Análise de Séries Temporais

0.6 - Aula Prática

Prof. Dr. Eder Angelo Milani

13/06/2025

Tendência - Suavização

Para exemplificar o cálculo da tendência utilizando o método de Média Móvel, vamos utilizar os valores apresentado da Aula Teórica.

```
z <- c(84.6, 89.9, 81.9, 95.4, 91.2, 89.8)

mm <- numeric()
for(t in 2:5){
  mm[t] <- (z[t-1] + z[t] + z[t+1]) / 3
}

round(mm, 1)
```

```
## [1]   NA 85.5 89.1 89.5 92.1
```

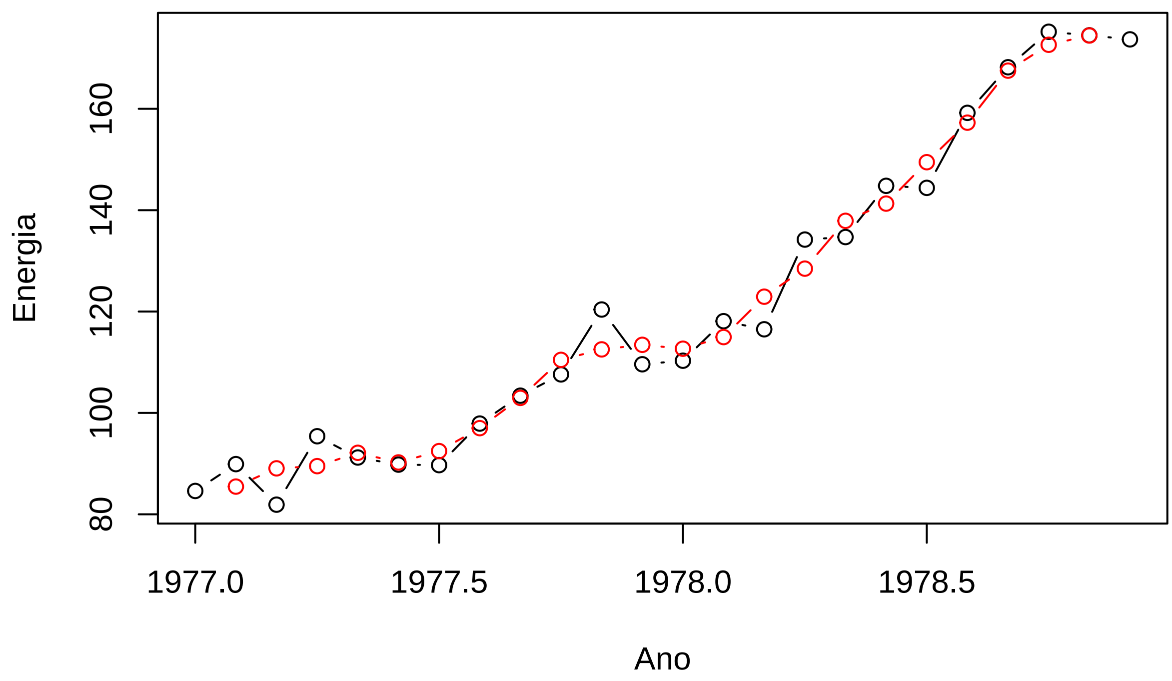
Exemplo de uma série temporal com tendência

A seguir apresentamos os valores mensais do consumo de energia elétrica no Estado do Espírito Santo, referentes aos anos 1977 e 1978, portanto são 24 observações. Série temporal retirada de Morettin e Toloi (2006). Diferentemente da Aula 2, agora vamos aplicar o método de Médias Móveis para estimar a tendência da série temporal.

```
energia <- c(84.6, 89.9, 81.9, 95.4, 91.2, 89.8, 89.7, 97.9, 103.4,
       107.6, 120.4, 109.6, 110.3, 118.1, 116.5, 134.2, 134.7,
       144.8, 144.4, 159.2, 168.2, 175.2, 174.5, 173.7)

energia_ts <- ts(energia, start=c(1977, 1), frequency = 12)

# grafico da serie temporal
plot(energia_ts, type='b', ylab = "Energia", xlab="Ano")

# calculo da media movel
mm_energia <- numeric()
for(i in 2:23){
  mm_energia[i] <- (energia[i-1] + energia[i] + energia[i+1]) / 3
}

round(mm_energia, 1)
```

```
##  [1]    NA  85.5  89.1  89.5  92.1  90.2  92.5  97.0 103.0 110.5 112.5 113.4
## [13] 112.7 115.0 122.9 128.5 137.9 141.3 149.5 157.3 167.5 172.6 174.5
```

```r
mm_energia_ts <- ts(mm_energia, start=c(1977, 1), frequency = 12)


# grafico da serie temporal
plot(energia_ts, type='b', ylab = "Energia", xlab="Ano")
lines(mm_energia_ts, type="b", col="red")
```




```r
# Existe uma função no R que calcula a media móvel
# ela é exemplificada a seguir.
# install.packages("zoo")
library(zoo)
```

```
## Warning: pacote 'zoo' foi compilado no R versão 4.4.3
```

```
##
## Anexando pacote: 'zoo'
```

```
## Os seguintes objetos são mascarados por 'package:base':
##
##     as.Date, as.Date.numeric
```

```r
# comando align determina como os as medias vao ser dispostas no vetor
# fill = NA, NULL ou 0 - preenchimento dos espacos vazios

round(rollmean(energia_ts, 3, fill = NA, align = "center"), 1)
```

```
##        Jan   Feb   Mar   Apr   May   Jun   Jul   Aug   Sep   Oct   Nov   Dec
## 1977    NA  85.5  89.1  89.5  92.1  90.2  92.5  97.0 103.0 110.5 112.5 113.4
```

```
## 1978 112.7 115.0 122.9 128.5 137.9 141.3 149.5 157.3 167.5 172.6 174.5    NA
```

```r
round(rollmean(energia_ts, 3, fill = NA, align = "right"), 1)
```

```
##        Jan   Feb   Mar   Apr   May   Jun   Jul   Aug   Sep   Oct   Nov   Dec
## 1977    NA    NA  85.5  89.1  89.5  92.1  90.2  92.5  97.0 103.0 110.5 112.5
## 1978 113.4 112.7 115.0 122.9 128.5 137.9 141.3 149.5 157.3 167.5 172.6 174.5
```

```r
round(rollmean(energia_ts, 3, fill = NA, align = "left"), 1)
```

```
##        Jan   Feb   Mar   Apr   May   Jun   Jul   Aug   Sep   Oct   Nov   Dec
## 1977  85.5  89.1  89.5  92.1  90.2  92.5  97.0 103.0 110.5 112.5 113.4 112.7
## 1978 115.0 122.9 128.5 137.9 141.3 149.5 157.3 167.5 172.6 174.5    NA    NA
```

```r
round(rollmean(energia_ts, 3, fill = 0, align = "center"), 1) # preenche com 0
```

```
##        Jan   Feb   Mar   Apr   May   Jun   Jul   Aug   Sep   Oct   Nov   Dec
## 1977   0.0  85.5  89.1  89.5  92.1  90.2  92.5  97.0 103.0 110.5 112.5 113.4
## 1978 112.7 115.0 122.9 128.5 137.9 141.3 149.5 157.3 167.5 172.6 174.5   0.0
```

```r
round(rollmean(energia_ts, 3, fill = NULL, align = "center"), 1) # nao preenche os espacos vazios
```

```
##        Jan   Feb   Mar   Apr   May   Jun   Jul   Aug   Sep   Oct   Nov   Dec
## 1977        85.5  89.1  89.5  92.1  90.2  92.5  97.0 103.0 110.5 112.5 113.4
## 1978 112.7 115.0 122.9 128.5 137.9 141.3 149.5 157.3 167.5 172.6 174.5
```

```r
# para quantidade par de observacoes
round(rollmean(energia_ts, 4, fill = NA, align = "center"), 1)
```

```
##        Jan   Feb   Mar   Apr   May   Jun   Jul   Aug   Sep   Oct   Nov   Dec
## 1977    NA  88.0  89.6  89.6  91.5  92.2  95.2  99.7 107.3 110.2 112.0 114.6
## 1978 113.6 119.8 125.9 132.6 139.5 145.8 154.2 161.8 169.3 172.9    NA    NA
```

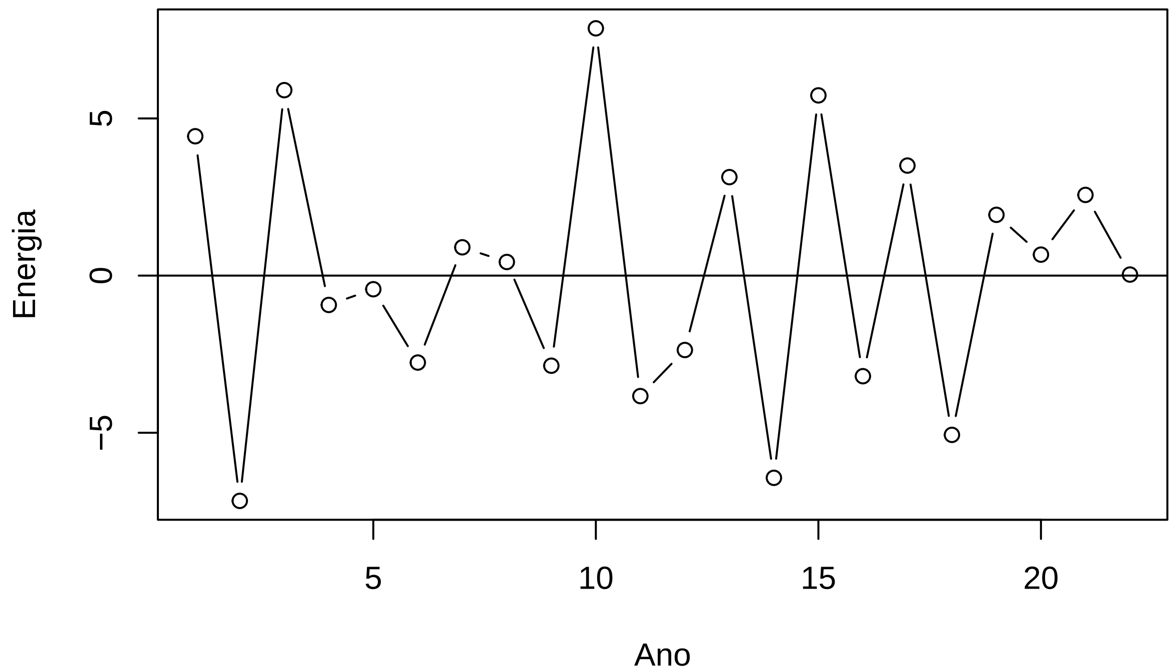
```r
round(rollmean(energia_ts, 4, fill = NA, align = "left"), 1)
```

```
##        Jan   Feb   Mar   Apr   May   Jun   Jul   Aug   Sep   Oct   Nov   Dec
## 1977  88.0  89.6  89.6  91.5  92.2  95.2  99.7 107.3 110.2 112.0 114.6 113.6
## 1978 119.8 125.9 132.6 139.5 145.8 154.2 161.8 169.3 172.9    NA    NA    NA
```

```r
round(rollmean(energia_ts, 4, fill = NA, align = "right"), 1)
```

```
##        Jan   Feb   Mar   Apr   May   Jun   Jul   Aug   Sep   Oct   Nov   Dec
## 1977    NA    NA    NA  88.0  89.6  89.6  91.5  92.2  95.2  99.7 107.3 110.2
## 1978 112.0 114.6 113.6 119.8 125.9 132.6 139.5 145.8 154.2 161.8 169.3 172.9
```

```r
# serie livre de sazonalidade

energia_st_ts <- energia_ts - mm_energia_ts

plot(energia_st_ts[2:23], type="b", ylab = "Energia", xlab="Ano")
abline(h=0)
```



Em vez de tomar médias móveis, podemos calcular medianas móveis, um exemplo é dado a seguir.

```
z <- c(84.6, 89.9, 81.9, 95.4, 91.2, 89.8)

mdm <- numeric()
for(t in 2:5){
  mdm[t] <- median(z[t-1], z[t], z[t+1])
}

round(mdm, 1)
```

```
## [1]   NA 84.6 89.9 81.9 95.4
```

```
# podemos fazer os calculos usando uma funcao pronta

round(rollmedian(z, 3, fill = NA, align = "center"), 1)
```

```
## [1]   NA 84.6 89.9 91.2 91.2   NA
```

Sazonalidade - Suavização

Exemplo de uma série temporal com sazonalidade

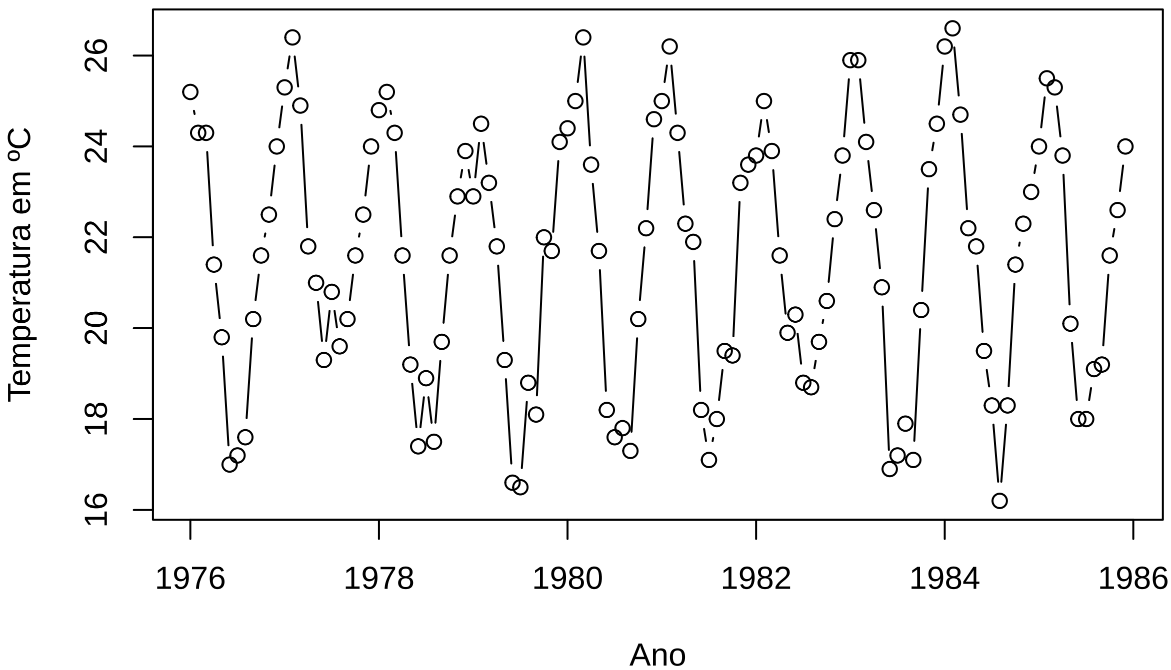
A seguir apresentamos as temperaturas médias mensais, em graus centígrados, da cidade de Cananéia (município brasileiro do litoral de São Paulo), de janeiro de 1976 a dezembro de 1985. Série temporal retirada de Morettin e Toloi (2006).

Obs.: como já discutido anteriormente, está série não apresenta tendência.

