

Prova Prática

Beatriz Lima Silveira

```
library(ggplot2)
library(tseries)
library(readODS)
library(tidyverse)
library(readxl)
library(forecast)
```

Suavização com Médias Móveis

```
moving_averages <- function(x, order, centre = TRUE) {

  if (centre == TRUE) {
    # Média Móvel centrada
    return(stats::filter(x, rep(1/order, order), sides = 2))
  } else {
    # Média Móvel não centrada
    return(stats::filter(x, rep(1/order, order), sides = 1))
  }
}
```

A função `ma()` calcula a média móvel de uma série temporal, ajudando a suavizar flutuações e identificar tendências. Se `centre = TRUE`, ela aplica um filtro simétrico, considerando valores antes e depois de cada ponto. Caso contrário, usa apenas os valores anteriores. O argumento `order` define o tamanho da janela da média, e precisa ser ímpar se a centralização for ativada.

Suavização com Holt-Winters

```
holt_winters <- function(y, sazonal = c("aditivo", "multiplicativo"),
                        alpha = NULL, beta = NULL, gamma = NULL,
                        phi = NULL, lambda = NULL, ajuste_sesgo = FALSE,
                        x = y, ...) {

  sazonal <- match.arg(sazonal)
  m <- frequency(x)

  if (m <= 1L) {
    stop("A série temporal deve ter uma frequência maior que 1.")
  }

  if (length(y) < m + 3) {
    stop(paste("São necessárias pelo menos", m + 3, "observações para estimar a sazonalidade"))
  }

  if (sazonal == "aditivo") {
    modelo <- ets(x, "AAA", alpha = alpha, beta = beta,
                  gamma = gamma, phi = phi, opt.crit = "mse",
                  lambda = lambda, biasadj = ajuste_sesgo)
  } else {
    modelo <- ets(x, "MAM", alpha = alpha, beta = beta,
                  gamma = gamma, phi = phi, opt.crit = "mse",
                  lambda = lambda, biasadj = ajuste_sesgo)
  }

  return(modelo)
}
```

A função verifica se a série tem dados suficientes e se a frequência é válida. Após isso, ela usa o modelo de suavização exponencial para estimar a série suavizada, retornando o modelo ajustado.

Comparando suavizações

```
# Cálculo das métricas de erro
mae <- function(real, suavizado) mean(abs(real - suavizado), na.rm = TRUE)
mse <- function(real, suavizado) mean((real - suavizado)^2, na.rm = TRUE)
mape <- function(real, suavizado) mean(abs((real - suavizado) / real), na.rm = TRUE) * 100
```

As métricas MAE, MSE e MAPE são usadas para avaliar a qualidade de modelos de suavização, cada uma de maneira diferente. O MAE calcula o erro médio absoluto e fornece uma medida simples da precisão do modelo, sem penalizar excessivamente erros grandes. O MSE dá mais peso aos erros maiores, tornando-o útil quando se deseja controlar grandes discrepâncias. O MAPE avalia o erro relativo em porcentagem, sendo útil para comparar modelos em diferentes escalas.

A série temperatura em Ubatuba:

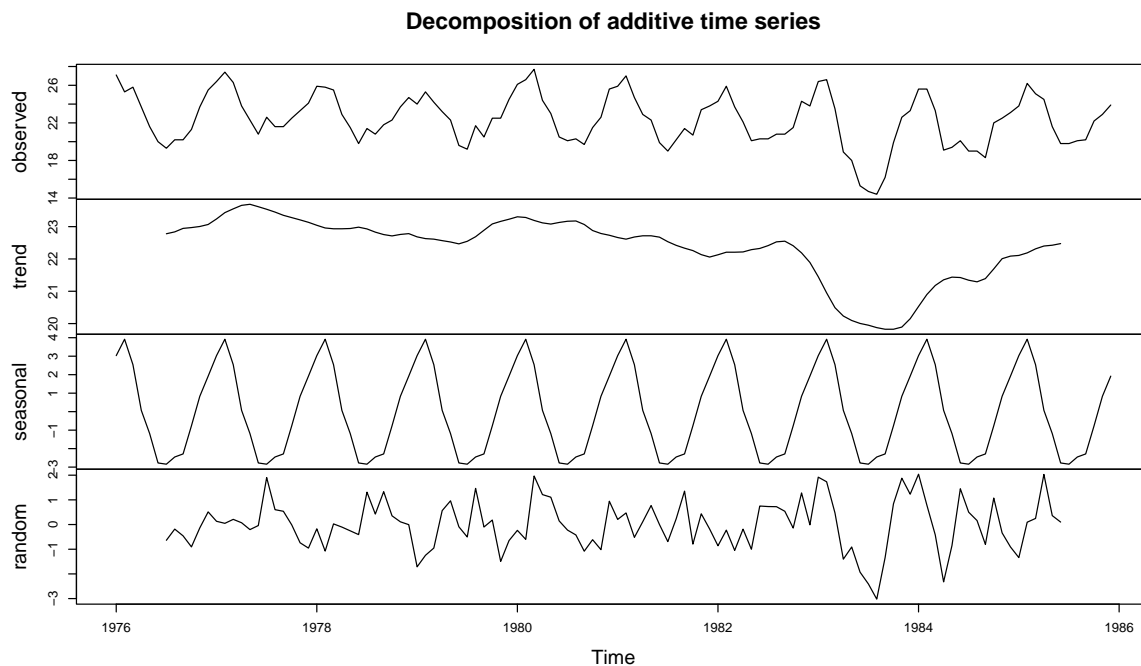
```
data <- read_ods("data/temperatura.ods")

# transformando em formato time series

data_ts <- ts(data$Ubatuba, start = c(1976, 1), frequency = 12)

decompose <- decompose(data_ts)

plot(decompose)
```



```
data_fit <- data %>% select(-c(Cananeia))

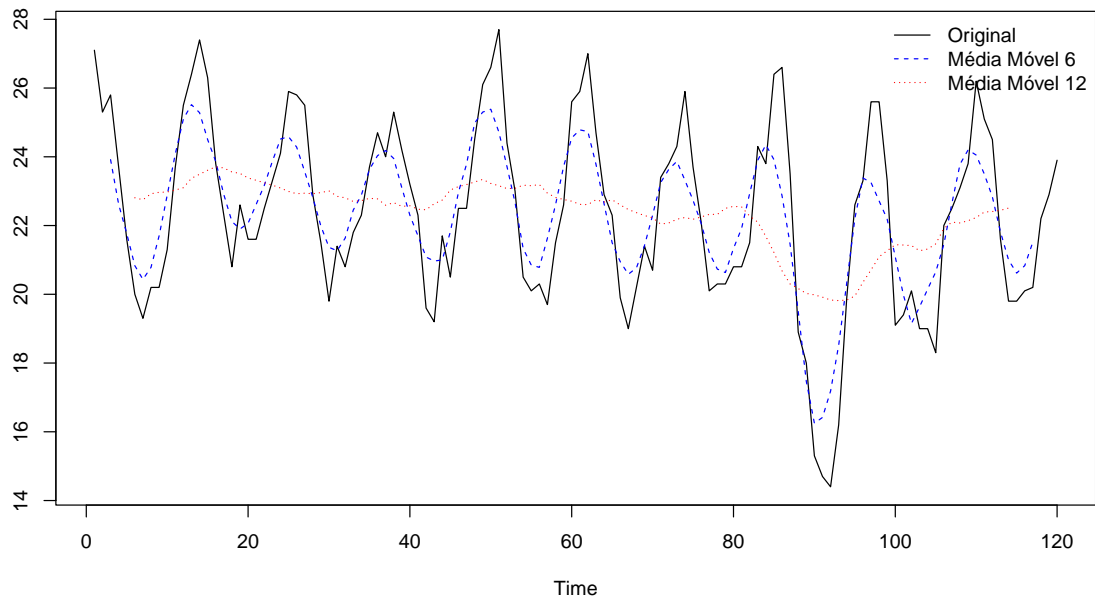
# suavização por médias móveis

ma_6 <- moving_averages(data_fit$Ubatuba, order = 6)
ma_12 <- moving_averages(data_fit$Ubatuba, order = 12)

df_ma <- cbind(data_fit$Ubatuba,
               ma_6,
               ma_12)

ts.plot(df_ma, col = c("black", "blue", "red"), lty = 1:3,
        main = "Temperatura em Ubatuba e Médias Móveis")
legend("topright",
      legend = c("Original",
                  "Média Móvel 6",
                  "Média Móvel 12"),
      col = c("black", "blue", "red"),
      lty = 1:3, bty = "n")
```

