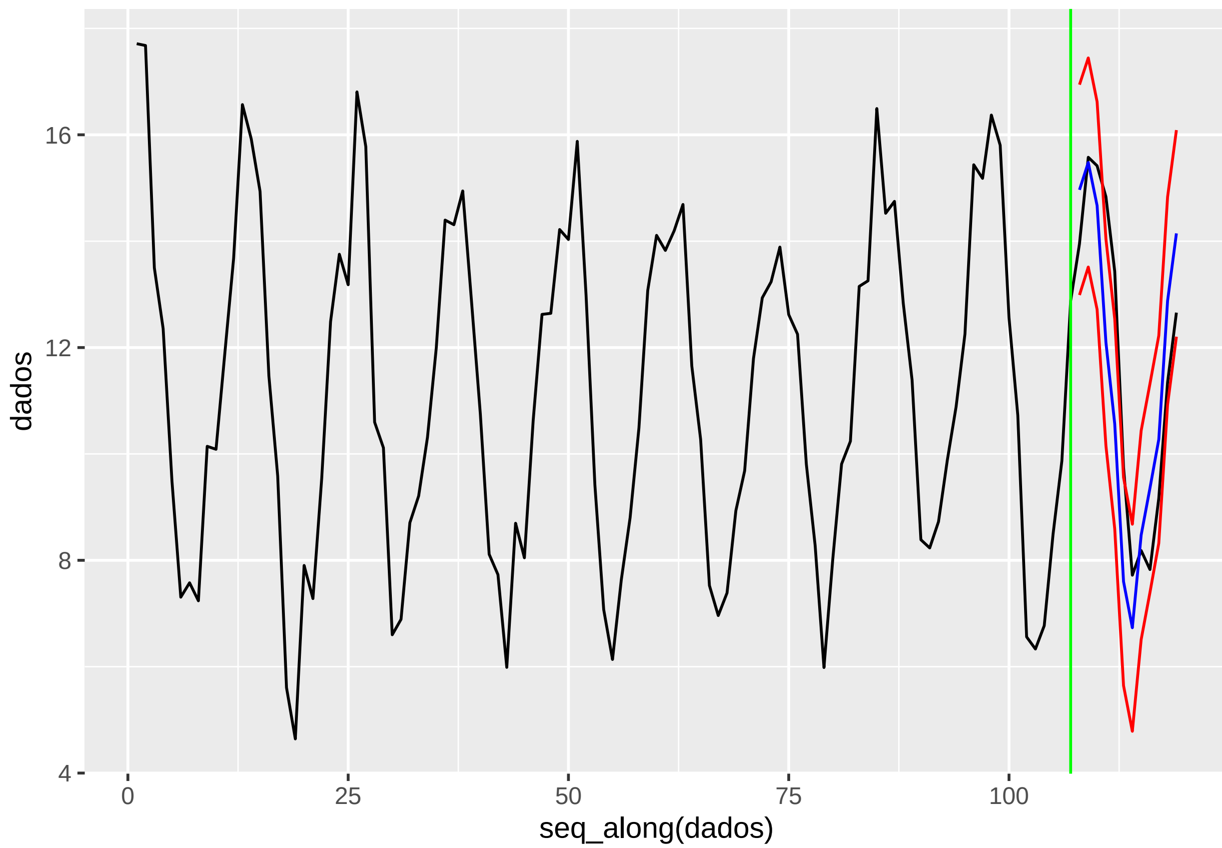
```
ARIMA_EQMP
```

```
## [1] 0.6061632
```

O alisamento de Holt-Winters tem um EQMP menor, logo é o melhor modelo e será o escolhido.

