

# Séries Temporais: Trabalho 1

Mariana Costa Freitas

2025-01-31

Geralmente, nas séries temporais há alguma variação aleatória ao longo do tempo. Um dos métodos para a redução do efeito da variação aleatória é a “suavização”, facilitando a identificação de elementos como a sazonalidade e tendência.

## IBOVESPA

### Suavização por médias móveis

A suavização por médias móveis consiste em calcular a média dos valores da série em um intervalo de tamanho  $k$  que pode ser deslocado. Temos a fórmula para a média móvel simples:

$$\text{Média móvel}_t = \frac{1}{k} \sum_{i=0}^{k-1} Y_{t-i}$$

Primeiro vamos carregar os dados dos retornos diários do IBOVESPA, armazenando seus valores no vetor `retornos`.

```
library(dplyr)
library(janitor)
library(TTR)
library(ggplot2)
library(forecast)

# Dados do IBOVESPA
url <- "https://www.ime.usp.br/~pam/M-IBV-SP"

# Ler o arquivo usando read.table()
dados <- read.table(url, header = TRUE) |>
  clean_names() |>
  select(ibov)

retornos <- ts(dados$ibov)

# Função para Média Móvel Manual
media_movel_manual <- function(serie, k) {
  n <- length(serie)
  ma <- numeric(n) # Vetor vazio para armazenar resultados

  for (i in k:n) {
```

```

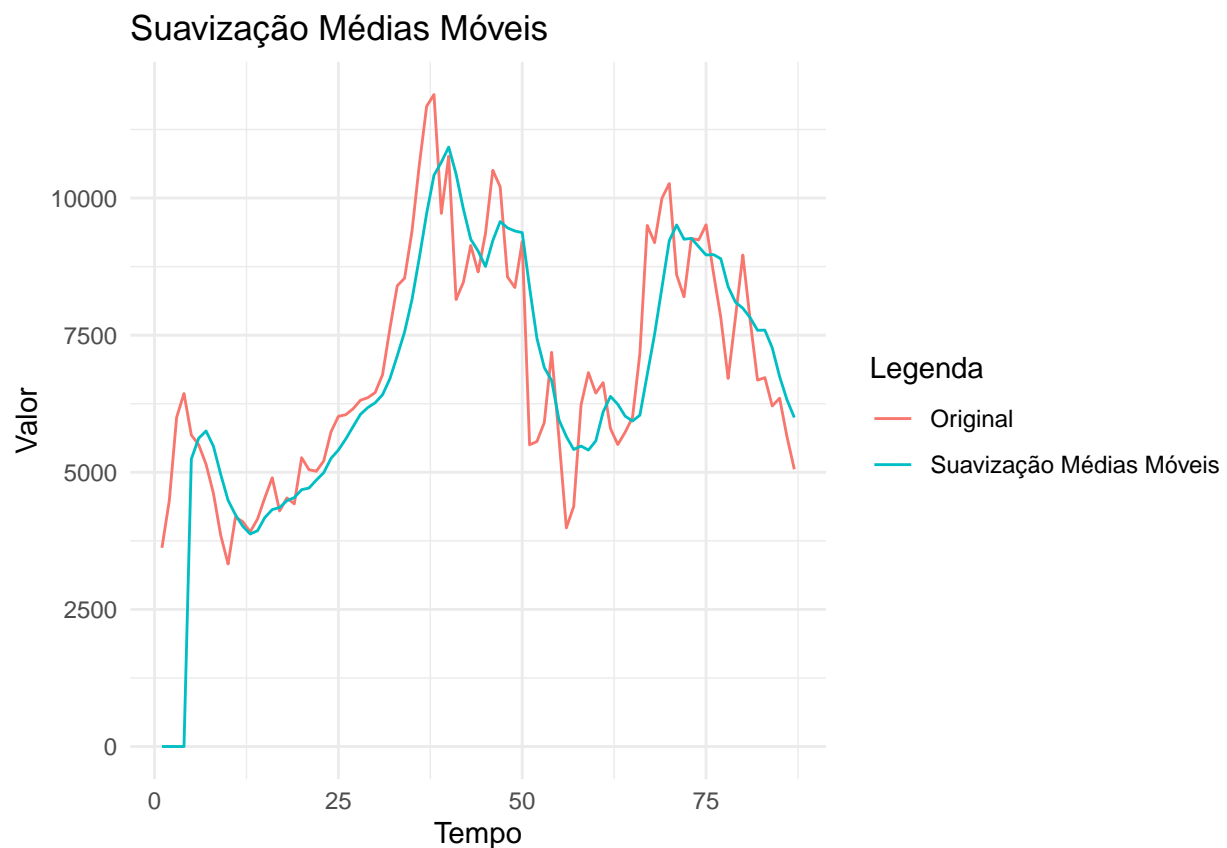
    ma[i] <- mean(serie[(i-k+1):i]) # Média dos últimos k valores
  }
  return(ma)
}

# Aplicar média móvel com janela de 5
ma_5_manual <- media_movel_manual(retornos, k = 5)

# Visualizar os dados originais e suavizados
# Converter para data frame
df <- data.frame(
  Tempo = 1:length(retornos),
  Original = as.numeric(retornos),
  Suavizado = as.numeric(ma_5_manual)
)

# Plot usando ggplot2
ggplot(df, aes(x = Tempo)) +
  geom_line(aes(y = Original, color = "Original")) +
  geom_line(aes(y = Suavizado, color = "Suavização Médias Móveis")) +
  labs(title = "Suavização Médias Móveis",
        x = "Tempo",
        y = "Valor",
        color = "Legenda") +
  theme_minimal()