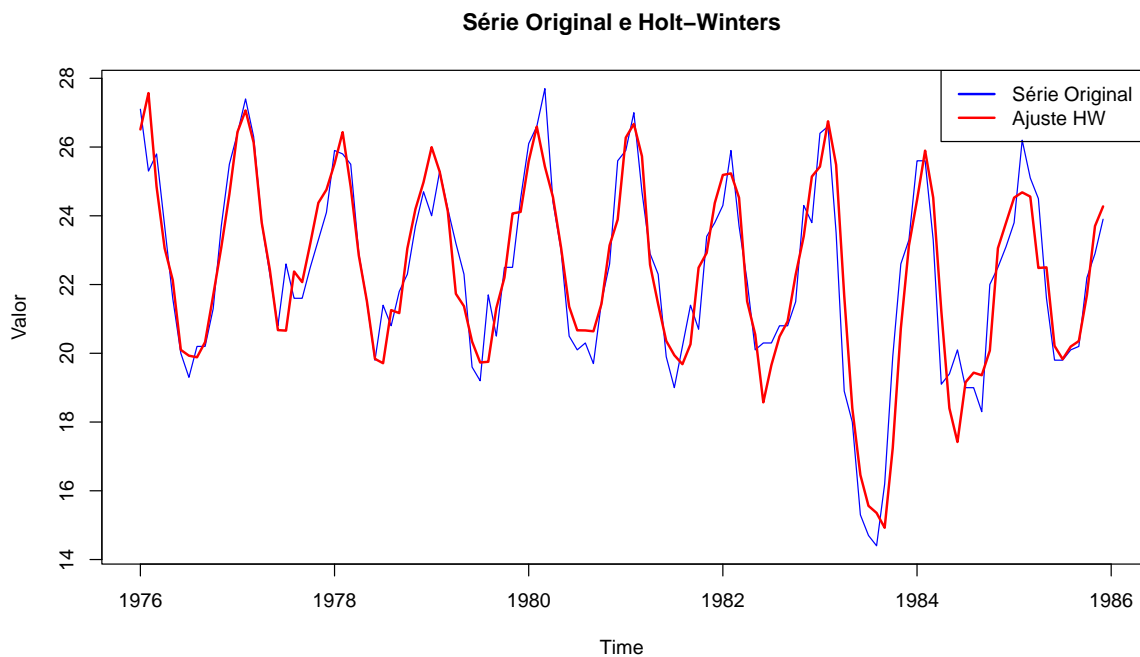
Temperatura em Ubatuba e Médias Móveis



```
# ajustar o modelo de holt-winters

modelo_hw <- holt_winters(data_ts, seasonal = "additive")

plot(data_ts, type = "l", col = "blue",
      main = "Série Original e Holt-Winters",
      ylab = "Valor")
lines(modelo_hw$fitted, col = "red",
      lwd = 2)
legend("topright",
      legend = c("Série Original", "Ajuste HW"),
      col = c("blue", "red"), lwd = 2)
```



```
# Comparando suavizações

# Criando um dataframe para comparação
df_comparacao <- data.frame(
  Data = time(data_ts),
  Original = as.numeric(data_ts),
  HoltWinters = as.numeric(modelo_hw$fitted),
  MediaMovel6 = as.numeric(ma_6),
  MediaMovel12 = as.numeric(ma_12)
)

resultados_erro <- data.frame(
  Metodo = c("Holt-Winters", "Média Móvel 6", "Média Móvel 12"),
  MAE = c(mae(df_comparacao$Original, df_comparacao$HoltWinters),
    mae(df_comparacao$Original, df_comparacao$MediaMovel6),
    mae(df_comparacao$Original, df_comparacao$MediaMovel12)),
  MSE = c(mse(df_comparacao$Original, df_comparacao$HoltWinters),
    mse(df_comparacao$Original, df_comparacao$MediaMovel6),
    mse(df_comparacao$Original, df_comparacao$MediaMovel12)),
  MAPE = c(mape(df_comparacao$Original, df_comparacao$HoltWinters),
```

```

        mape(df_comparacao$Original, df_comparacao$MediaMovel6),
        mape(df_comparacao$Original, df_comparacao$MediaMovel12))
    )

print(resultados_erro)

```

	Metodo	MAE	MSE	MAPE
1	Holt-Winters	0.7837703	1.027558	3.588200
2	Média Móvel 6	1.0340580	1.609114	4.706469
3	Média Móvel 12	2.1440367	6.339873	9.868902

A série temperatura em Cananeia:

```

data <- read_ods("data/temperatura.ods")

# transformando em formato time series

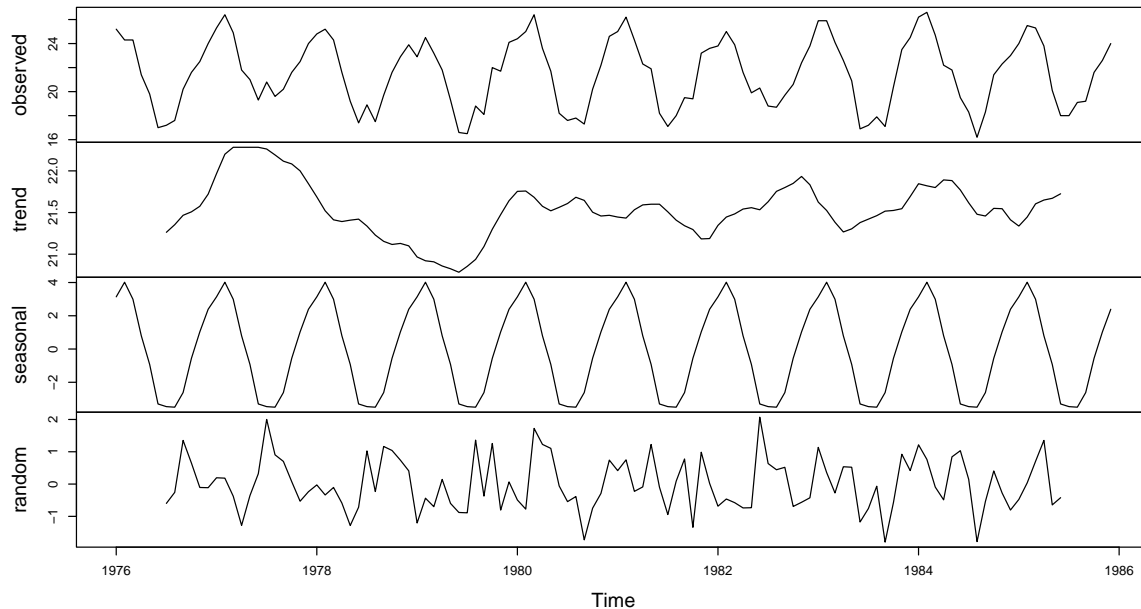
data_ts <- ts(data$Cananeia, start = c(1976, 1), frequency = 12)

decompose <- decompose(data_ts)

plot(decompose)