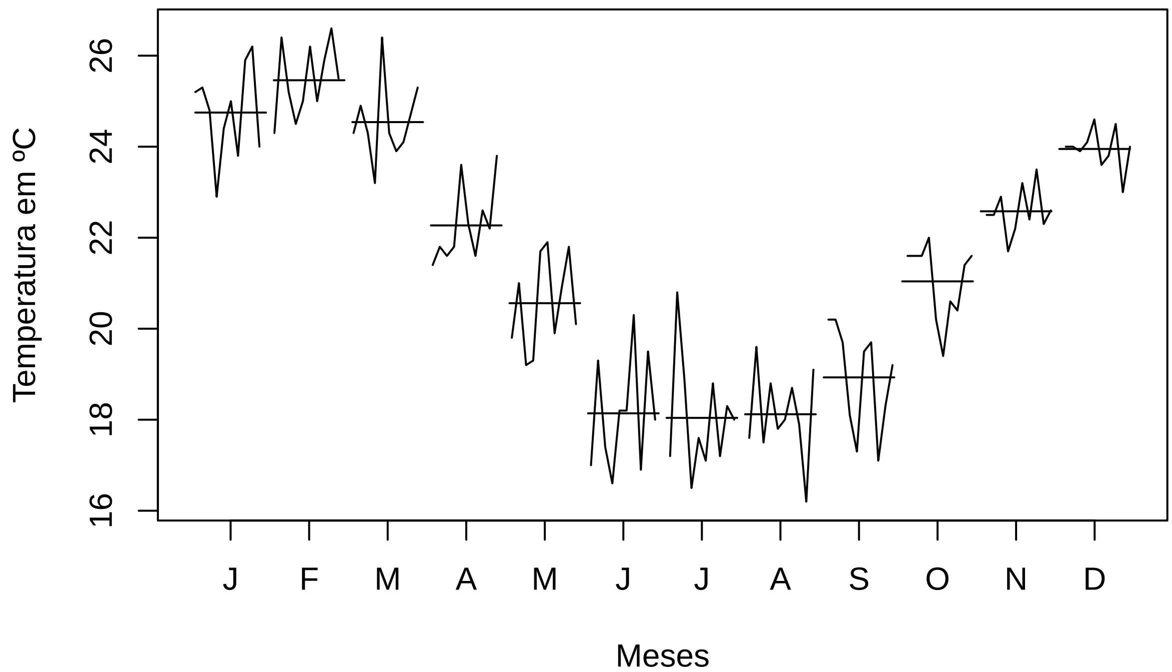
```
# fazendo a leitura do conjunto de dados
setwd("G:\\Meu Drive\\UFG\\Especializacao\\Aulas de series temporais\\Codigos")

library(readxl)
temperatura <- read_excel("temperatura.xls")
head(temperatura)
```

```
## # A tibble: 6 x 3
##     Ano Cananeia Ubatuba
##   <dbl>    <dbl>   <dbl>
## 1  1976     25.2    27.1
## 2    NA     24.3    25.3
## 3    NA     24.3    25.8
## 4    NA     21.4    23.7
## 5    NA     19.8    21.6
## 6    NA     17      20
```

```
temp.cananeia_ts <- ts(temperatura$Cananeia, start = c(1976, 1), frequency = 12)

# grafico da serie temporal
plot.ts(temp.cananeia_ts, type="b", ylab="Temperatura em ºC", xlab="Ano")
```




```
# grafico dos meses separadamente
monthplot(temp.cananeia_ts,  xlab="Meses", ylab="Temperatura em ºC", main="")
```




```
# dispor os dados em uma matriz
# colunas = meses
# linhas = anos
cananeia_matriz <- matrix(temp.cananeia_ts, ncol=12, nrow=10, byrow=T)
cananeia_matriz
```

```
##       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12]
##  [1,] 25.2 24.3 24.3 21.4 19.8 17.0 17.2 17.6 20.2  21.6  22.5  24.0
##  [2,] 25.3 26.4 24.9 21.8 21.0 19.3 20.8 19.6 20.2  21.6  22.5  24.0
##  [3,] 24.8 25.2 24.3 21.6 19.2 17.4 18.9 17.5 19.7  21.6  22.9  23.9
##  [4,] 22.9 24.5 23.2 21.8 19.3 16.6 16.5 18.8 18.1  22.0  21.7  24.1
##  [5,] 24.4 25.0 26.4 23.6 21.7 18.2 17.6 17.8 17.3  20.2  22.2  24.6
##  [6,] 25.0 26.2 24.3 22.3 21.9 18.2 17.1 18.0 19.5  19.4  23.2  23.6
##  [7,] 23.8 25.0 23.9 21.6 19.9 20.3 18.8 18.7 19.7  20.6  22.4  23.8
##  [8,] 25.9 25.9 24.1 22.6 20.9 16.9 17.2 17.9 17.1  20.4  23.5  24.5
##  [9,] 26.2 26.6 24.7 22.2 21.8 19.5 18.3 16.2 18.3  21.4  22.3  23.0
## [10,] 24.0 25.5 25.3 23.8 20.1 18.0 18.0 19.1 19.2  21.6  22.6  24.0
```

```
# vetor de medias
vetor_media <- colMeans(cananeia_matriz, na.rm = T)
vetor_media
```

```
##  [1] 24.75 25.46 24.54 22.27 20.56 18.14 18.04 18.12 18.93 21.04 22.58 23.95
```

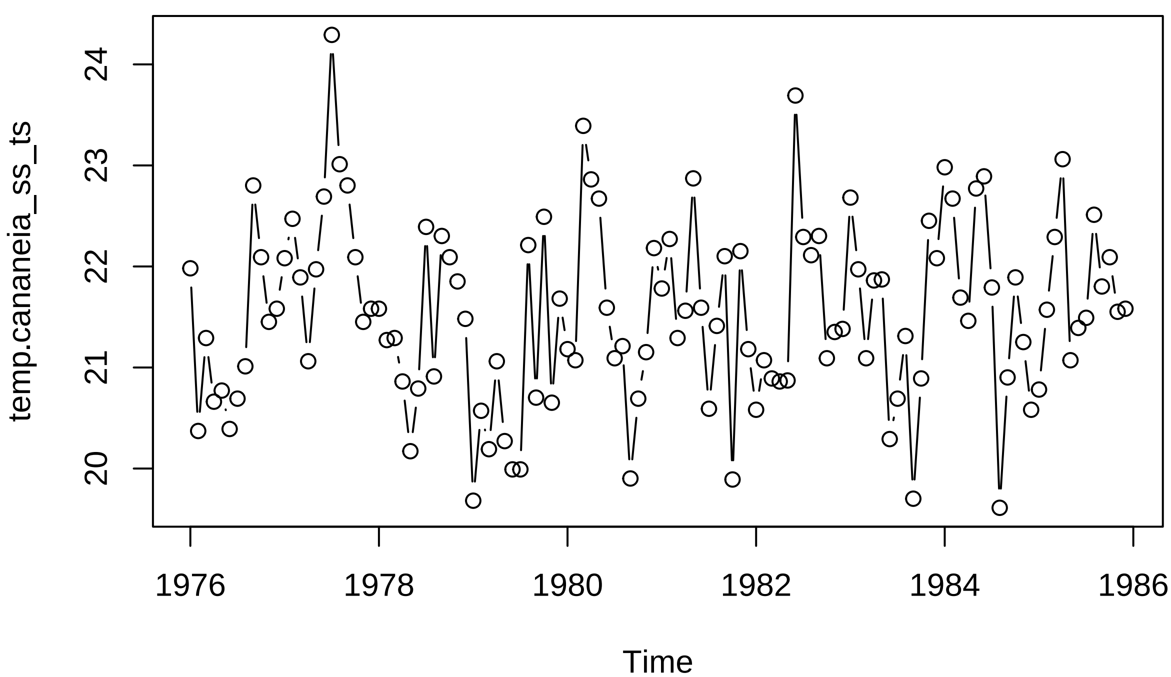
```
# media
temp_media <- mean(vetor_media)
temp_media
```

```
## [1] 21.53167
```

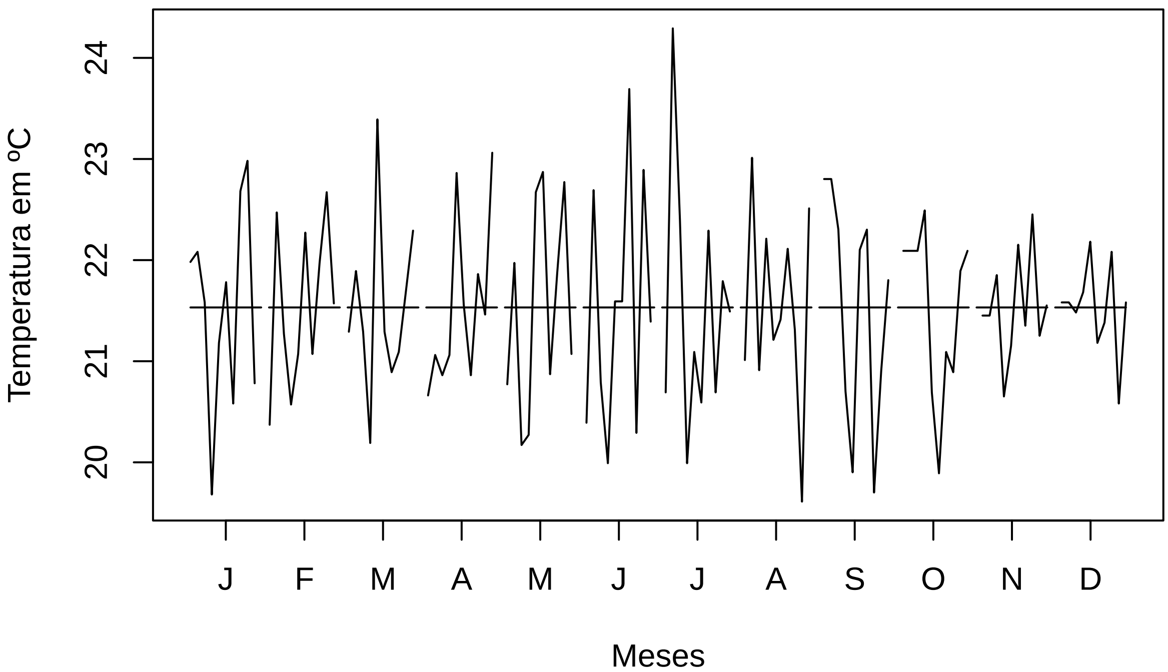
```r
# sazonalidade estimada
sazonalidade <- vetor_media - temp_media
sazonalidade
```

```
##  [1]  3.2183333  3.9283333  3.0083333  0.7383333 -0.9716667 -3.3916667
##  [7] -3.4916667 -3.4116667 -2.6016667 -0.4916667  1.0483333  2.4183333
```

```r
# vetor com a sazonaldiade
sazonalidade <- rep(sazonalidade, 10)

# serie livre de sazonalidade
temp.cananeia_ss <- temp.cananeia_ts - sazonalidade

temp.cananeia_ss_ts <- ts(temp.cananeia_ss, start = c(1976, 1), frequency = 12)

# grafico da serie livre de sazonalidade
plot(temp.cananeia_ss_ts, type="b")
```




```r
# grafico dos meses separadamente
monthplot(temp.cananeia_ss_ts,  xlab="Meses", ylab="Temperatura em ºC", main="")
```


Temperatura em ºC
20
21
22
23
24
J
F
M
A
M
J
J
A
S
O
N
D
Meses

Exemplo de uma série temporal com tendência e sazonalidade

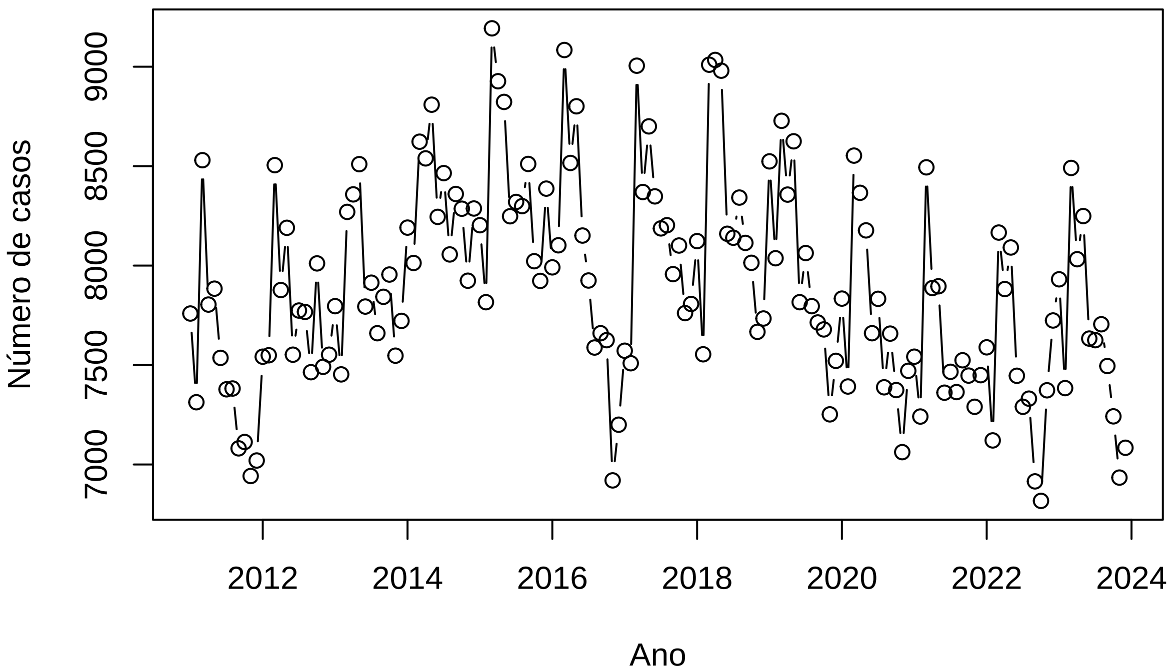
A série temporal em estudo é do número mensal de nascidos vivos em Goiás no período de janeiro de 2011 até dezembro de 2023. Os valors foram obtidos no site do DATASUS, para acessar click aqui. Os dados foram obtidos em 20/09/2024.

```r
library(forecast)
```

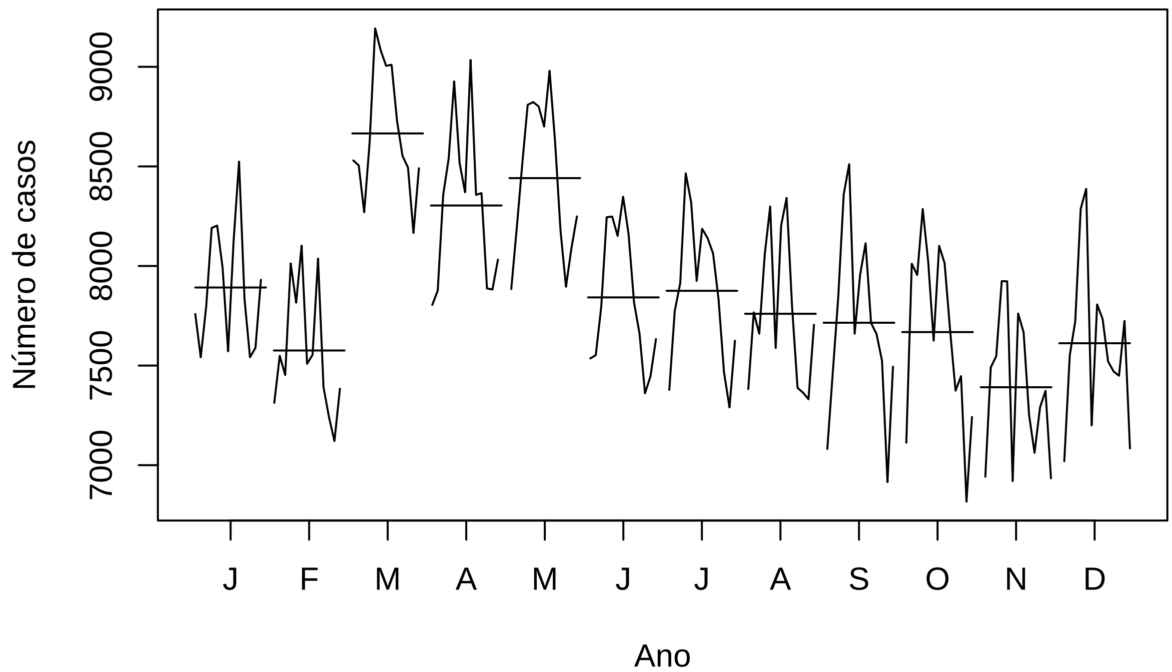
```
## Warning: pacote 'forecast' foi compilado no R versão 4.4.3
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```r
#serie temporal do numero mensal de nascidos vivos em Goias

serie_nascido <- c(7759, 7313, 8530, 7804, 7884, 7536, 7378, 7382, 7081, 7113, 6942, 7020,
                  7542, 7549, 8505, 7877, 8190, 7552, 7774, 7767, 7464, 8011, 7491, 7552,
                  7796, 7453, 8270, 8358, 8510, 7794, 7914, 7660, 7843, 7955, 7547, 7722,
                  8191, 8013, 8623, 8539, 8809, 8245, 8465, 8056, 8360, 8286, 7924, 8287,
                  8203, 7816, 9193, 8927, 8823, 8248, 8320, 8299, 8511, 8022, 7923, 8387,
                  7991, 8102, 9084, 8516, 8801, 8151, 7925, 7588, 7660, 7625, 6920, 7200,
                  7572, 7509, 9005, 8370, 8700, 8348, 8187, 8203, 7957, 8101, 7761, 7807,
                  8123, 7554, 9010, 9034, 8980, 8160, 8140, 8342, 8114, 8014, 7667, 7734,
                  8524, 8037, 8728, 8357, 8625, 7816, 8063, 7796, 7714, 7679, 7252, 7521,
                  7834, 7392, 8553, 8366, 8177, 7660, 7833, 7388, 7658, 7374, 7062, 7471,
                  7542, 7241, 8494, 7887, 7896, 7361, 7466, 7364, 7524, 7447, 7290, 7449,
                  7589, 7121, 8166, 7882, 8091, 7446, 7290, 7331, 6915, 6817, 7373, 7724,
                  7931, 7384, 8491, 8032, 8249, 7633, 7625, 7705, 7495, 7242, 6934, 7084)


nascido_ts <- ts(serie_nascido, start= c(2011, 1), frequency = 12)

plot(nascido_ts, type="b", ylab="Número de casos", xlab="Ano")
```