```



## Suavização exponencial

Diferente da média móvel, a suavização exponencial dá pesos maiores para valores mais recentes. O peso é controlado pelo fator de suavização  $0 < \alpha \leq 1$ .

$$S_t = \alpha Y_t + (1 - \alpha)S_{t-1}$$

```
# Função para Suavização Exponencial Simples Manual
suav_exp_manual <- function(serie, alpha) {
  n <- length(serie)
  ses <- numeric(n) # Vetor vazio para armazenar resultados

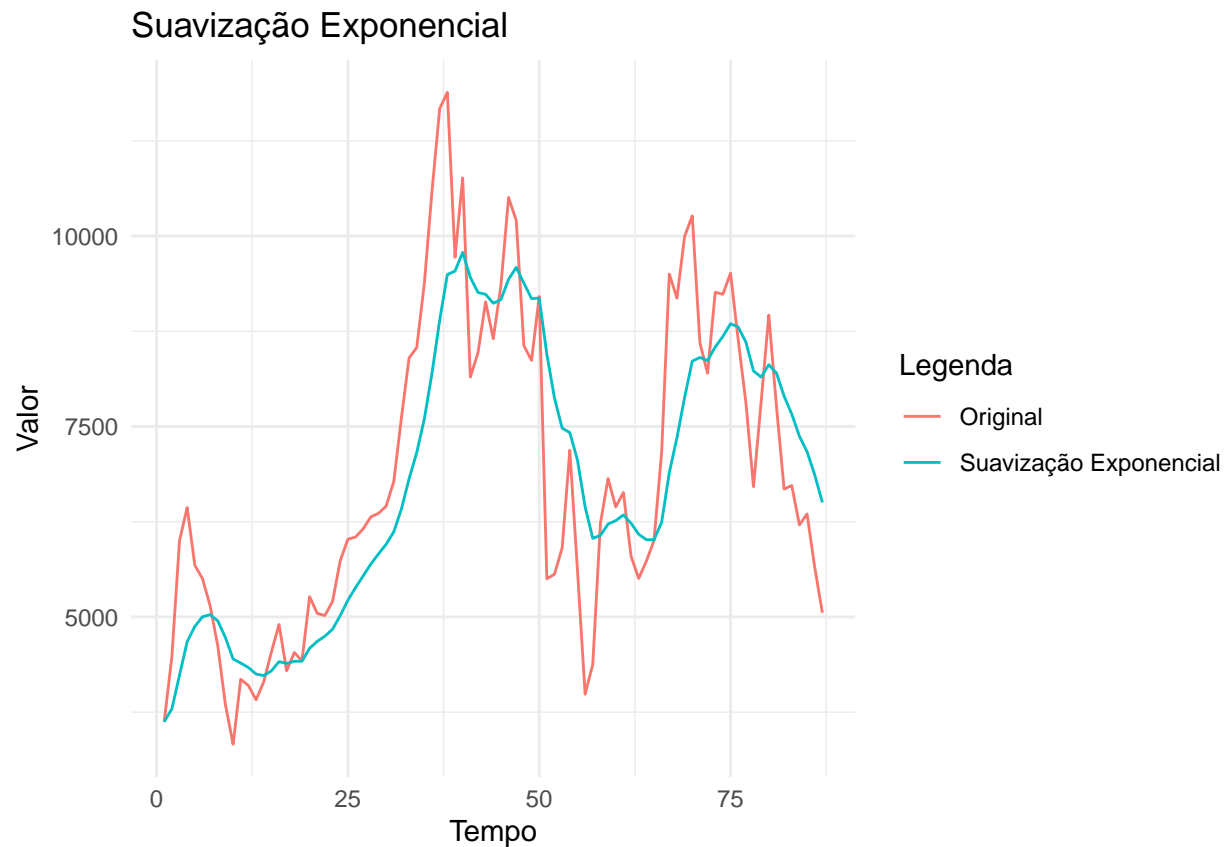
  # Inicializando o primeiro valor
  ses[1] <- serie[1]

  # Aplicando a fórmula iterativamente
  for (i in 2:n) {
    ses[i] <- alpha * serie[i] + (1 - alpha) * ses[i - 1]
  }
  return(ses)
}

# Aplicar suavização exponencial com alpha = 0.2
ses_manual <- suav_exp_manual(retornos, alpha = 0.2)

# Visualizar os dados originais e suavizados
# Converter para data frame
df <- data.frame(
  Tempo = 1:length(retornos),
  Original = as.numeric(retornos),
  Suavizado = as.numeric(ses_manual)
)

# Plot usando ggplot2
ggplot(df, aes(x = Tempo)) +
  geom_line(aes(y = Original, color = "Original")) +
  geom_line(aes(y = Suavizado, color = "Suavização Exponencial")) +
  labs(title = "Suavização Exponencial",
       x = "Tempo",
       y = "Valor",
       color = "Legenda") +
  theme_minimal()
```



## Série temporal de Umidade

Carregando os dados de umidade a serem utilizados:

```
library(readxl)

atmosfera <- read_excel("atmosfera.xls",
                        sheet = "Plan1") |>
  select(-c(`18.399999999999999`))

serie <- atmosfera$umidade
```