```

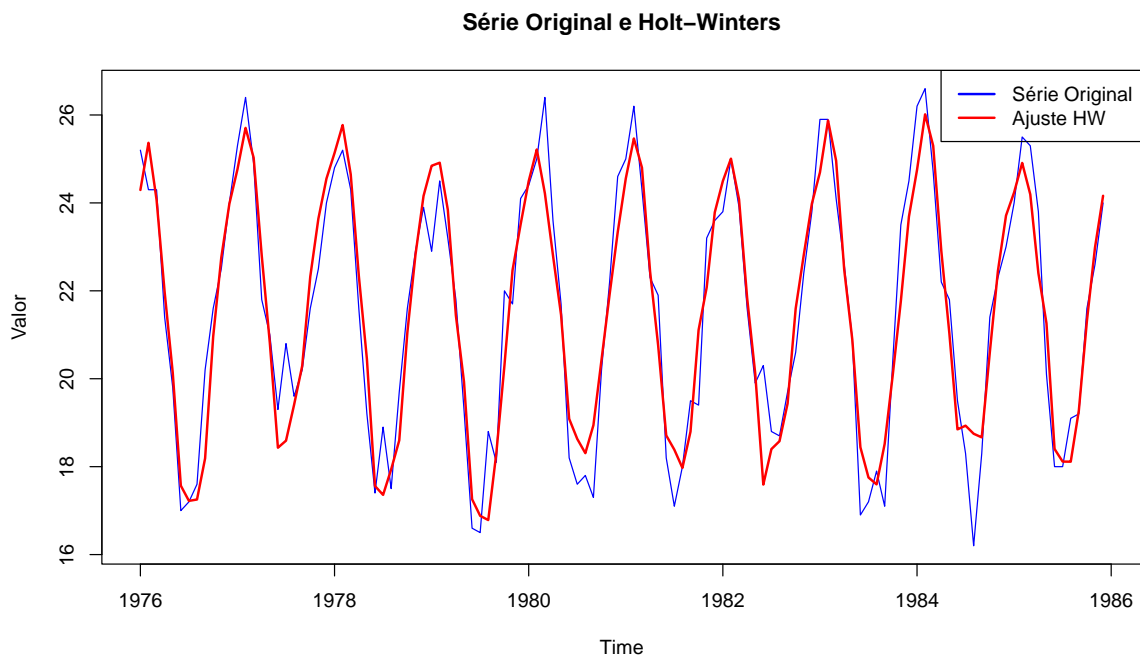
Decomposition of additive time series



```
data_fit <- data %>% select(-c(Ubatuba))

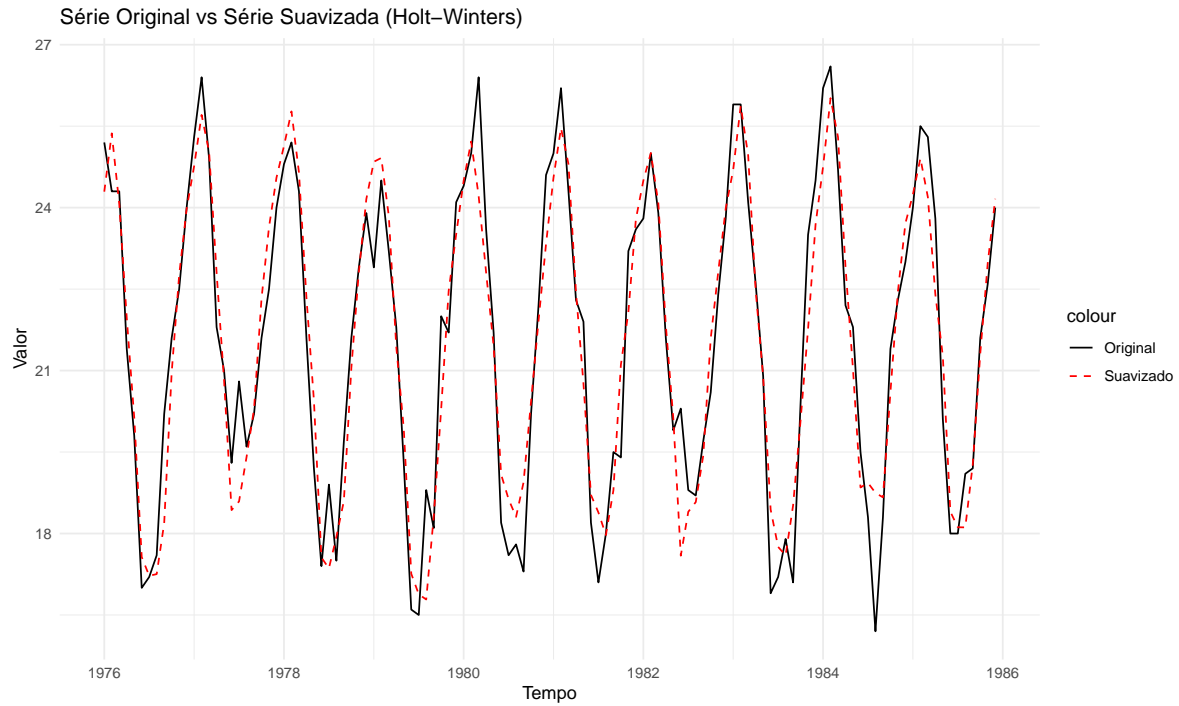
# ajustar o modelo de holt-winters
modelo_hw <- holt_winters(data_ts, seasonal = "additive")

plot(data_ts, type = "l", col = "blue",
      main = "Série Original e Holt-Winters",
      ylab = "Valor")
lines(modelo_hw$fitted, col = "red",
      lwd = 2)
legend("topright", legend = c("Série Original", "Ajuste HW"), col = c("blue", "red"), lwd
```

```
# criando um dataframe para visualização
df_plot <- data.frame(
  Data = time(data_ts),
  Original = as.numeric(data_ts),
  Suavizado = as.numeric(modelo_hw$fitted)
)

# plotando com ggplot2
ggplot(df_plot, aes(x = Data)) +
  geom_line(aes(y = Original, color = "Original")) +
  geom_line(aes(y = Suavizado, color = "Suavizado"),
            linetype = "dashed") +
  labs(title = "Série Original vs Série Suavizada (Holt-Winters)",
        x = "Tempo", y = "Valor") +
  scale_color_manual(values = c("Original" = "black",
                                "Suavizado" = "red")) +
  theme_minimal()
```



```
# suavização por médias móveis
```