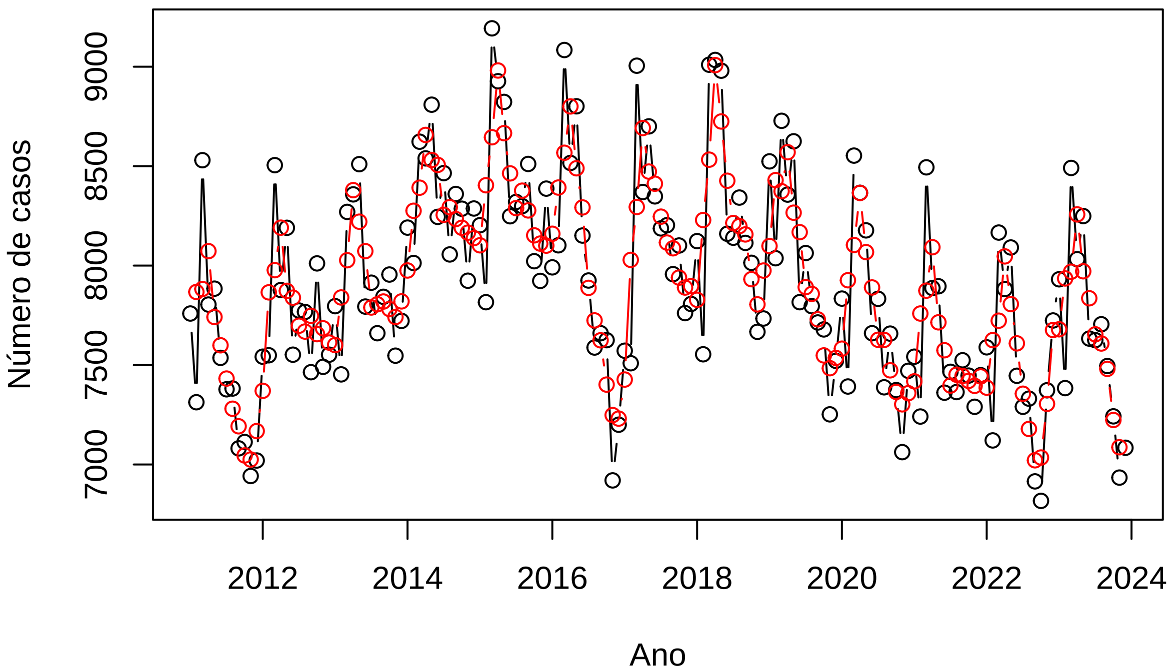

```r
# grafico dos meses separadamente
monthplot(nascido_ts, ylab="Número de casos", xlab="Ano")
```



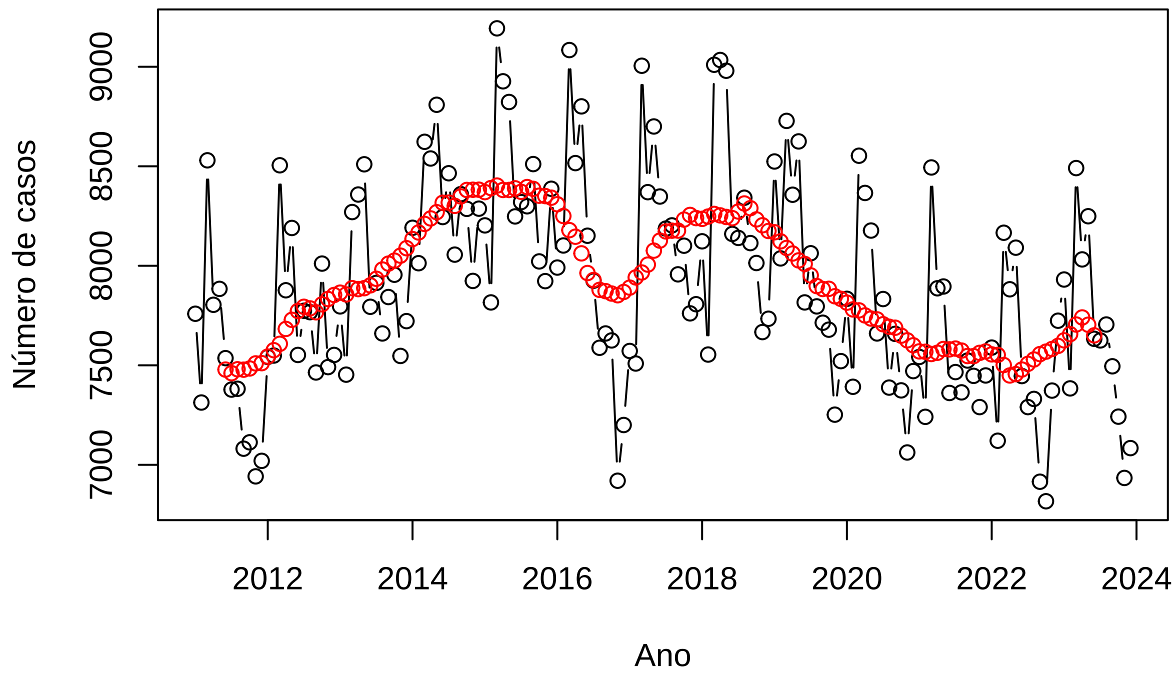

```r
# estimando a tendencia
# calculo da media movel

# install.packages("zoo")
library(zoo)

# comando align determina como as medias vao ser dispostas no vetor
# fill = NA, NULL ou 0 - preenchimento dos espacos vazios

# media de 3 meses
mm3_nascido_ts <- rollmean(nascido_ts, 3, fill = NA, align = "center")
is.ts(mm3_nascido_ts)
```

```
## [1] TRUE
```

```r
# grafico da serie temporal
plot(nascido_ts, type="b", ylab="Número de casos", xlab="Ano")
lines(mm3_nascido_ts, type="b", col="red")
```




```r
# media de 12 meses
mm12_nascido_ts <- rollmean(nascido_ts, 12, fill = NA, align = "center")
#View(mm12_nascido_ts)

# grafico da serie temporal
plot(nascido_ts, type="b", ylab="Número de casos", xlab="Ano")
lines(mm12_nascido_ts, type="b", col="red")
```




```r
# observe que para capturar um comportamento de mais longo prazo
# a quantidade de valores para a media nao pode ser pequena

# serie livre de tendencia
nascido_st <- nascido_ts - mm12_nascido_ts

# estimando a sazonalidade

# dispor os dados em uma matriz
# colunas = meses
# linhas = anos
nascido_matriz <- matrix(nascido_st, ncol=12, nrow=13, byrow=T)
nascido_matriz
```

```
##              [,1]       [,2]      [,3]     [,4]     [,5]       [,6]        [,7]
##  [1,]          NA         NA        NA       NA       NA   57.50000  -82.416667
##  [2,]   -1.916667  -27.00000  897.0833 194.2500 461.5000 -220.83333  -20.000000
##  [3,]  -69.000000 -403.08333  382.3333 475.0000 622.3333 -107.83333  -20.750000
##  [4,]   56.666667 -154.33333  412.5833 301.0000 539.5833  -71.50000  147.500000
##  [5,] -167.250000 -574.50000  789.9167 545.9167 442.0000 -141.33333  -51.666667
##  [6,] -318.333333 -148.08333  904.8333 369.9167 738.5000  187.41667   -3.666667
##  [7,] -318.333333 -432.58333 1038.6667 364.0000 623.9167  221.33333   14.416667
##  [8,] -112.833333 -693.41667  749.5000 780.7500 734.5833  -79.33333 -132.750000
##  [9,]  355.583333  -85.91667  638.4167 295.3333 597.9167 -193.33333  111.166667
## [10,]   19.250000 -388.75000  776.9167 615.3333 442.1667  -70.66667  126.666667
## [11,]  -28.000000 -327.00000  937.1667 324.0833 314.0833 -219.08333 -118.000000
```

```
## [12,]   34.083333 -431.16667  664.5833 433.0833 635.1667  -32.75000 -217.250000
## [13,]  305.583333 -272.58333  786.0833 291.6667 545.2500  -17.41667          NA
##             [,8]       [,9]      [,10]     [,11]      [,12]
##  [1,]  -98.08333 -397.00000 -371.08333 -567.5833 -490.91667
##  [2,]  -19.00000 -302.41667  204.50000 -342.1667 -301.33333
##  [3,] -321.41667 -167.83333  -70.91667 -503.8333 -366.41667
##  [4,] -245.08333   11.41667  -94.91667 -458.0833  -95.33333
##  [5,]  -96.50000  124.58333 -330.16667 -427.3333   44.75000
##  [6,] -291.25000 -212.66667 -235.50000 -932.0833 -668.50000
##  [7,]   26.66667 -219.75000 -131.08333 -494.4167 -432.75000
##  [8,]   29.00000 -175.50000 -219.08333 -536.5000 -440.83333
##  [9,] -102.08333 -169.50000 -205.25000 -594.9167 -312.91667
## [10,] -305.75000  -30.83333 -274.91667 -563.5000 -129.58333
## [11,] -210.00000  -22.66667  -99.25000 -272.5000 -120.58333
## [12,] -198.16667 -641.25000 -751.75000 -208.9167  126.50000
## [13,]         NA         NA         NA        NA         NA
```

```
# vetor de medias
vetor_media <- colMeans(nascido_matriz, na.rm = T)
vetor_media
```

```
##  [1]  -20.37500 -328.20139  748.17361  415.86111  558.08333  -52.91026
##  [7]  -20.56250 -152.63889 -183.61806 -214.95139 -491.81944 -265.65972
```

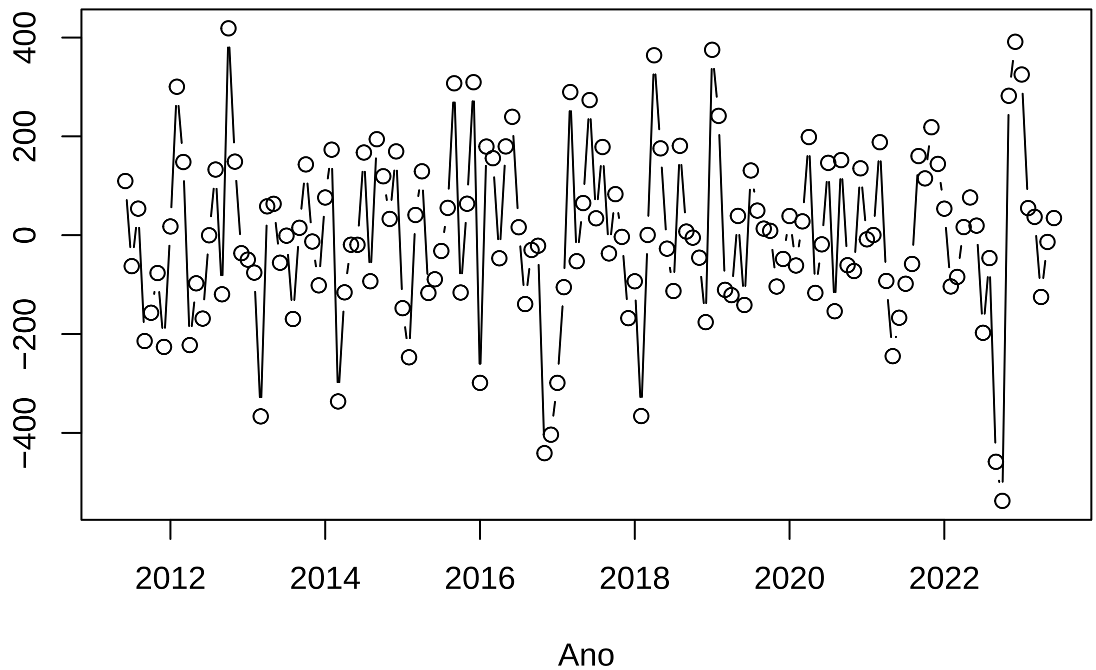
```
# media
nasc_media <- mean(vetor_media)
nasc_media
```

```
## [1] -0.7182158
```

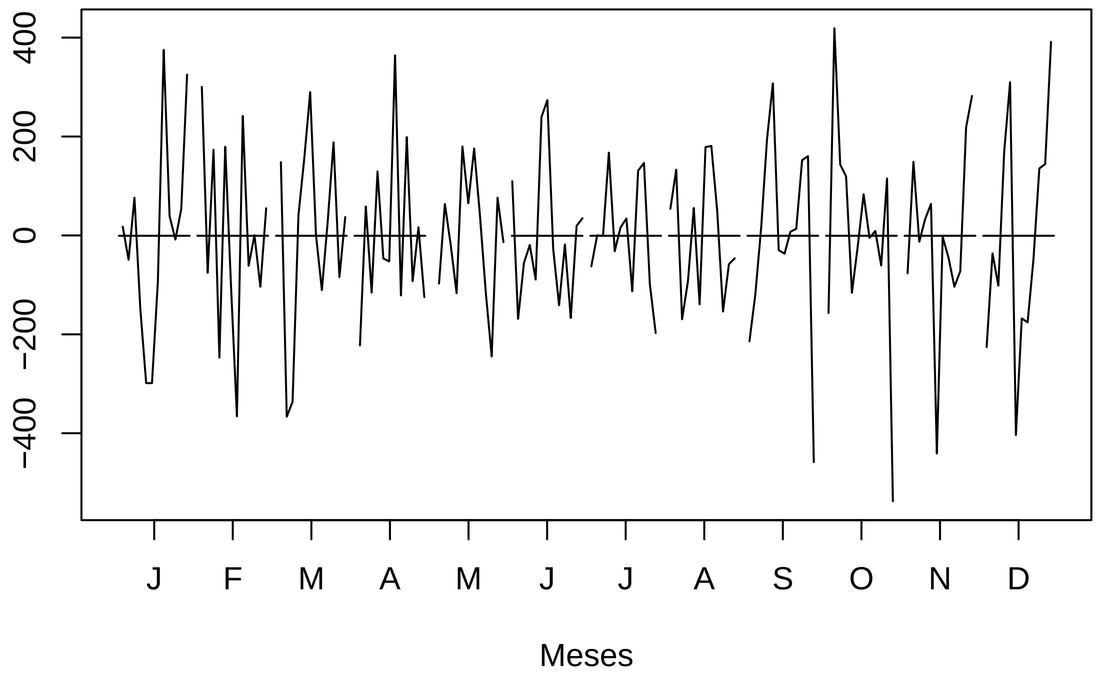
```
# sazonalidade estimada
sazonalidade <- vetor_media - nasc_media
sazonalidade
```

```
##  [1]  -19.65678 -327.48317  748.89183  416.57933  558.80155  -52.19204
##  [7]  -19.84428 -151.92067 -182.89984 -214.23317 -491.10123 -264.94151
```