## Suavização por médias móveis

```
# Aplicar média móvel com janela de 7
ma_3_manual <- media_movel_manual(serie, k = 3)

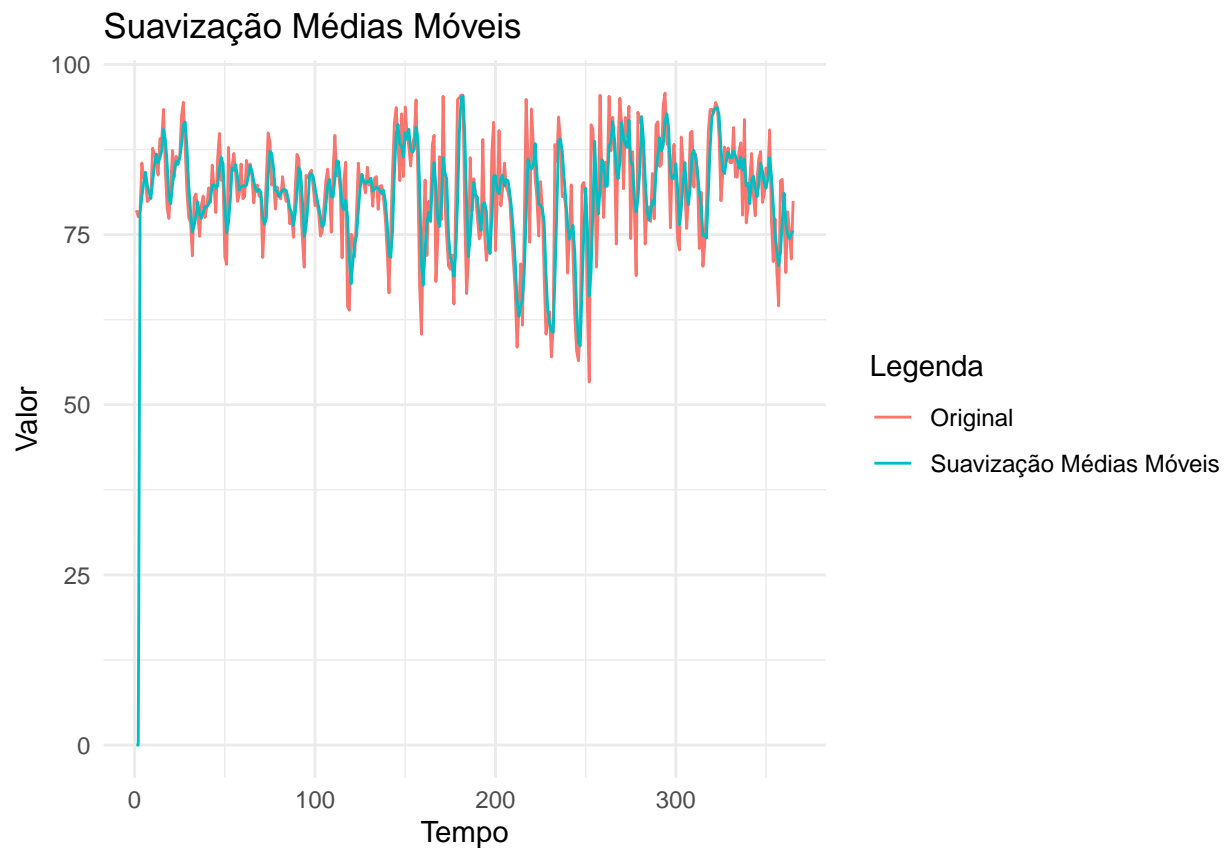
# Visualizar os dados originais e suavizados
# Converter para data frame
df <- data.frame(
  Tempo = 1:length(serie),
  Original = as.numeric(serie),
```

```

Suavizado = as.numeric(ma_3_manual)
)

# Plot usando ggplot2
ggplot(df, aes(x = Tempo)) +
  geom_line(aes(y = Original, color = "Original")) +
  geom_line(aes(y = Suavizado, color = "Suavização Médias Móveis")) +
  labs(title = "Suavização Médias Móveis",
       x = "Tempo",
       y = "Valor",
       color = "Legenda") +
  theme_minimal()