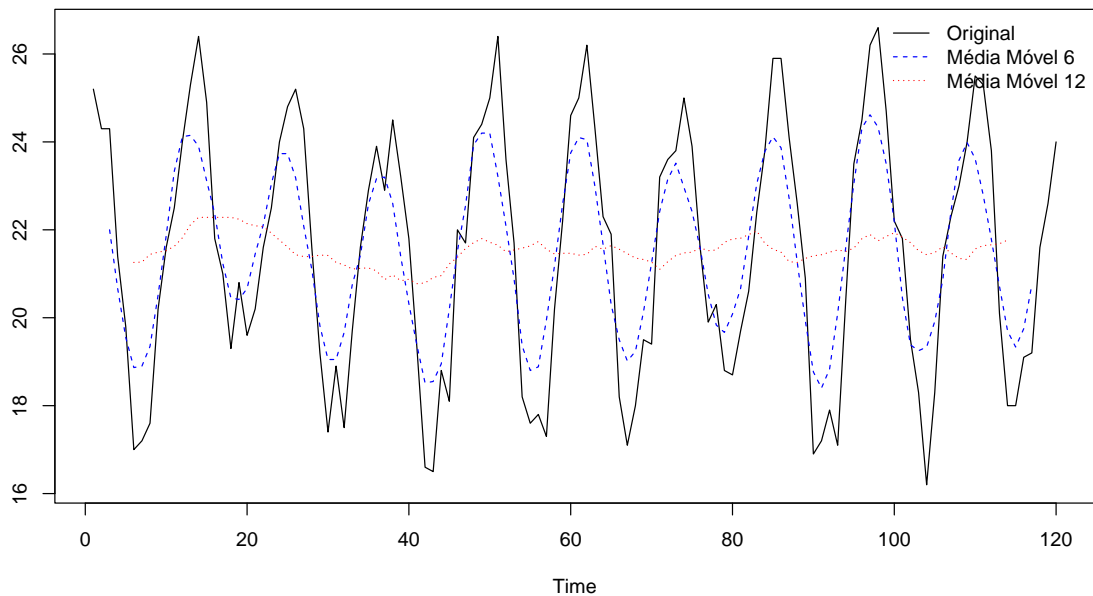
```
ma_6 <- moving_averages(data_fit$Cananeia, order = 6)
```

```
ma_12 <- moving_averages(data_fit$Cananeia, order = 12)
```

```
df_ma <- cbind(data_fit$Cananeia,
               ma_6,
               ma_12)
```

```
ts.plot(df_ma, col = c("black", "blue", "red"),
        lty = 1:3,
        main = "Temperatura em Cananeia e Médias Móveis")
legend("topright",
      legend = c("Original", "Média Móvel 6", "Média Móvel 12"),
      col = c("black", "blue", "red"),
      lty = 1:3, bty = "n")
```