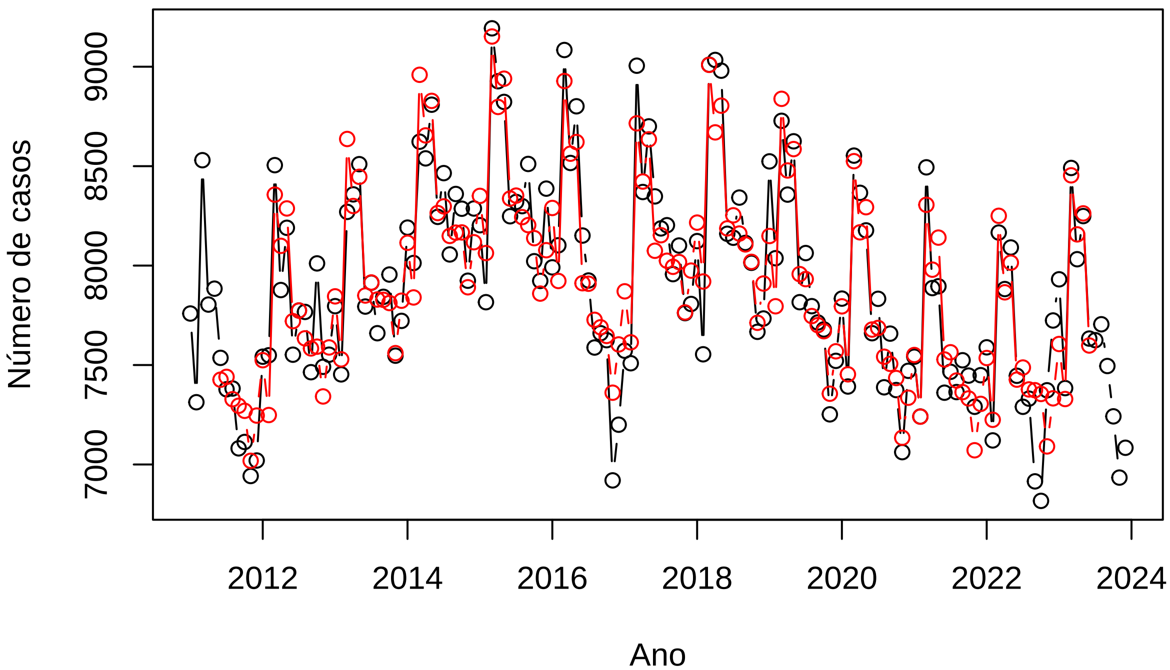
```
# vetor com a sazonaldiade
sazonalidade <- rep(sazonalidade, 13)

# serie livre de sazonalidade
# note os valores perdidos na estimacao da tendencia
nascido_st_ss <- nascido_st[5:150] - sazonalidade[5:150]

# note onde vai iniciar a nova serie
nascido_st_ss <- ts(nascido_st_ss, start = c(2011, 5), frequency = 12)

# grafico da serie livre de tendencia e sazonalidade
plot(nascido_st_ss, type="b", xlab="Ano", ylab="")
```




```
# grafico dos meses separadamente
monthplot(nascido_st_ss,  xlab="Meses", ylab="", main="")
```




```r
# grafico da serie temporal + estimacao da tendencia e sazonalidade

# calculo da serie temporal da tendencia + sazonalidade
nascido_tend_saz <- ts((mm12_nascido_ts[5:150] + sazonalidade[5:150]), start = c(2011, 5), frequency = 1

plot(nascido_ts, type="b", ylab="Número de casos", xlab="Ano")
lines(nascido_tend_saz, type="b", col="red")
```



Número de casos
7000
7500
8000
8500
9000
2012
2014
2016
2018
2020
2022
2024
Ano

Suavização

Suavização Exponencial Simples (SES)

Exemplo 1

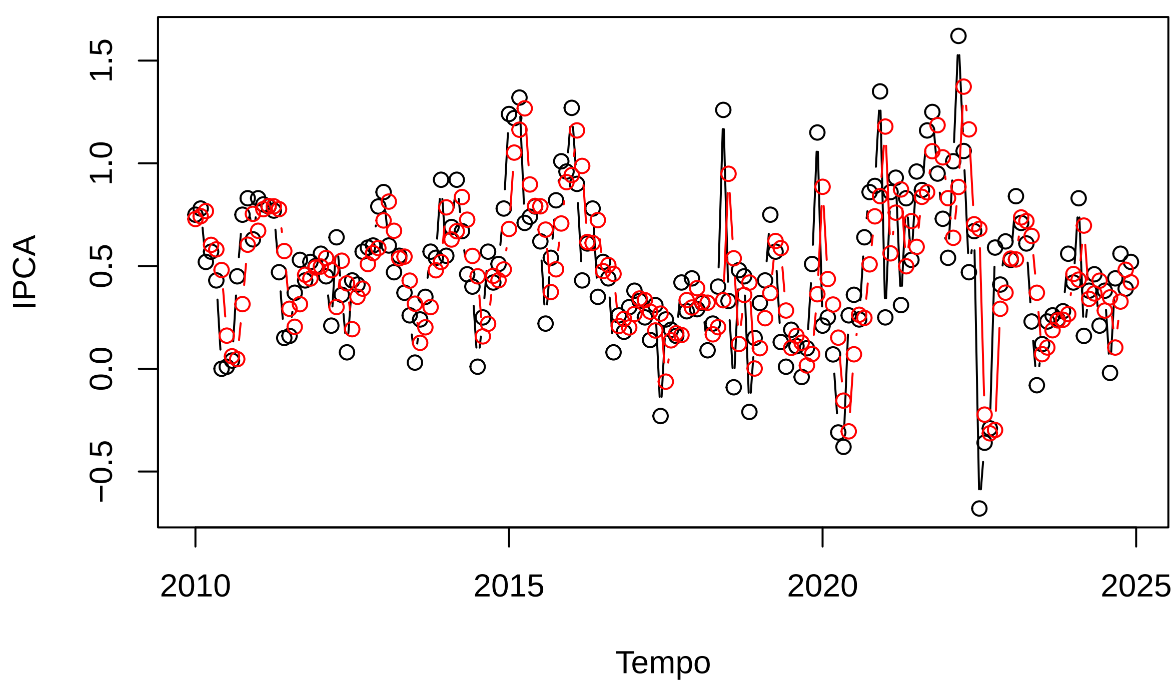
A série temporal a seguir é do IPCA (índice nacional de preços ao consumidor amplo) do período de janeiro de 2010 até dezembro de 2024. Os valores foram obtidos no site do IBGE, para acessar click aqui. Os dados foram acessados em 01/05/2024. Vamos aplicar o método SES para obter previsões.

```r
# serie temporal do IPCA

ipca <- c(0.75, 0.78, 0.52, 0.57, 0.43, 0.00, 0.01, 0.04, 0.45, 0.75, 0.83, 0.63,
          0.83, 0.80, 0.79, 0.77, 0.47, 0.15, 0.16, 0.37, 0.53, 0.43, 0.52, 0.50,
          0.56, 0.45, 0.21, 0.64, 0.36, 0.08, 0.43, 0.41, 0.57, 0.59, 0.60, 0.79,
          0.86, 0.60, 0.47, 0.55, 0.37, 0.26, 0.03, 0.24, 0.35, 0.57, 0.54, 0.92,
          0.55, 0.69, 0.92, 0.67, 0.46, 0.40, 0.01, 0.25, 0.57, 0.42, 0.51, 0.78,
          1.24, 1.22, 1.32, 0.71, 0.74, 0.79, 0.62, 0.22, 0.54, 0.82, 1.01, 0.96,
          1.27, 0.90, 0.43, 0.61, 0.78, 0.35, 0.52, 0.44, 0.08, 0.26, 0.18, 0.30,
          0.38, 0.33, 0.25, 0.14, 0.31,-0.23, 0.24, 0.19, 0.16, 0.42, 0.28, 0.44,
          0.29, 0.32, 0.09, 0.22, 0.40, 1.26, 0.33,-0.09, 0.48, 0.45,-0.21, 0.15,
          0.32, 0.43, 0.75, 0.57, 0.13, 0.01, 0.19, 0.11,-0.04, 0.10, 0.51, 1.15,
          0.21, 0.25, 0.07,-0.31,-0.38, 0.26, 0.36, 0.24, 0.64, 0.86, 0.89, 1.35,
          0.25, 0.86, 0.93, 0.31, 0.83, 0.53, 0.96, 0.87, 1.16, 1.25, 0.95, 0.73,
          0.54, 1.01, 1.62, 1.06, 0.47, 0.67,-0.68,-0.36,-0.29, 0.59, 0.41, 0.62,
          0.53, 0.84, 0.71, 0.61, 0.23,-0.08, 0.12, 0.23, 0.26, 0.24, 0.28, 0.56,
          0.42, 0.83, 0.16, 0.38, 0.46, 0.21, 0.38,-0.02, 0.44, 0.56, 0.39, 0.52)

ipca_ts <- ts(ipca, start= c(2010, 1), frequency = 12)

plot(ipca_ts, type="b", ylab="IPCA", xlab="Tempo")

# ja discutimos que essa serie nao apresenta
# tendencia nem sazonalidade

# Suavizacao Exponencial Simples
# Instale o pacote forecast, se necessário
# install.packages("forecast")

# Carregue o pacote forecast
library(forecast)

# Ajuste do modelo de suavização exponencial simples

modelo_ses <- ses(ipca_ts, h = 5)  # h = número de períodos para previsão
summary(modelo_ses)
```

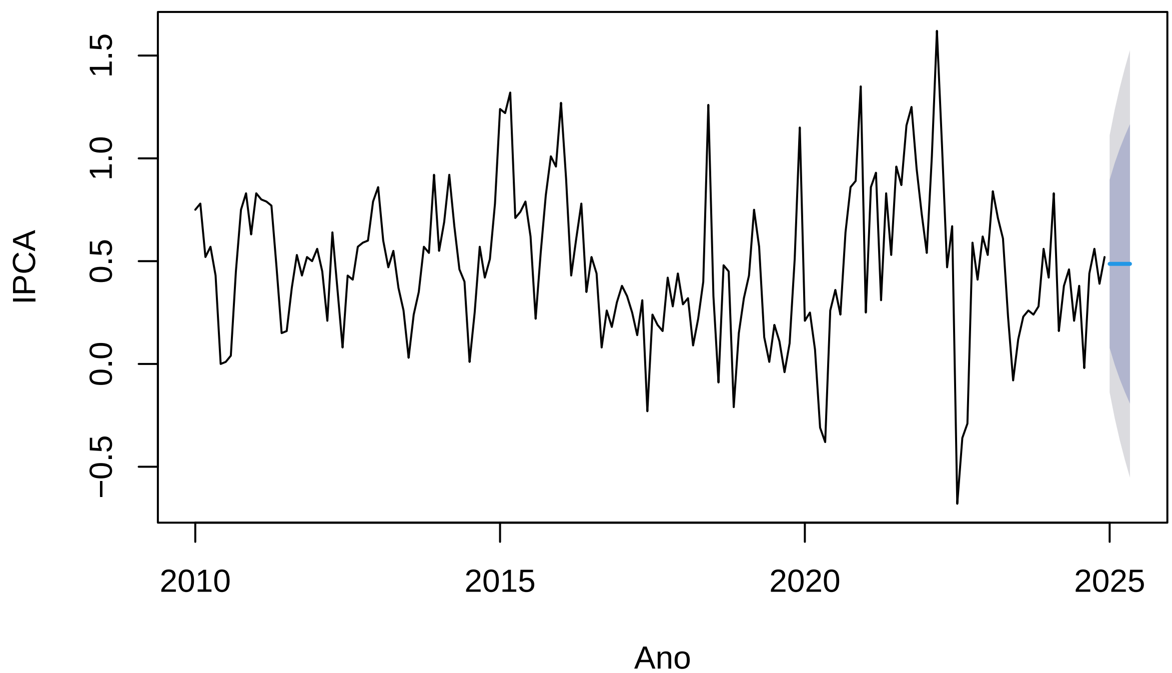
```
##
## Forecast method: Simple exponential smoothing
##
## Model Information:
## Simple exponential smoothing
##
## Call:
## ses(y = ipca_ts, h = 5)
##
```

```
##   Smoothing parameters:
##     alpha = 0.6643
##
##   Initial states:
##     l = 0.7289
##
##   sigma:  0.319
##
##      AIC     AICc      BIC
## 527.3808 527.5172 536.9597
##
## Error measures:
##                        ME      RMSE       MAE  MPE MAPE      MASE       ACF1
## Training set -0.002025229 0.3172077 0.2369852 -Inf  Inf 0.6971373 0.04206412
##
## Forecasts:
##          Point Forecast        Lo 80     Hi 80      Lo 95    Hi 95
## Jan 2025      0.4867065  0.077910961 0.8955020 -0.1384923 1.111905
## Feb 2025      0.4867065 -0.004077567 0.9774905 -0.2638829 1.237296
## Mar 2025      0.4867065 -0.074207225 1.0476201 -0.3711370 1.344550
## Apr 2025      0.4867065 -0.136494440 1.1099074 -0.4663971 1.439810
## May 2025      0.4867065 -0.193098338 1.1665113 -0.5529653 1.526378
```

```r
# visualizacao do ajuste
plot(ipca_ts, type="b", ylab="IPCA", xlab="Tempo")
lines(modelo_ses$fitted, type="b", col="red")
```



```r
plot(modelo_ses, main="", xlab="Ano", ylab="IPCA")
```




```r
# Visualizacao da previsao
plot(modelo_ses, main="", xlab="Ano", ylab="IPCA")

# analisando a opção "initial"

modelo_ses2 <- ses(ipca_ts, h = 5, initial = "optimal")
modelo_ses3 <- ses(ipca_ts, h = 5, initial = "simple")

# Veja as previsões
head(cbind(modelo_ses2$fitted, modelo_ses3$fitted,ipca_ts))
```

```
##          modelo_ses2$fitted modelo_ses3$fitted ipca_ts
## Jan 2010          0.7288856          0.7500000    0.75
## Feb 2010          0.7429127          0.7500000    0.78
## Mar 2010          0.7675513          0.7699412    0.52
## Apr 2010          0.6030931          0.6038039    0.57
## May 2010          0.5811080          0.5813343    0.43
## Jun 2010          0.4807210          0.4807415    0.00
```

```r
tail(cbind(modelo_ses2$fitted, modelo_ses2$fitted,ipca_ts))
```