```



## Suavização Exponencial

```

# Aplicar suavização exponencial com alpha = 0.2
ses_manual <- suav_exp_manual(serie, alpha = 0.2)

# Visualizar os dados originais e suavizados
# Converter para data frame
df <- data.frame(
  Tempo = 1:length(serie),
  Original = as.numeric(serie),

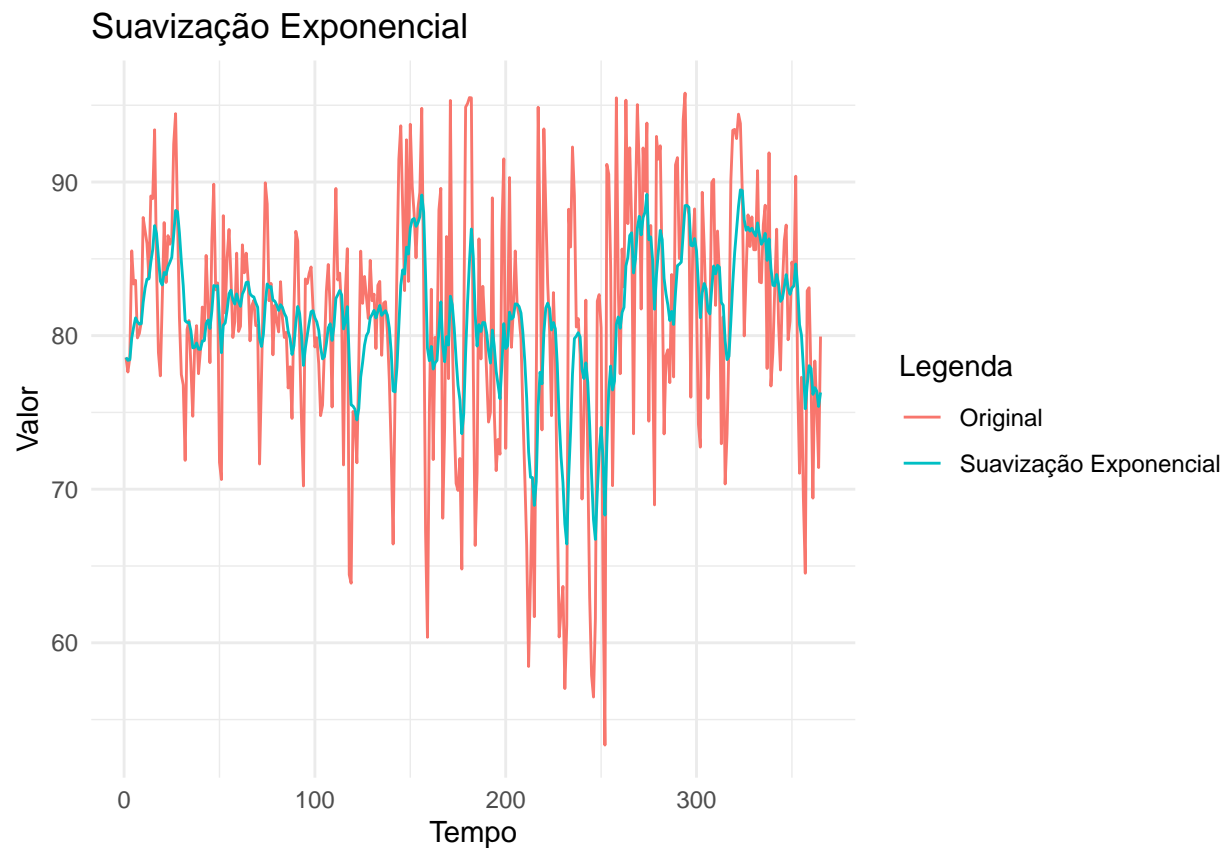
```

```

    Suavizado = as.numeric(ses_manual)
  )

  # Plot usando ggplot2
  ggplot(df, aes(x = Tempo)) +
    geom_line(aes(y = Original, color = "Original")) +
    geom_line(aes(y = Suavizado, color = "Suavização Exponencial")) +
    labs(title = "Suavização Exponencial",
         x = "Tempo",
         y = "Valor",
         color = "Legenda") +
    theme_minimal()