Temperatura em Cananeia e Médias Móveis



```
# comparando suavizações

# Criando um dataframe para comparação
df_comparacao <- data.frame(
  Data = time(data_ts),
  Original = as.numeric(data_ts),
  HoltWinters = as.numeric(modelo_hw$fitted),
  MediaMovel6 = as.numeric(ma_6),
  MediaMovel12 = as.numeric(ma_12)
)

resultados_erro <- data.frame(
  Metodo = c("Holt-Winters", "Média Móvel 6", "Média Móvel 12"),
  MAE = c(mae(df_comparacao$Original, df_comparacao$HoltWinters),
          mae(df_comparacao$Original, df_comparacao$MediaMovel6),
          mae(df_comparacao$Original, df_comparacao$MediaMovel12)),
  MSE = c(mse(df_comparacao$Original, df_comparacao$HoltWinters),
          mse(df_comparacao$Original, df_comparacao$MediaMovel6),
          mse(df_comparacao$Original, df_comparacao$MediaMovel12)),
  MAPE = c(mape(df_comparacao$Original, df_comparacao$HoltWinters),
```

```

        mape(df_comparacao$Original, df_comparacao$MediaMovel6),
        mape(df_comparacao$Original, df_comparacao$MediaMovel12))
    )

print(resultados_erro)

```

	Metodo	MAE	MSE	MAPE
1	Holt-Winters	0.6926066	0.8111715	3.310496
2	Média Móvel 6	1.1237681	1.8314444	5.401060
3	Média Móvel 12	2.4500000	7.8858180	11.784360

A série de Consumo

```

data <- read_excel("data/CONSUMO.xls")

# transformando em formato time series

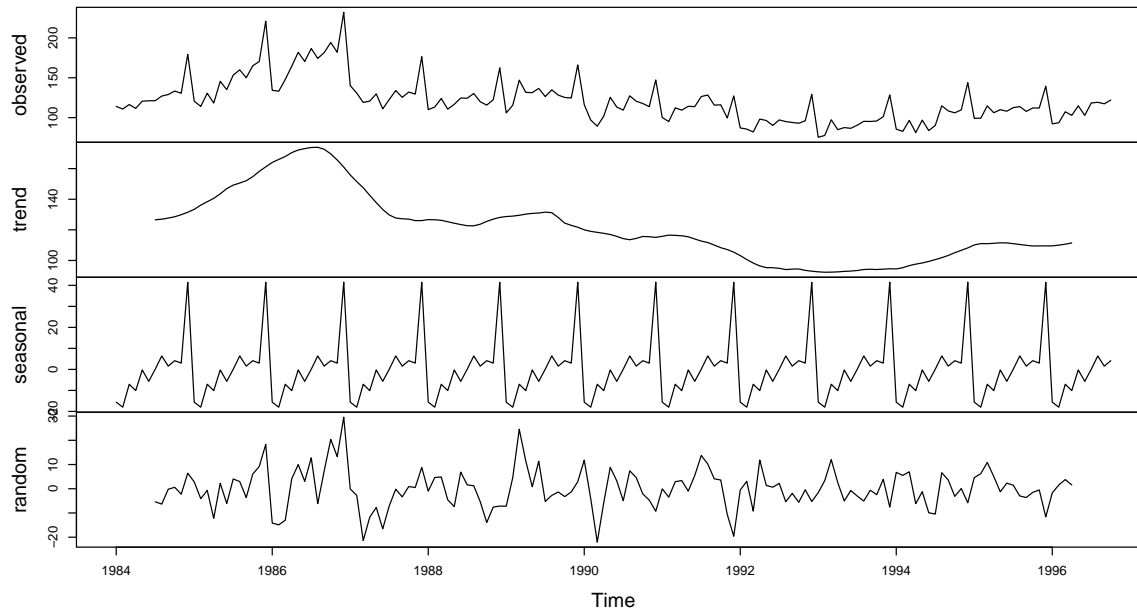
data_ts <- ts(data$consumo, start = c(1984,1), frequency = 12)

decompose <- decompose(data_ts)

plot(decompose)

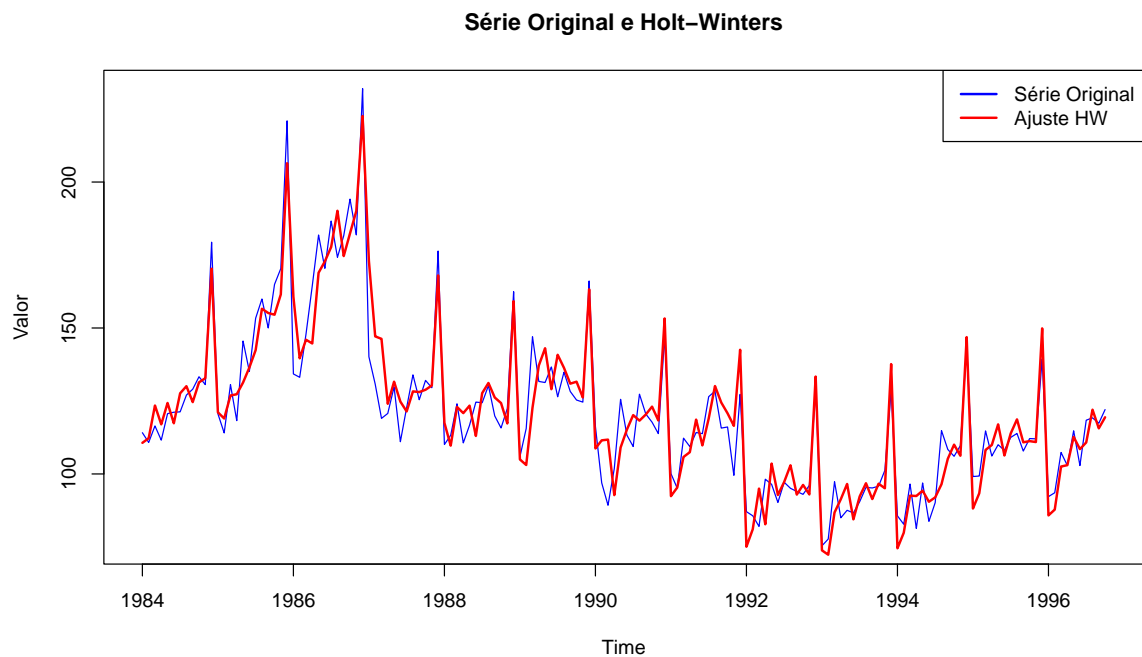
```