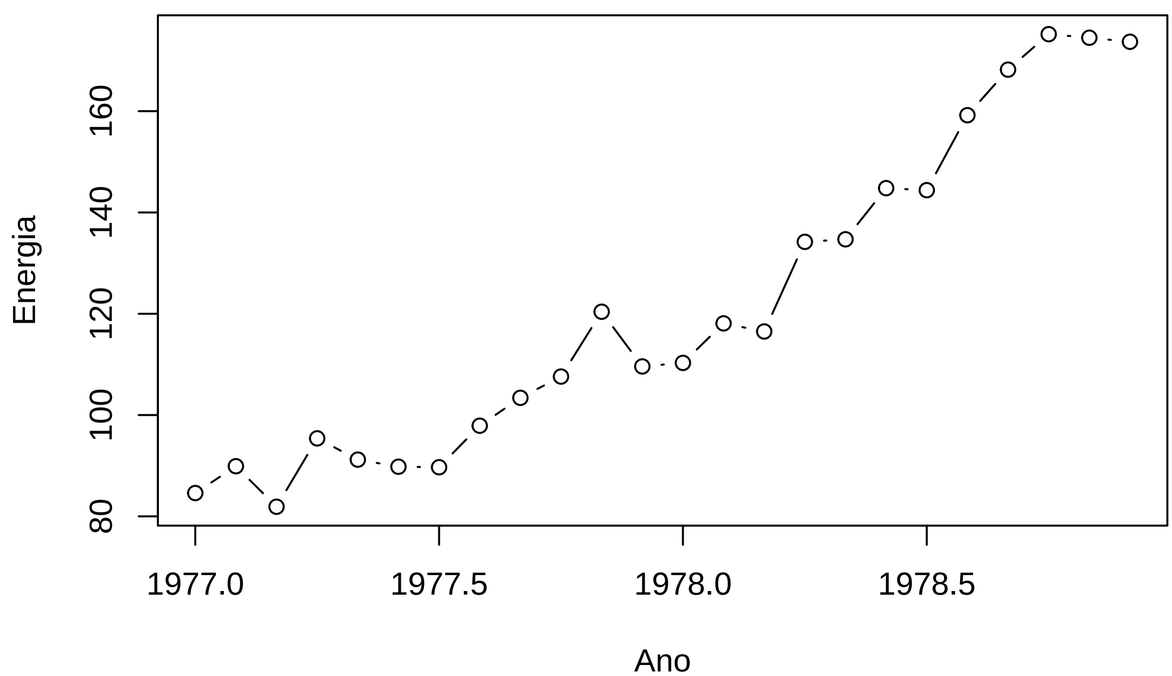
```
##          modelo_ses2$fitted modelo_ses2$fitted ipca_ts
## Jul 2024          0.2833988          0.2833988    0.38
## Aug 2024          0.3475748          0.3475748   -0.02
## Sep 2024          0.1033803          0.1033803    0.44
```

```
## Oct 2024          0.3270101          0.3270101    0.56
## Nov 2024          0.4817946          0.4817946    0.39
## Dec 2024          0.4208118          0.4208118    0.52
```

Exemplo 2

A seguir apresentamos os valores mensais do consumo de energia elétrica no Estado do Espírito Santo, referentes aos anos 1977 e 1978, portanto são 24 observações. Série temporal retirada de Morettin e Toloi (2006).

```
energia <- c(84.6, 89.9, 81.9, 95.4, 91.2, 89.8, 89.7, 97.9, 103.4,
       107.6, 120.4, 109.6, 110.3, 118.1, 116.5, 134.2, 134.7,
       144.8, 144.4, 159.2, 168.2, 175.2, 174.5, 173.7)

energia_ts <- ts(energia, start=c(1977, 1), frequency = 12)

# grafico da serie temporal
plot(energia_ts, type='b', ylab = "Energia", xlab="Ano")
```

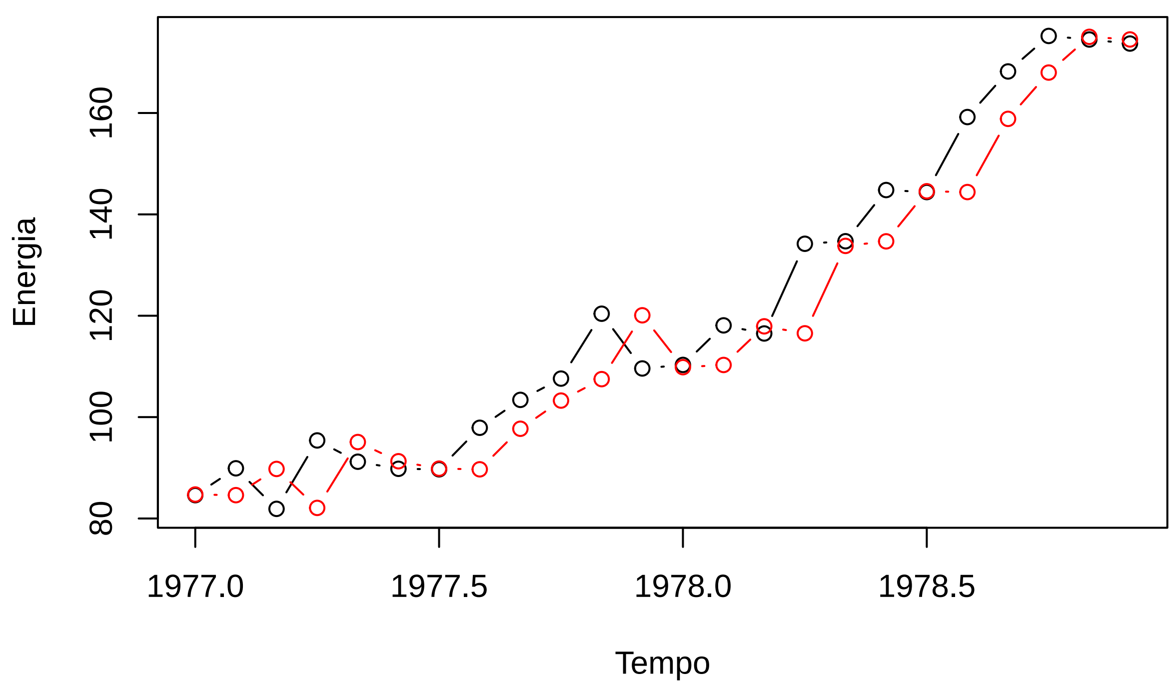



```
# Ajuste do modelo de suavização exponencial simples
modelo_ses <- ses(energia_ts, h = 5)  # h = número de períodos para previsão
summary(modelo_ses)
```

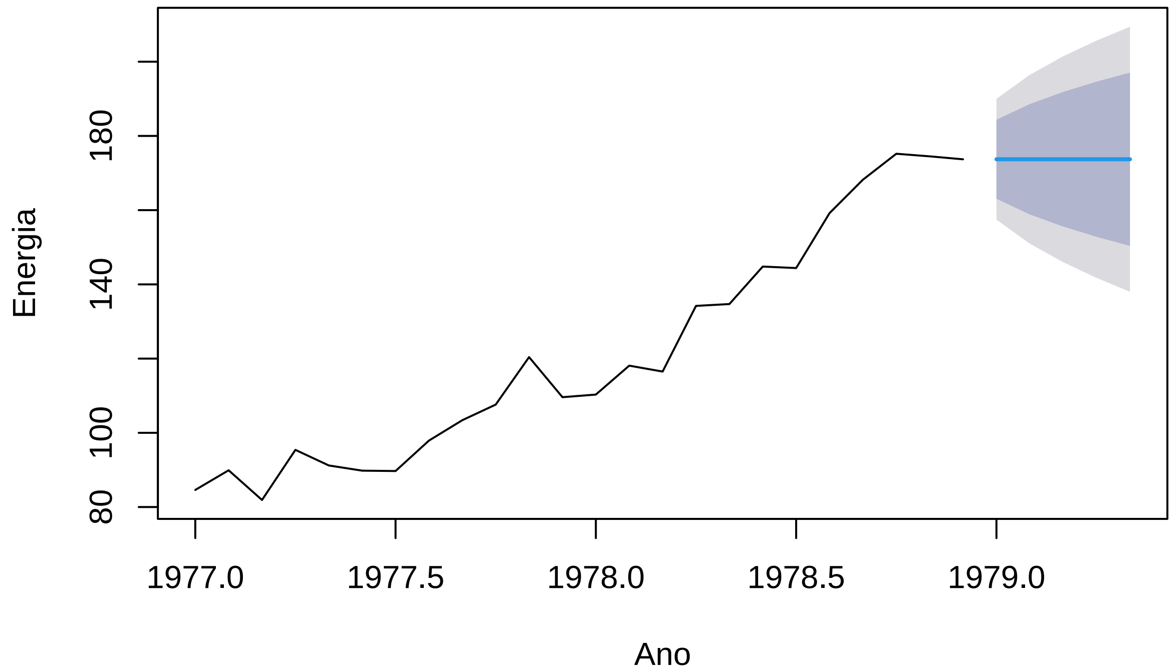
```
##
## Forecast method: Simple exponential smoothing
##
## Model Information:
## Simple exponential smoothing
##
## Call:
## ses(y = energia_ts, h = 5)
##
```

```
##   Smoothing parameters:
##     alpha = 0.9761
##
##   Initial states:
##     l = 84.7297
##
##   sigma:  8.306
##
##      AIC     AICc      BIC
## 181.7999 182.9999 185.3340
##
## Error measures:
##                   ME    RMSE      MAE      MPE     MAPE      MASE       ACF1
## Training set 3.79869 7.95237 6.041726 2.825423 5.117265 0.1223847 -0.3237923
##
## Forecasts:
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Jan 1979       173.7194 163.0749 184.3640 157.4400 189.9988
## Feb 1979       173.7194 158.8446 188.5943 150.9703 196.4686
## Mar 1979       173.7194 155.5751 191.8637 145.9701 201.4688
## Apr 1979       173.7194 152.8108 194.6281 141.7424 205.6964
## May 1979       173.7194 150.3715 197.0674 138.0118 209.4270
```

```r
# visualizacao do ajuste
plot(energia_ts, type="b", ylab="Energia", xlab="Tempo")
lines(modelo_ses$fitted, type="b", col="red")
```



```r
# Visualizacao da previsao
plot(modelo_ses, main="", xlab="Ano", ylab="Energia")
```

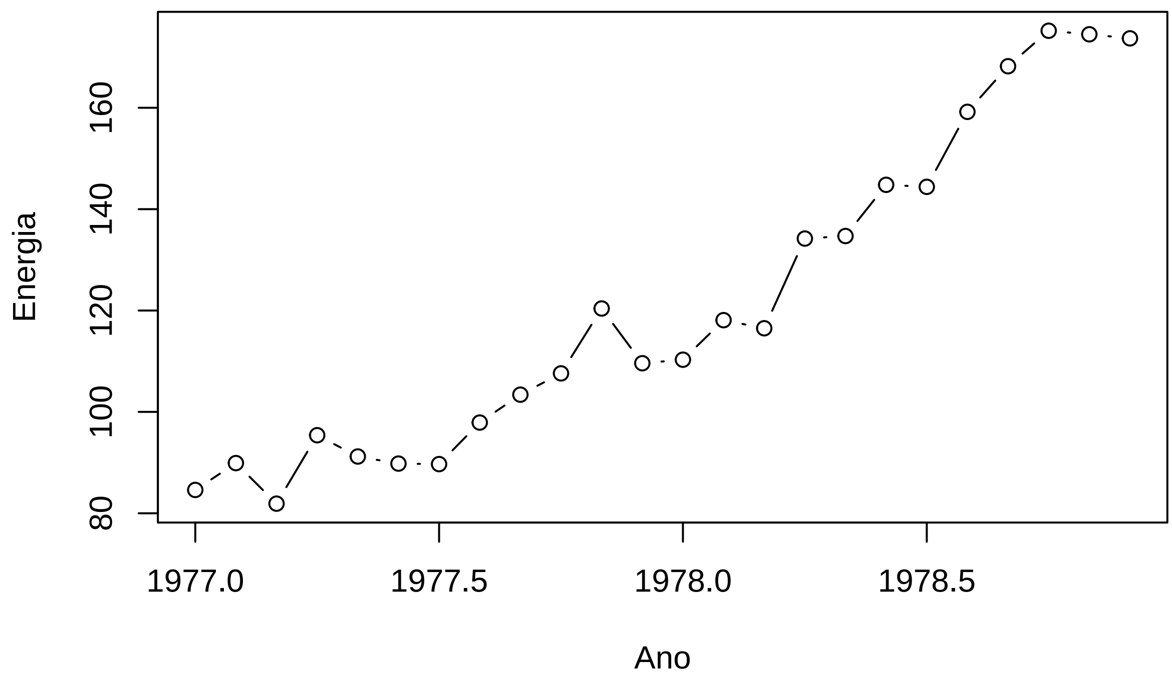


Suavização Exponencial de Holt (SEH)

Exemplo 1

A seguir apresentamos os valores mensais do consumo de energia elétrica no Estado do Espírito Santo, referentes aos anos 1977 e 1978, portanto são 24 observações. Série temporal retirada de Morettin e Toloi (2006).

```r
energia <- c(84.6, 89.9, 81.9, 95.4, 91.2, 89.8, 89.7, 97.9, 103.4,
       107.6, 120.4, 109.6, 110.3, 118.1, 116.5, 134.2, 134.7,
       144.8, 144.4, 159.2, 168.2, 175.2, 174.5, 173.7)

energia_ts <- ts(energia, start=c(1977, 1), frequency = 12)

# grafico da serie temporal
plot(energia_ts, type='b', ylab = "Energia", xlab="Ano")
```

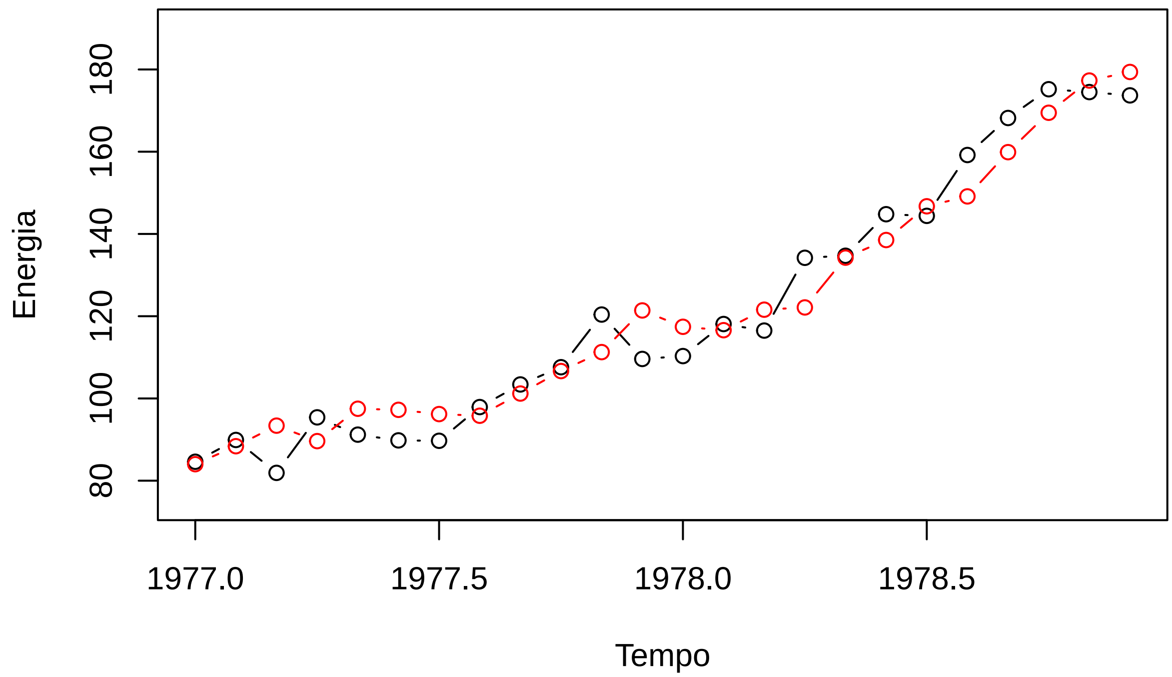



```r
# Ajuste do modelo de suavização exponencial simples
modelo_seh <- holt(energia_ts, h = 5)
summary(modelo_seh)
```