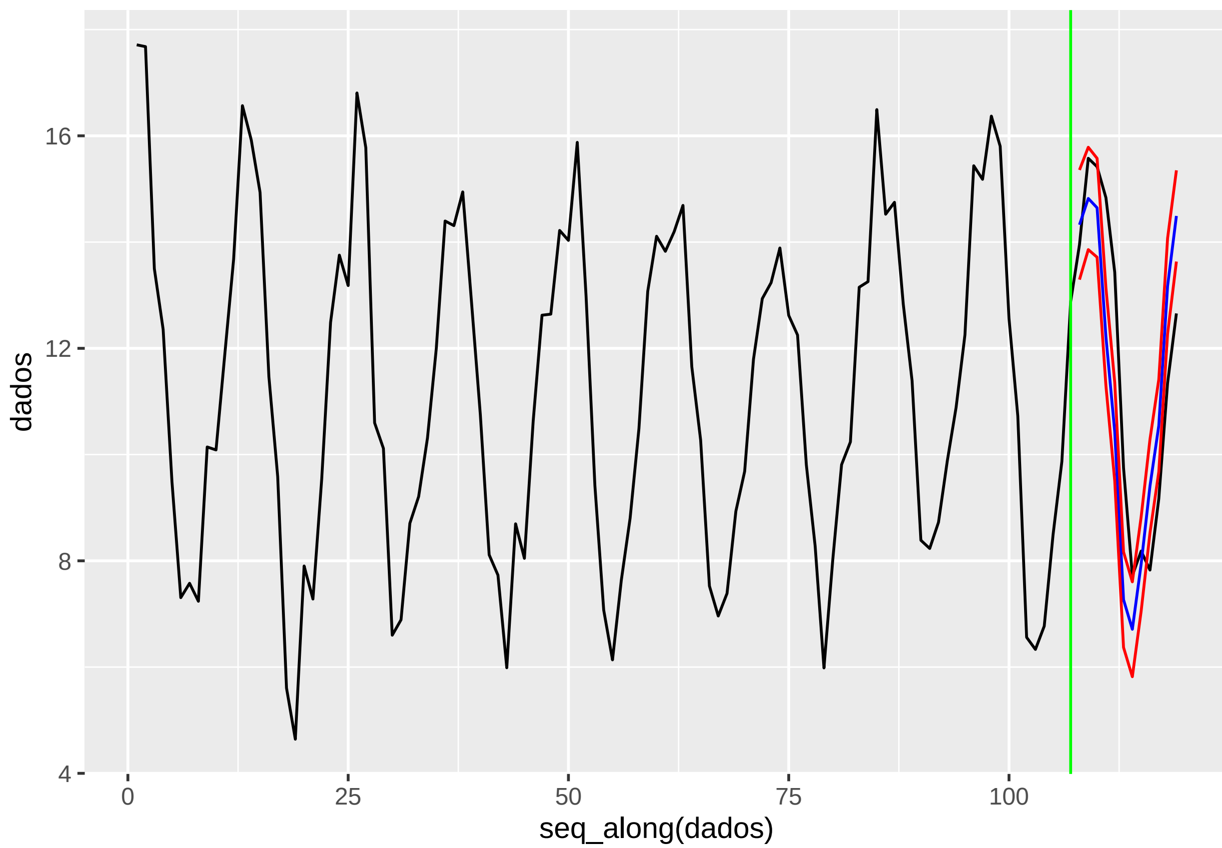
4.3 - Gráfico Previsão - Holt-Winters

```
ggplot()+
  geom_line(aes(x=seq_along(dados), y=dados))+
  geom_vline(aes(xintercept=length(dados)-12), colour='green')+
  geom_line(aes(x=seq_len(12)+length(dados) - 12, y=previsoes$AEHW_xhat), colour='blue')+
  geom_line(aes(x=seq_len(12)+length(dados) - 12, y = previsoes$AEHW_lwr), colour='red')+
  geom_line(aes(x=seq_len(12)+length(dados) - 12, y = previsoes$AEHW_upp), colour='red')
```



4.4 - Gráfico Previsão - ARIMA

```
ggplot()+
  geom_line(aes(x=seq_along(dados), y=dados))+
  geom_vline(aes(xintercept=length(dados)-12), colour='green')+
  geom_line(aes(x=seq_len(12)+length(dados) - 12, y=previsoes$ARIMA_xhat), colour='blue')+
  geom_line(aes(x=seq_len(12)+length(dados) - 12, y = previsoes$ARIMA_lwr), colour='red')+
  geom_line(aes(x=seq_len(12)+length(dados) - 12, y = previsoes$ARIMA_upp), colour='red')
```


16
12
8
4
dados
0
25
50
75
100
seq_along(dados)