Decomposition of additive time series



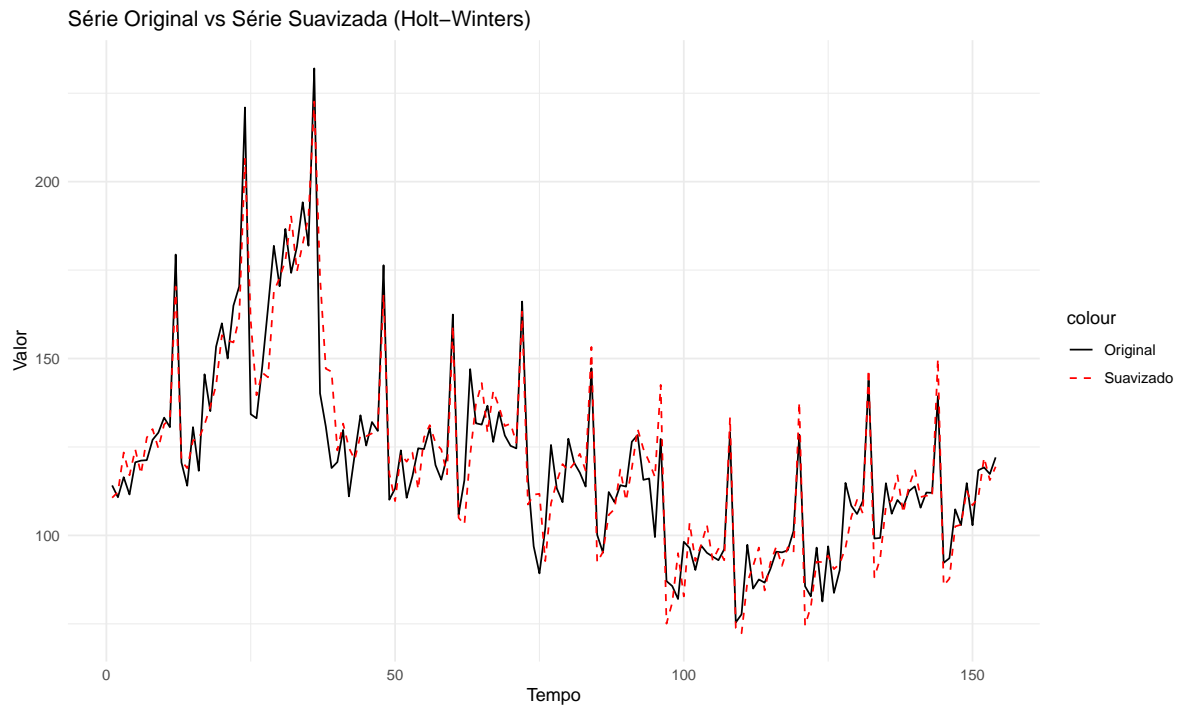
```
# ajustar o modelo de holt-winters
modelo_hw <- holt_winters(data_ts, seasonal = "additive")

plot(data_ts, type = "l", col = "blue",
      main = "Série Original e Holt-Winters", ylab = "Valor")
lines(modelo_hw$fitted, col = "red", lwd = 2)
legend("topright", legend = c("Série Original", "Ajuste HW"),
      col = c("blue", "red"), lwd = 2)
```



```
# criando um dataframe para visualização
df_plot <- data.frame(
  Data = time(data$data),
  Original = as.numeric(data$consumo),
  Suavizado = as.numeric(modelo_hw$fitted)
)

# plotando com ggplot2
ggplot(df_plot, aes(x = Data)) +
  geom_line(aes(y = Original, color = "Original")) +
  geom_line(aes(y = Suavizado, color = "Suavizado"), linetype = "dashed") +
  labs(title = "Série Original vs Série Suavizada (Holt-Winters)",
       x = "Tempo", y = "Valor") +
  scale_color_manual(values = c("Original" = "black", "Suavizado" = "red")) +
  theme_minimal()
```



```
# suavização por médias móveis
```

```
ma_6 <- moving_averages(data_ts, order = 6)
```