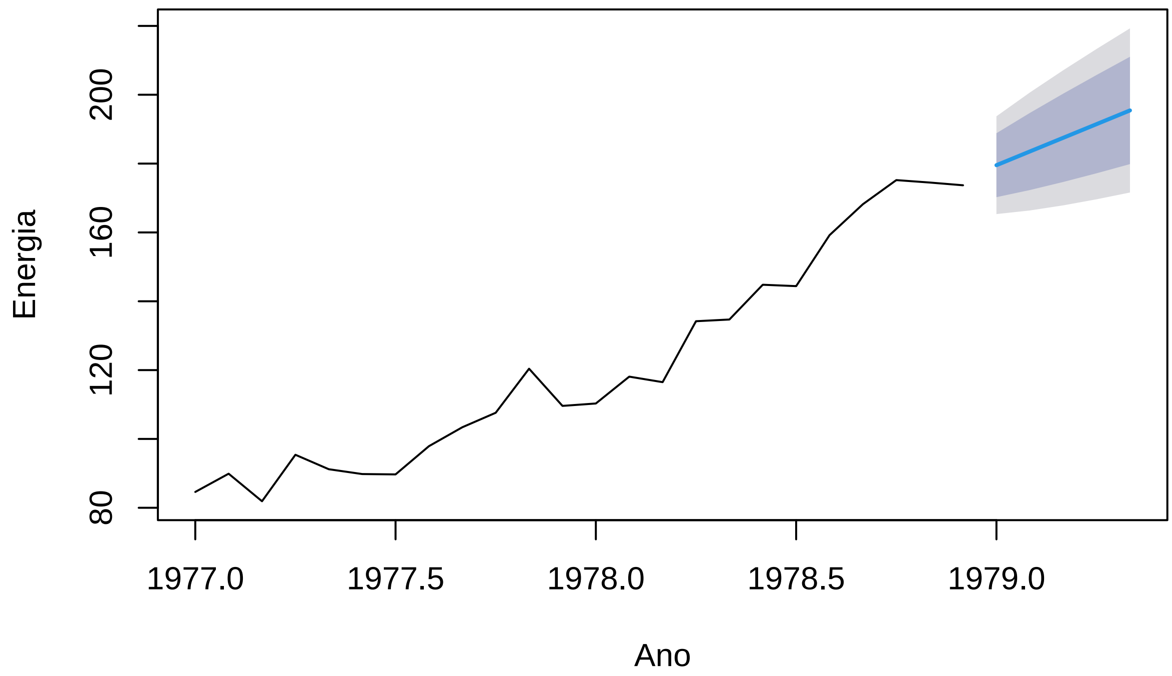
```
##
## Forecast method: Holt's method
##
## Model Information:
## Holt's method
##
## Call:
```

```
## holt(y = energia_ts, h = 5)
##
##   Smoothing parameters:
##     alpha = 0.6739
##     beta  = 1e-04
##
##   Initial states:
##     l = 80.033
##     b = 3.9773
##
##   sigma:  7.2479
##
##      AIC     AICc      BIC
## 176.9719 180.3052 182.8621
##
## Error measures:
##                       ME     RMSE      MAE        MPE    MAPE      MASE
## Training set 0.006013315 6.616414 5.551099 -0.5775396 4.79958 0.1124463
##                    ACF1
## Training set -0.0233501
##
## Forecasts:
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Jan 1979       179.5341 170.2455 188.8227 165.3285 193.7398
## Feb 1979       183.5115 172.3100 194.7129 166.3803 200.6426
## Mar 1979       187.4888 174.6560 200.3216 167.8628 207.1148
## Apr 1979       191.4662 177.1869 205.7454 169.6279 213.3045
## May 1979       195.4435 179.8509 211.0361 171.5967 219.2903
```

```
# comparando com as metricas obtidas no ajuste do modelo de SES
## Error measures:
## ME RMSE MAE MPE MAPE MASE ACF1
## Training set 3.79869 7.95237 6.041726 2.825423 5.117265 0.1223847 -0.3237923

# visualizacao do ajuste
plot(energia_ts, type="b", ylab="Energia", xlab="Tempo", ylim=c(75, 190))
lines(modelo_seh$fitted, type="b", col="red")
```




```r
# Visualizacao da previsao
plot(modelo_seh, main="", xlab="Ano", ylab="Energia")
```




```r
##########
##########

# considere agora previsoes com atualizacoes

ajuste1 <- holt(energia_ts[1:20], h = 5)
summary(ajuste1)
```

```
##
## Forecast method: Holt's method
##
## Model Information:
## Holt's method
##
## Call:
## holt(y = energia_ts[1:20], h = 5)
##
##   Smoothing parameters:
##     alpha = 0.11
##     beta  = 0.11
##
##   Initial states:
##     l = 82.1054
##     b = 1.1585
##
##   sigma:  7.3313
```

```
##
##      AIC     AICc      BIC
## 145.1377 149.4234 150.1164
##
## Error measures:
##                    ME     RMSE      MAE      MPE     MAPE     MASE       ACF1
## Training set 2.661063 6.557287 5.350389 2.019704 4.753574 0.796688 0.07177193
##
## Forecasts:
##    Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 21       156.9896 147.5942 166.3850 142.6206 171.3587
## 22       164.0015 154.3815 173.6215 149.2890 178.7140
## 23       171.0134 160.9062 181.1205 155.5558 186.4709
## 24       178.0253 167.1056 188.9449 161.3251 194.7254
## 25       185.0371 172.9569 197.1174 166.5620 203.5123
```

```
# previsao para o instante 21
# vamos utilizar os parametros obtidos
# no ajuste utilizando o conjunto de treinamento
# alpha = 0.11
# beta = 0.11

holt(energia_ts[1:21], h = 1, alpha = 0.11, beta = 0.11)$mean
```

```
## Time Series:
## Start = 22
## End = 22
## Frequency = 1
## [1] 166.6731
```

```
# previsao para o instante 22
holt(energia_ts[1:22], h = 1, alpha = 0.9, beta = 0.3)$mean
```

```
## Time Series:
## Start = 23
## End = 23
## Frequency = 1
## [1] 183.2026
```

```
# previsao para o instante 23
holt(energia_ts[1:23], h = 1, alpha = 0.9, beta = 0.3)$mean
```

```
## Time Series:
## Start = 24
## End = 24
## Frequency = 1
## [1] 180.6684
```

```
# previsao para o instante 24
holt(energia_ts[1:24], h = 1, alpha = 0.9, beta = 0.3)$mean
```

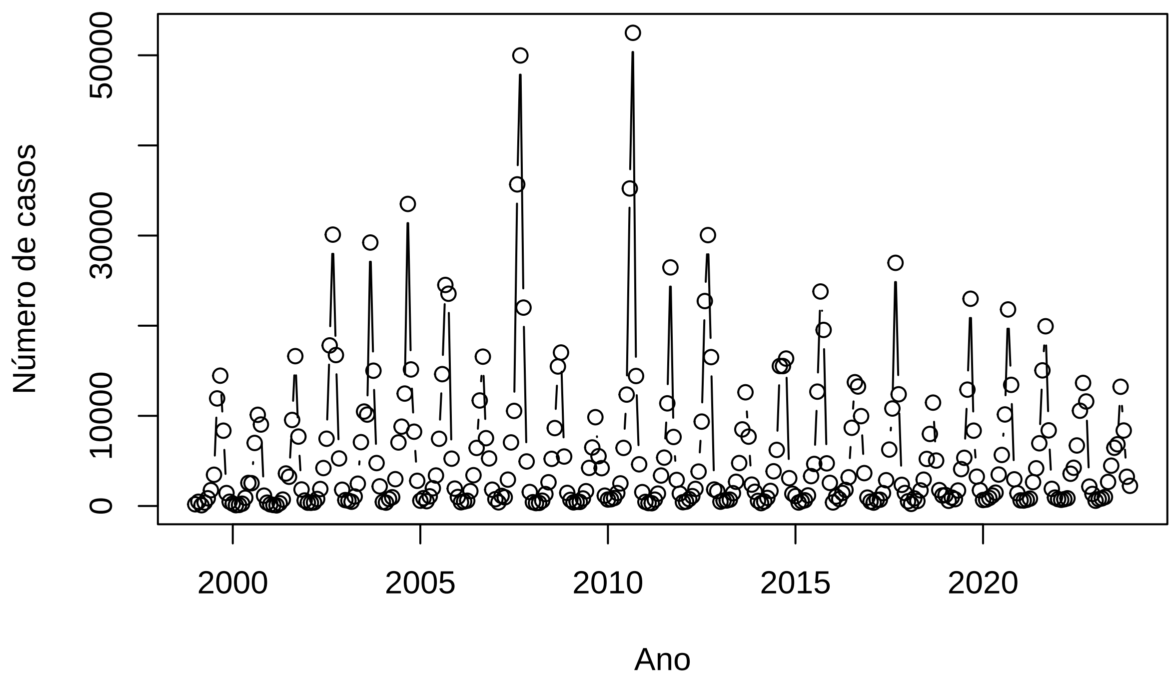
```
## Time Series:
## Start = 25
## End = 25
## Frequency = 1
## [1] 177.6044
```

Suavização Exponencial Holt-Winters (HW)

Exemplo 1

A série temporal em estudo é do número mensal de casos de queimadas no bioma cerrado no período de janeiro de 1999 até dezembro de 2023. Os valores foram obtidos no site do INPE, para acessar click aqui.

```r
# numero mensal de casos de queimada no bioma cerrado

serie_queimada <-
  c(188, 469,   82, 382,    856,    1757,    3465,    11933, 14455,   8354,   1434,   475,
  266, 89,  114,    253,    914,    2556,    2530,    6995, 10117,    9038,   1149,   372,
  182, 109, 69, 304,    713,    3606,    3242,    9541, 16623,    7702,   1834,   609,
  368, 355, 391,    777,    1884,    4210,    7462,    17815, 30105,   16744,   5273,   1802,
  654, 587, 472,    1052,    2475,    7079,    10477,   10140, 29227,   15008,   4763,   2179,
  452, 376, 773,    970,    2975,    7047,    8804,    12473, 33509,   15148,   8243,   2792,
  587, 869, 536,    1071,    1968,    3383,    7454,    14627, 24521,   23550,   5251,   1930,
  1001, 413,    498,    584,    1650,    3415,    6442,    11705, 16566,   7524,   5274,   1808,
  755, 402, 1118,    964,    2924,    7051,    10548,   35678, 49980,   22008,   4939,   1551,
  421, 321, 339,    590,    1320,    2632,    5229,    8643,    15477, 17024,   5486,   1452,
  676, 403, 461,    448,    836,    1641,    4224,    6492,    9851, 5526, 4195,   1146,
  717, 751, 883,    1438,    2508,    6443,    12359,   35226, 52491,   14419,   4623,   1536,
  456, 321, 308,    703,    1376,    3378,    5366,    11387, 26468,   7656,   2889,   1374,
  421, 478, 764,    1092,    1896,    3817,    9362,    22737, 30053,   16515,   1830,   1635,
  475, 604, 613,    706,    1418,    2684,    4761,    8496, 12615,   7696,   2370,   1579,
  567, 315, 497,    897,    1673,    3849,    6220,    15525, 15523,   16357,   3085,   1363,
  1096, 383,    528,    634,    1174,    3313,    4662,    12684, 23795,   19531,   4731,   2563,
  392, 1048,    767,    1449,    1782,    3187,    8675,    13730, 13256,   9968,   3647,   932,
  491, 376, 687,    696,    1428,    2858,    6258,    10815,   26975, 12393,   2345,   1440,
  521, 235, 842,    549,    1729,    2922,    5220,    7992,    11467, 5041,    1763,   1168,
  1213, 574,    936,    753,    1719,    4088,    5346,    12906, 22989,   8356,   3251,   1743,
  662, 709, 890,    1158,    1481,    3487,    5663,    10155, 21802,   13440,   2957,   1415,
  625, 607, 686,    820,    2649,    4181,    6955,    15043, 19939,   8389,   1901,   933,
  763, 685, 751,    853,    3578,    4239,    6713,    10567, 13651,   11594,   2154,   1337,
  585, 759, 845,    993,    2668,    4472,    6459,    6850, 13230,    8371,   3242,   2239)


queimada_ts <- ts(serie_queimada, start= c(1999, 1), frequency = 12)

plot(queimada_ts, type="b", ylab="Número de casos", xlab="Ano")
```




```r
# No software R e possível encontrar pelo menos duas funções do
# metodo de suavização exponencial sazonal de Holt-Winters
# Em ambos os metodos que serao apresentados a seguir e preciso
# definir a serie temporal em um objeto do tipo *ts*

# usando a função Holt-Winters do pacote básico stats
# modelo multiplicativo
ajuste1 <- HoltWinters(queimada_ts, seasonal = "multiplicative")
ajuste1
```

```
## Holt-Winters exponential smoothing with trend and multiplicative seasonal component.
##
## Call:
## HoltWinters(x = queimada_ts, seasonal = "multiplicative")
##
## Smoothing parameters:
##  alpha: 0.3174661
##  beta : 0
##  gamma: 0.3983397
##
## Coefficients:
##              [,1]
## a   8868.61280094
## b    -85.57794289
## s1     0.09888259
## s2     0.09105427
```

```
## s3     0.10538005
## s4     0.11669204
## s5     0.29729072
## s6     0.46032158
## s7     0.72675819
## s8     1.20371131
## s9     2.10443859
## s10    1.31747720
## s11    0.37013307
## s12    0.21455610
```

```r
head(ajuste1$fitted)
```

```
##                xhat    level     trend     season
## Jan 2000  269.87689 3935.614 -85.57794 0.07009723
## Feb 2000   94.18703 3832.478 -85.57794 0.02513732
## Mar 2000  130.58615 3681.392 -85.57794 0.03631616
## Apr 2000  285.61519 3450.822 -85.57794 0.08487206
## May 2000  962.60554 3243.246 -85.57794 0.30484695
## Jun 2000 2590.32574 3107.051 -85.57794 0.85730567
```

```r
ajuste1$alpha
```

```
##     alpha
## 0.3174661
```

```r
ajuste1$beta
```

```
## beta
##    0
```

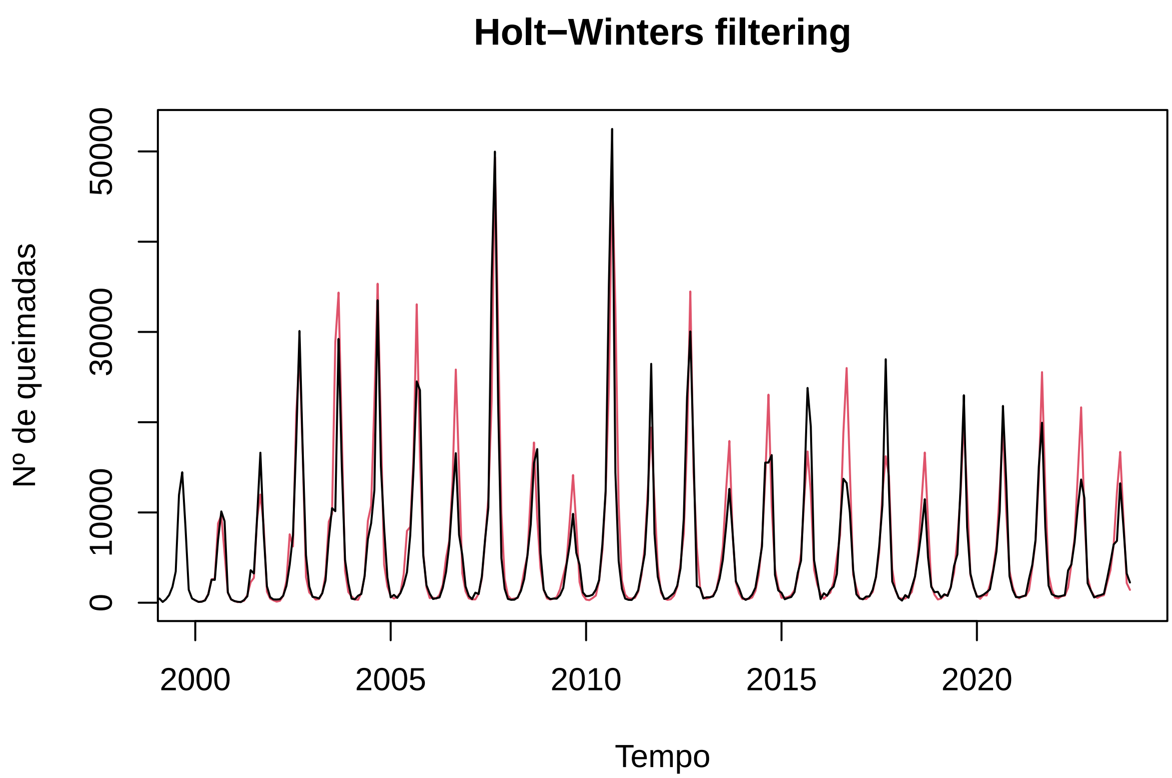
```r
ajuste1$gamma
```

```
##     gamma
## 0.3983397
```

```r
ajuste1$coefficients
```

```
##             a             b            s1            s2            s3
## 8868.61280094  -85.57794289    0.09888259    0.09105427    0.10538005
##            s4            s5            s6            s7            s8
##    0.11669204    0.29729072    0.46032158    0.72675819    1.20371131
##            s9           s10           s11           s12
##    2.10443859    1.31747720    0.37013307    0.21455610
```

```r
plot(ajuste1, xlab="Tempo", ylab="Nº de queimadas")
```




```r
# nao e possível fazer previsao direto da funcao HoltWinters
# e necessario fazer uso das formulas ou da funcao forecast

forecast(ajuste1, h=12)
```

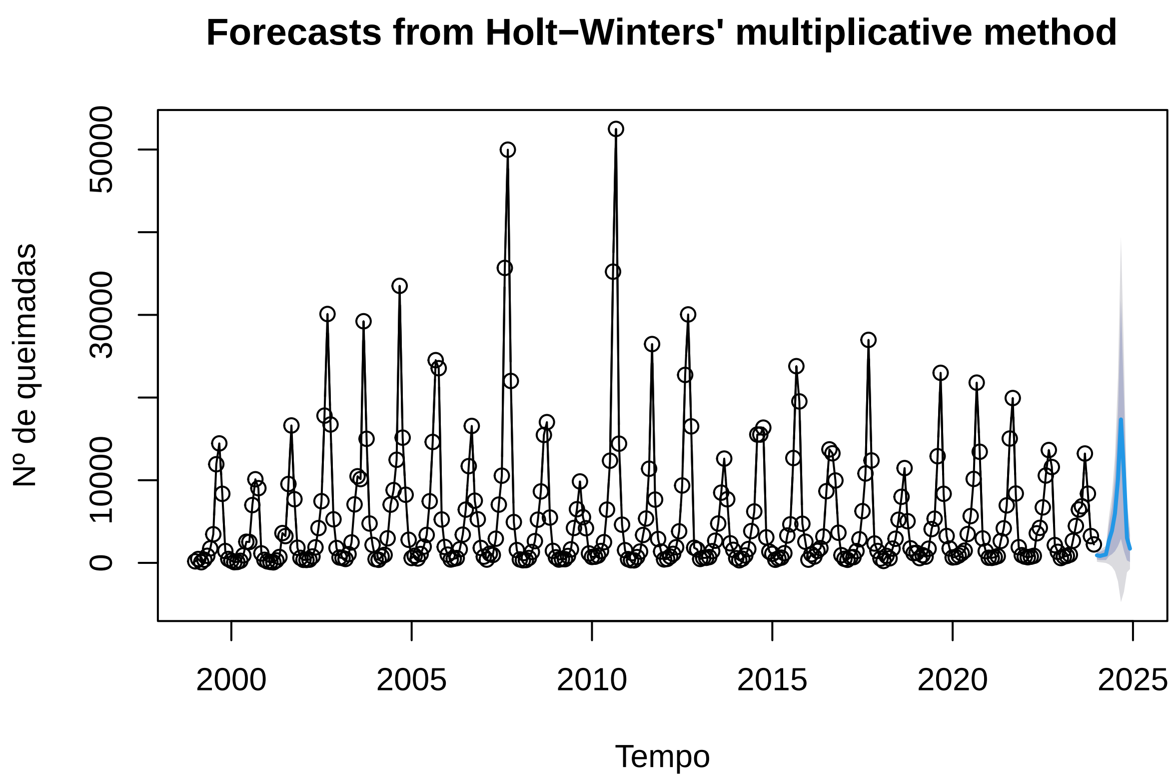
```
##          Point Forecast       Lo 80     Hi 80      Lo 95    Hi 95
## Jan 2024       868.4893  -1492.109  3229.087  -2741.733  4478.712
## Feb 2024       791.9406  -1843.109  3426.991  -3238.020  4821.902
## Mar 2024       907.5202  -2186.769  4001.810  -3824.788  5639.828
## Apr 2024       994.9515  -2535.304  4525.207  -4404.109  6394.012
## May 2024      2509.3487  -5287.375 10306.073  -9414.712 14433.409
## Jun 2024      3846.0537  -8063.229 15755.336 -14367.622 22059.730
## Jul 2024      6009.9757 -12678.113 24698.064 -22570.989 34590.940
## Aug 2024      9851.1604 -21016.067 40718.387 -37356.190 57058.510
## Sep 2024     17042.6093 -36861.889 70947.108 -65397.207 99482.425
## Oct 2024     10556.7253 -23249.558 44363.009 -41145.523 62258.974
## Nov 2024      2934.1394  -6836.426 12704.704 -12008.650 17876.929
## Dec 2024      1682.4797  -3862.678  7227.637  -6798.106 10163.066
```

```r
# Usando o pacote forecast
library(forecast)

# modelo multiplicativo
ajuste2 <- hw(queimada_ts, h = 12, seasonal = "multiplicative", initial = "simple")
summary(ajuste2)
```

```
##
```

```
## Forecast method: Holt-Winters' multiplicative method
##
## Model Information:
## Holt-Winters' multiplicative method
##
## Call:
## hw(y = queimada_ts, h = 12, seasonal = "multiplicative", initial = "simple")
##
##   Smoothing parameters:
##     alpha = 0.3334
##     beta  = 0
##     gamma = 0.2621
##
##   Initial states:
##     l = 3654.1667
##     b = -65.6736
##     s = 0.13 0.3924 2.2862 3.9558 3.2656 0.9482
##            0.4808 0.2343 0.1045 0.0224 0.1283 0.0514
##
##   sigma:  0.4332
## Error measures:
##                     ME     RMSE      MAE       MPE     MAPE     MASE      ACF1
## Training set -298.4325 3013.806 1490.634 -8.623385 30.91856 0.532473 0.1049345
##
## Forecasts:
##          Point Forecast     Lo 80     Hi 80       Lo 95     Hi 95
## Jan 2024       925.0232  411.5283  1438.518   139.70061  1710.346
## Feb 2024       832.9785  340.4324  1325.525    79.69418  1586.263
## Mar 2024       932.8419  348.1567  1517.527    38.64301  1827.041
## Apr 2024      1017.2087  344.1095  1690.308   -12.20775  2046.625
## May 2024      2694.0164  818.2918  4569.741  -174.65687  5562.690
## Jun 2024      3893.9772 1049.2986  6738.656  -456.58327  8244.538
## Jul 2024      5985.5050 1408.7014 10562.309 -1014.11211 12985.122
## Aug 2024     10066.1911 2026.4131 18105.969 -2229.58823 22361.970
## Sep 2024     17348.3818 2901.6441 31795.120 -4745.99693 39442.761
## Oct 2024     10702.0329 1424.7202 19979.346 -3486.39248 24890.458
## Nov 2024      3016.6091  298.1923  5735.026 -1140.85064  7174.069
## Dec 2024      1723.0681  110.9129  3335.223  -742.51044  4188.647
```

```r
plot(ajuste2, xlab="Tempo", ylab="Nº de queimadas")
lines(queimada_ts, type="b")
```

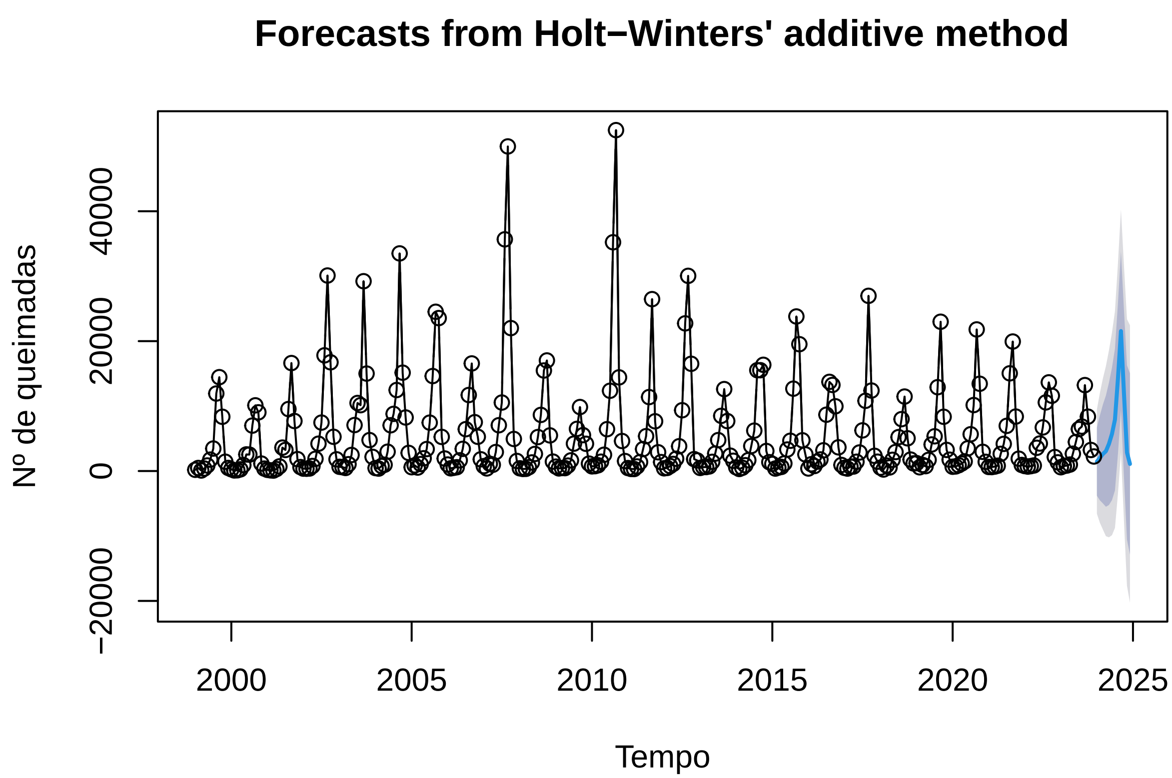



```r
# modelo aditivo
ajuste3 <- hw(queimada_ts, h = 12, seasonal = "additive", initial = "simple")
summary(ajuste3)
```

```
##
## Forecast method: Holt-Winters' additive method
##
## Model Information:
## Holt-Winters' additive method
##
## Call:
## hw(y = queimada_ts, h = 12, seasonal = "additive", initial = "simple")
##
##   Smoothing parameters:
##     alpha = 0.743
##     beta  = 0
##     gamma = 0.0644
##
##   Initial states:
##     l = 3654.1667
##     b = -65.6736
##     s = -3179.167 -2220.167 4699.833 10800.83 8278.833 -189.1667
##            -1897.167 -2798.167 -3272.167 -3572.167 -3185.167 -3466.167
##
##   sigma:  4099.153
## Error measures:
```

```
##                     ME     RMSE      MAE       MPE     MAPE     MASE       ACF1
## Training set 102.5687 4099.153 2305.793 -41.57863 83.92015 0.823658 0.08118103
##
## Forecasts:
##          Point Forecast       Lo 80     Hi 80      Lo 95     Hi 95
## Jan 2024       1453.713  -3799.5631  6706.989  -6580.479  9487.905
## Feb 2024       2031.487  -4513.1555  8576.129  -7977.680 12040.654
## Mar 2024       2632.216  -4988.0053 10252.438  -9021.907 14286.340
## Apr 2024       3062.973  -5498.7563 11624.703 -10031.062 16157.009
## May 2024       4179.725  -5229.7732 13589.222 -10210.861 18570.310
## Jun 2024       5699.034  -4487.9231 15885.990  -9880.572 21278.639
## Jul 2024       7922.369  -2986.7795 18831.517  -8761.734 24606.472
## Aug 2024      14228.667   2642.2543 25815.079  -3491.222 31948.556
## Sep 2024      21550.656   9324.4388 33776.873   2852.270 40249.042
## Oct 2024      11937.157   -897.0094 24771.323  -7691.007 31565.321
## Nov 2024       2853.214 -10561.3776 16267.805 -17662.633 23369.061
## Dec 2024       1089.101 -12881.8224 15060.024 -20277.582 22455.784
```

```r
plot(ajuste3, xlab="Tempo", ylab="Nº de queimadas")
lines(queimada_ts, type="b")
```




```r
# qual o melhor modelo?
# ajuste2
#                     ME     RMSE      MAE       MPE     MAPE     MASE      ACF1
#Training set -298.4325 3013.806 1490.634 -8.623385 30.91856 0.532473 0.1049345
# ajuste3
#                   ME     RMSE      MAE       MPE     MAPE     MASE        ACF1
```

```r
#Training set 102.5687 4099.153 2305.793 -41.57863 83.92015 0.823658 0.08118103

# utilizando RMSE ou MAE ou MAPE para responder a pergunta
# a resposta e ajuste2 = modelo multiplicativo

# interpretacao dos parametros
#    alpha = 0.743
#    beta  = 0
#    gamma = 0.0644

# as previsoes sao
ajuste2$mean
```

```
##              Jan        Feb        Mar        Apr        May        Jun
## 2024   925.0232   832.9785   932.8419  1017.2087  2694.0164  3893.9772
##              Jul        Aug        Sep        Oct        Nov        Dec
## 2024  5985.5050 10066.1911 17348.3818 10702.0329  3016.6091  1723.0681
```

```r
# as estimativas intervalares sao
ajuste2$lower # limite inferior
```

```
##                80%         95%
## Jan 2024  411.5283   139.70061
## Feb 2024  340.4324    79.69418
## Mar 2024  348.1567    38.64301
## Apr 2024  344.1095   -12.20775
## May 2024  818.2918  -174.65687
## Jun 2024 1049.2986  -456.58327
## Jul 2024 1408.7014 -1014.11211
## Aug 2024 2026.4131 -2229.58823
## Sep 2024 2901.6441 -4745.99693
## Oct 2024 1424.7202 -3486.39248
## Nov 2024  298.1923 -1140.85064
## Dec 2024  110.9129  -742.51044
```