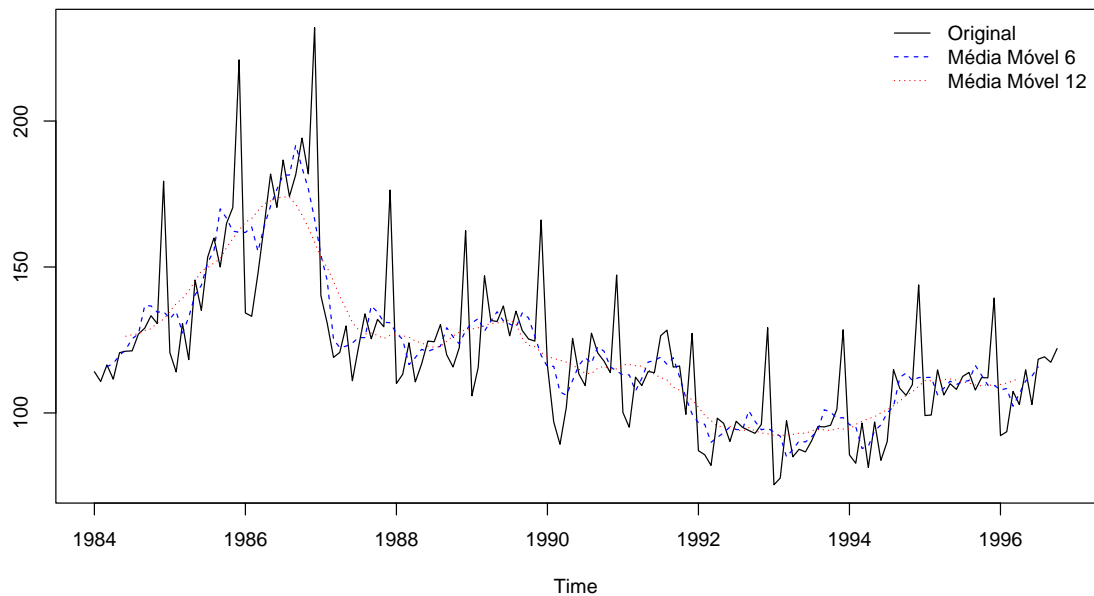
```
ma_12 <- moving_averages(data_ts, order = 12)
```

```
df_ma <- cbind(data$consumo,  
               ma_6,  
               ma_12)
```

```
ts.plot(df_ma, col = c("black", "blue", "red"), lty = 1:3,  
        main = "Temperatura em Cananeia e Médias Móveis")
```

```
legend("topright",  
      legend = c("Original", "Média Móvel 6", "Média Móvel 12"),  
      col = c("black", "blue", "red"), lty = 1:3, bty = "n")
```

Temperatura em Cananeia e Médias Móveis



```
# comparando suavizações

# Criando um dataframe para comparação
df_comparacao <- data.frame(
  Data = time(data$data),
  Original = as.numeric(data$consumo),
  HoltWinters = as.numeric(modelo_hw$fitted),
  MediaMovel6 = as.numeric(ma_6),
  MediaMovel12 = as.numeric(ma_12)
)

resultados_erro <- data.frame(
  Metodo = c("Holt-Winters", "Média Móvel 6", "Média Móvel 12"),
  MAE = c(mae(df_comparacao$Original, df_comparacao$HoltWinters),
    mae(df_comparacao$Original, df_comparacao$MediaMovel6),
    mae(df_comparacao$Original, df_comparacao$MediaMovel12)),
  MSE = c(mse(df_comparacao$Original, df_comparacao$HoltWinters),
    mse(df_comparacao$Original, df_comparacao$MediaMovel6),
    mse(df_comparacao$Original, df_comparacao$MediaMovel12)),
  MAPE = c(mape(df_comparacao$Original, df_comparacao$HoltWinters),
```



```

        mape(df_comparacao$Original, df_comparacao$MediaMovel6),
        mape(df_comparacao$Original, df_comparacao$MediaMovel12))
    )

print(resultados_erro)

```

	Metodo	MAE	MSE	MAPE
1	Holt-Winters	6.730536	78.02996	5.652430
2	Média Móvel 6	9.955951	221.86385	7.980290
3	Média Móvel 12	11.331917	278.56541	9.086068