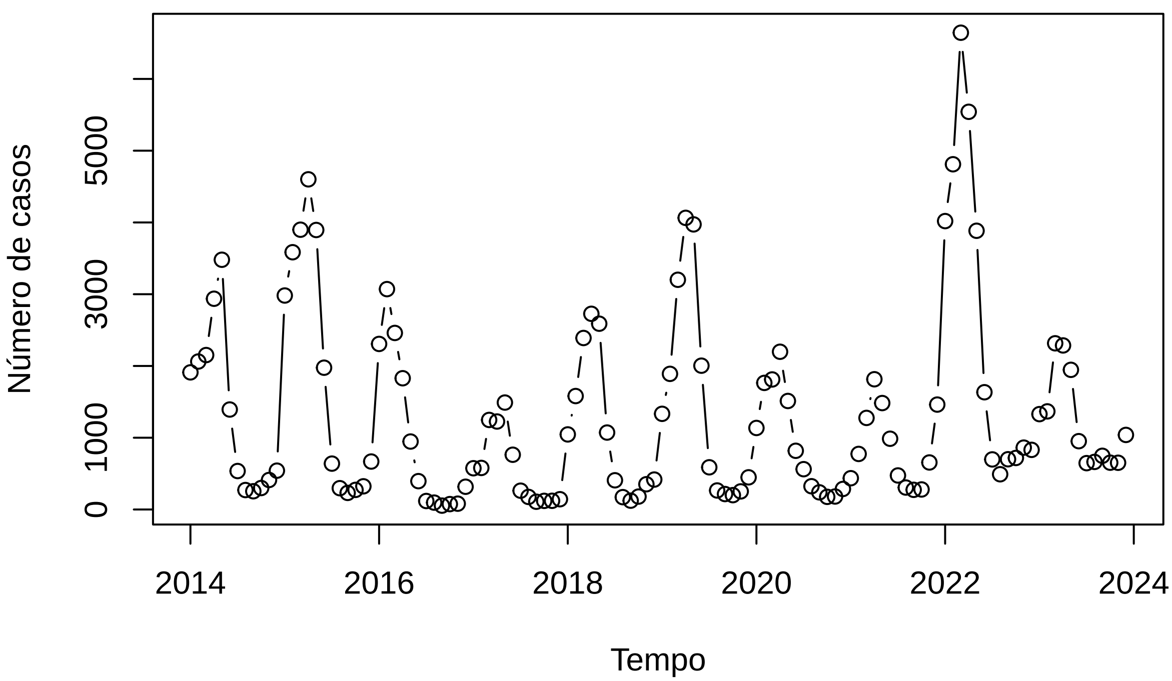
```r
ajuste2$upper # limite superior
```

```
##                80%       95%
## Jan 2024  1438.518  1710.346
## Feb 2024  1325.525  1586.263
## Mar 2024  1517.527  1827.041
## Apr 2024  1690.308  2046.625
## May 2024  4569.741  5562.690
## Jun 2024  6738.656  8244.538
## Jul 2024 10562.309 12985.122
## Aug 2024 18105.969 22361.970
## Sep 2024 31795.120 39442.761
## Oct 2024 19979.346 24890.458
## Nov 2024  5735.026  7174.069
## Dec 2024  3335.223  4188.647
```

Exemplo 2

A seguir é apresentado a série temporal do número de casos de dengue no Estado de Goiás (UF da residência é Goiás) com exame sorológico (IgM) positivo, compreende o número de casos mensal de janeiro de 2014 a dezembro de 2023. Os valores foram obtido a partir do Sistema de Informação de Agravos de Notificação - Sinan Net - acessado em 20/09/2024, no link aqui.

```
# serie do numero de casos mensal
serie_dengue <- c(1911, 2062, 2152, 2937, 3480, 1393, 536, 271, 254, 299, 411, 541,
                  2982, 3585, 3899, 4601, 3895, 1976, 639, 297, 231, 273, 323, 668,
                  2307, 3071, 2460, 1829, 947, 395, 118, 96, 54, 75, 81, 317,
                  575, 579, 1248, 1226, 1490, 763, 262, 175, 108, 120, 121, 143,
                  1046, 1581, 2390, 2727, 2589, 1072, 407, 172, 123, 180, 352, 417,
                  1333, 1890, 3201, 4064, 3973, 2004, 588, 265, 215, 200, 252, 448,
                  1136, 1764, 1812, 2198, 1512, 818, 562, 324, 239, 176, 181, 286,
                  436, 774, 1276, 1815, 1483, 986, 474, 306, 274, 279, 655, 1462,
                  4019, 4811, 6644, 5542, 3884, 1634, 698, 491, 700, 718, 862, 831,
                  1329, 1367, 2316, 2286, 1946, 952, 646, 663, 748, 652, 651, 1039)

dengue_ts <- ts(serie_dengue, start= c(2014, 1), frequency = 12)

plot(dengue_ts, type="b", ylab="Número de casos", xlab="Tempo")
```

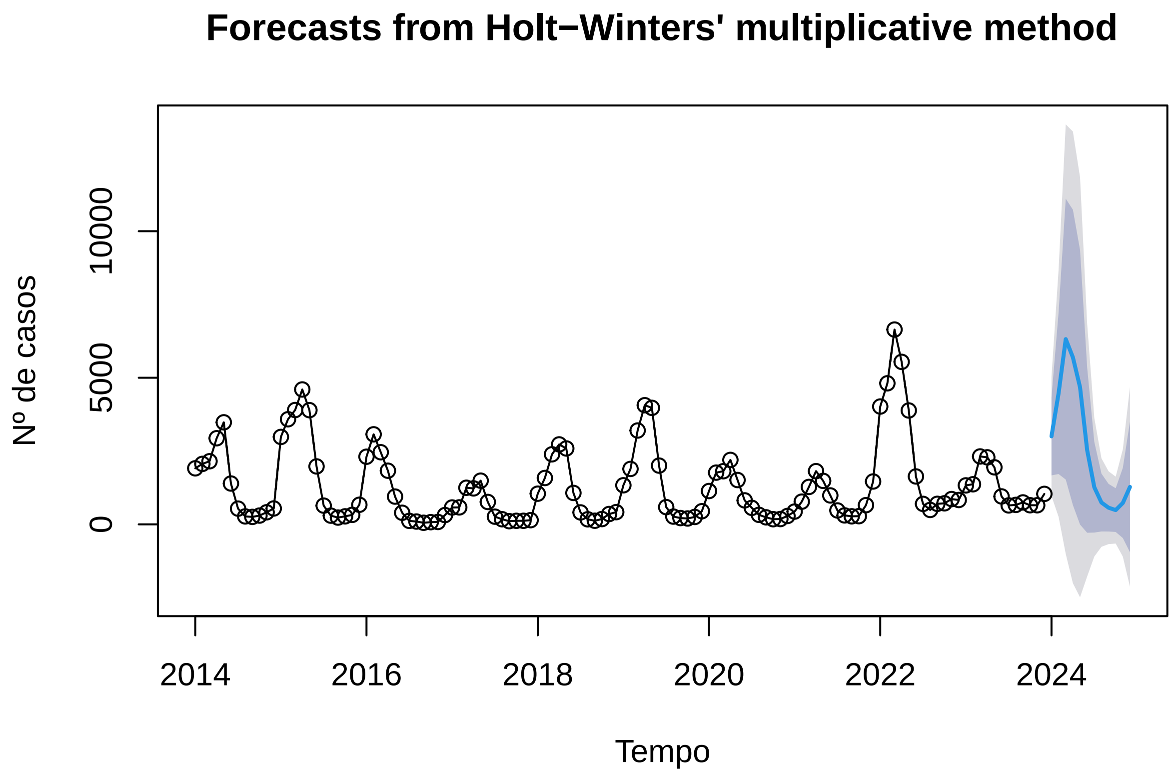



```
# Usando o pacote forecast
#library(forecast)

# modelo multiplicativo
```

```r
ajuste1 <- hw(dengue_ts, h = 12, seasonal = "multiplicative", initial = "simple")
summary(ajuste1)
```

```
##
## Forecast method: Holt-Winters' multiplicative method
##
## Model Information:
## Holt-Winters' multiplicative method
##
## Call:
## hw(y = dengue_ts, h = 12, seasonal = "multiplicative", initial = "simple")
##
##   Smoothing parameters:
##     alpha = 0.9297
##     beta  = 0
##     gamma = 0.2108
##
##   Initial states:
##     l = 1353.9167
##     b = 49.4583
##     s = 0.3996 0.3036 0.2208 0.1876 0.2002 0.3959
##            1.0289 2.5703 2.1693 1.5895 1.523 1.4115
##
##   sigma:  0.3461
## Error measures:
##                     ME     RMSE      MAE      MPE     MAPE     MASE      ACF1
## Training set -73.06769 465.9379 298.2452 -10.9462 27.38975 0.359524 0.1859611
##
## Forecasts:
##          Point Forecast       Lo 80     Hi 80      Lo 95     Hi 95
## Jan 2024      3002.2058 1670.431122  4333.981   965.4322  5038.979
## Feb 2024      4474.6690 1716.257417  7233.081   256.0425  8693.295
## Mar 2024      6318.1922 1527.990560 11108.394 -1007.7890 13644.173
## Apr 2024      5695.7765  658.414788 10733.138 -2008.2034 13399.756
## May 2024      4675.7903   -7.411566  9358.992 -2486.5489 11838.129
## Jun 2024      2531.4028 -286.284653  5349.090 -1777.8783  6840.684
## Jul 2024      1266.0462 -279.788911  2811.881 -1098.1046  3630.197
## Aug 2024       743.8563 -242.999325  1730.712  -765.4092  2253.122
## Sep 2024       567.4583 -244.650589  1379.567  -674.5550  1809.472
## Oct 2024       485.9143 -260.027341  1231.856  -654.9050  1626.734
## Nov 2024       730.1993 -466.791671  1927.190 -1100.4404  2560.839
## Dec 2024      1276.7176 -949.929043  3503.364 -2128.6446  4682.080
```

```r
plot(ajuste1, xlab="Tempo", ylab="Nº de casos")
lines(dengue_ts, type="b")
```

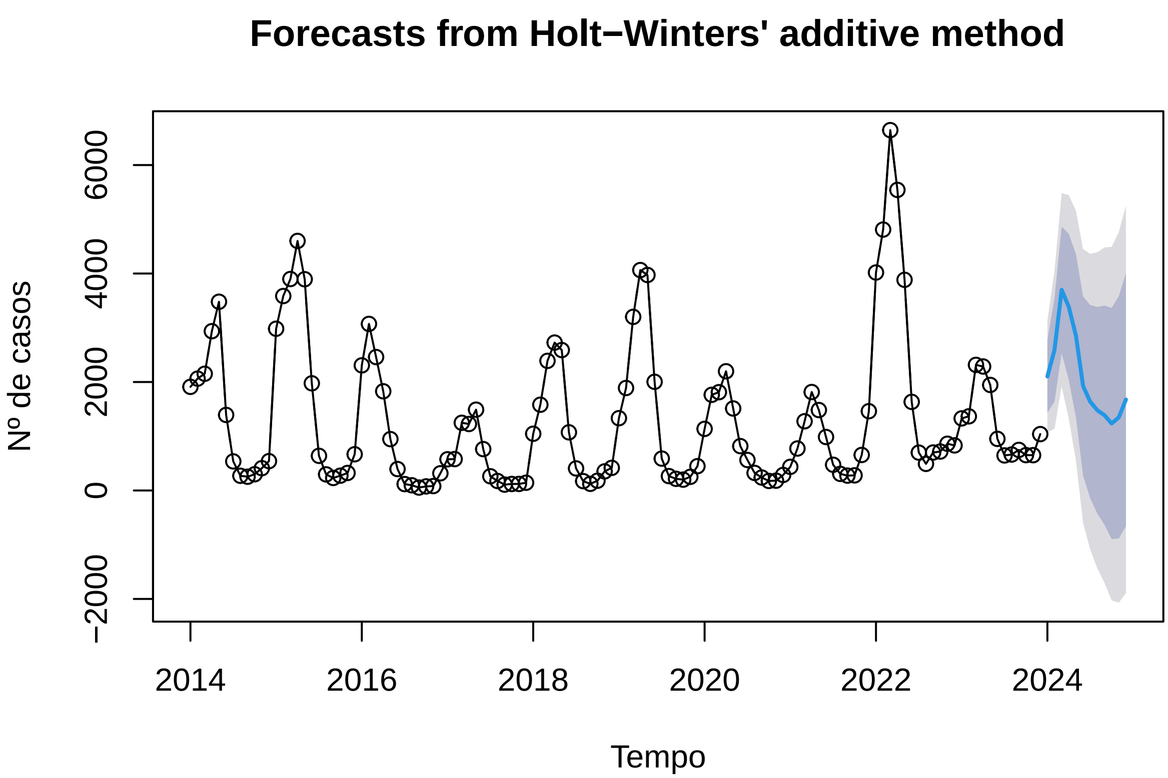



```r
# modelo aditivo
ajuste2 <- hw(dengue_ts, h = 12, seasonal = "additive", initial = "simple")
summary(ajuste2)
```

```
##
## Forecast method: Holt-Winters' additive method
##
## Model Information:
## Holt-Winters' additive method
##
## Call:
## hw(y = dengue_ts, h = 12, seasonal = "additive", initial = "simple")
##
##   Smoothing parameters:
##     alpha = 1
##     beta  = 0
##     gamma = 0.2881
##
##   Initial states:
##     l = 1353.9167
##     b = 49.4583
##     s = -812.9167 -942.9167 -1054.917 -1099.917 -1082.917 -817.9167
##            39.0833 2126.083 1583.083 798.0833 708.0833 557.0833
##
##   sigma:  525.8293
## Error measures:
```

```
##                     ME     RMSE    MAE       MPE     MAPE      MASE      ACF1
## Training set -44.76212 525.8293 349.22 -5.134708 48.00331 0.4209724 0.2141061
##
## Forecasts:
##          Point Forecast     Lo 80    Hi 80      Lo 95    Hi 95
## Jan 2024       2103.023 1429.1451 2776.900  1072.4160 3133.629
## Feb 2024       2592.644 1639.6376 3545.651  1135.1464 4050.142
## Mar 2024       3699.047 2531.8570 4866.237  1913.9840 5484.110
## Apr 2024       3383.169 2035.4145 4730.924  1321.9562 5444.382
## May 2024       2852.689 1345.8533 4359.525   548.1827 5157.195
## Jun 2024       1926.763  276.1069 3577.418  -597.6974 4451.223
## Jul 2024       1636.009 -146.9033 3418.921 -1090.7199 4362.737
## Aug 2024       1478.373 -427.6405 3384.386 -1436.6229 4393.368
## Sep 2024       1388.115 -633.5170 3409.747 -1703.7045 4479.935
## Oct 2024       1234.582 -896.4051 3365.570 -2024.4817 4493.646
## Nov 2024       1348.697 -886.3014 3583.696 -2069.4382 4766.832
## Dec 2024       1676.589 -657.7906 4010.969 -1893.5366 5246.715
```

```
plot(ajuste2, xlab="Tempo", ylab="Nº de casos")
lines(dengue_ts, type="b")
```




```
# qual o melhor modelo?
# ajuste1
#                     ME     RMSE      MAE       MPE     MAPE     MASE      ACF1
#Training set -73.06769 465.9379 298.2452 -10.9462 27.38975 0.359524 0.1859611
# ajuste2
#                     ME     RMSE    MAE       MPE     MAPE      MASE      ACF1
```

```r
#Training set -44.76212 525.8293 349.22 -5.134708 48.00331 0.4209724 0.2141061

# utilizando RMSE ou MAE ou MAPE para responder a pergunta
# a resposta e ajuste3 = modelo multiplicativo

# interpretacao dos parametros
#    alpha = 1
#    beta  = 0
#    gamma = 0.2881

# as previsoes sao
ajuste1$mean
```

```
##            Jan       Feb       Mar       Apr       May       Jun       Jul
## 2024 3002.2058 4474.6690 6318.1922 5695.7765 4675.7903 2531.4028 1266.0462
##            Aug       Sep       Oct       Nov       Dec
## 2024  743.8563  567.4583  485.9143  730.1993 1276.7176
```

```r
# as estimativas intervalares sao
ajuste1$lower # limite inferior
```

```
##                  80%        95%
## Jan 2024 1670.431122   965.4322
## Feb 2024 1716.257417   256.0425
## Mar 2024 1527.990560 -1007.7890
## Apr 2024  658.414788 -2008.2034
## May 2024   -7.411566 -2486.5489
## Jun 2024 -286.284653 -1777.8783
## Jul 2024 -279.788911 -1098.1046
## Aug 2024 -242.999325  -765.4092
## Sep 2024 -244.650589  -674.5550
## Oct 2024 -260.027341  -654.9050
## Nov 2024 -466.791671 -1100.4404
## Dec 2024 -949.929043 -2128.6446
```

```r
ajuste1$upper # limite superior
```

```
##                80%       95%
## Jan 2024  4333.981  5038.979
## Feb 2024  7233.081  8693.295
## Mar 2024 11108.394 13644.173
## Apr 2024 10733.138 13399.756
## May 2024  9358.992 11838.129
## Jun 2024  5349.090  6840.684
## Jul 2024  2811.881  3630.197
## Aug 2024  1730.712  2253.122
## Sep 2024  1379.567  1809.472
## Oct 2024  1231.856  1626.734
## Nov 2024  1927.190  2560.839
## Dec 2024  3503.364  4682.080
```