```



## Série temporal de Temperatura

```

library(readxl)
temperatura <- read_excel("temperatura.xls") |>
  clean_names() |>
  select(ubatuba)

serie <- ts(temperatura, frequency = 12, start = c(1976, 1)) # Dados mensais começando em janeiro de 1976

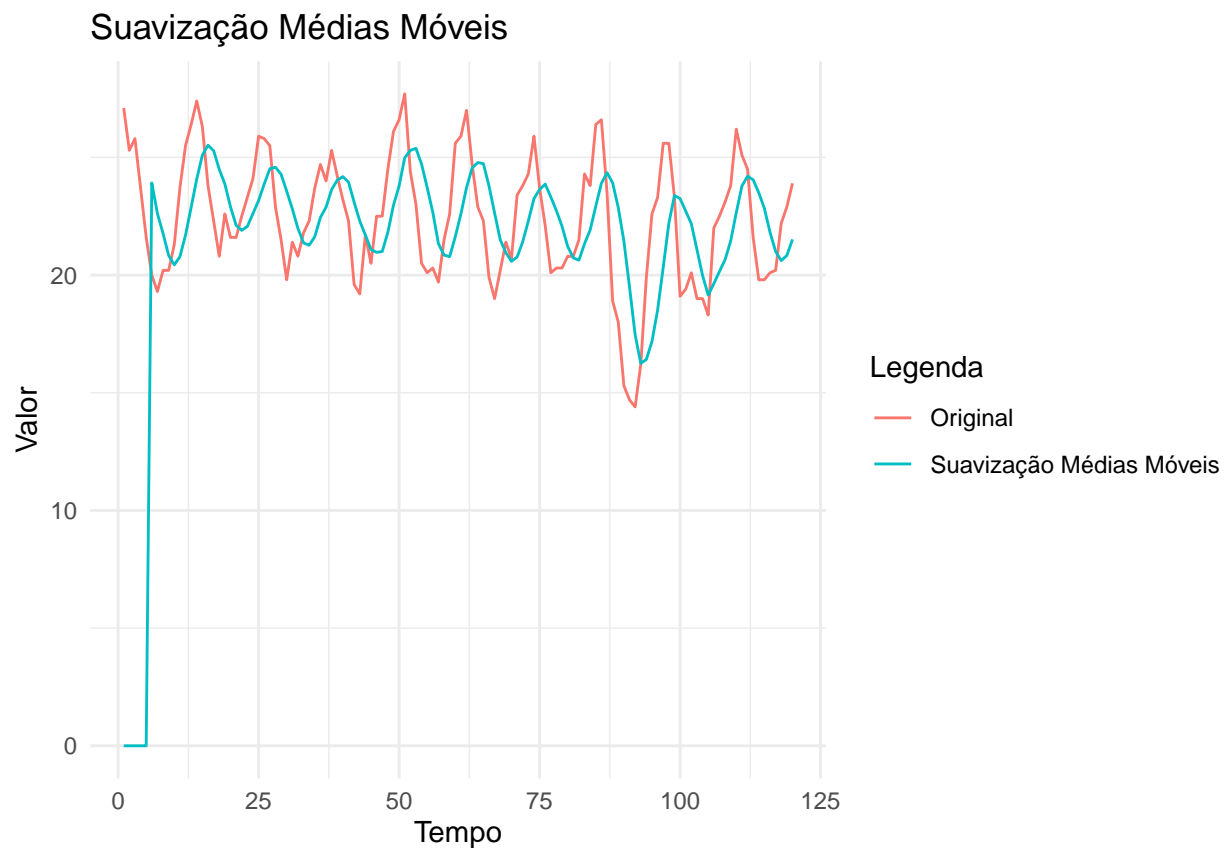
```

## Suavização por médias móveis

```
# Aplicar média móvel com janela de 7
ma_6_manual <- media_movel_manual(serie, k = 6)

# Visualizar os dados originais e suavizados
# Converter para data frame
df <- data.frame(
  Tempo = 1:length(serie),
  Original = as.numeric(serie),
  Suavizado = as.numeric(ma_6_manual)
)

# Plot usando ggplot2
ggplot(df, aes(x = Tempo)) +
  geom_line(aes(y = Original, color = "Original")) +
  geom_line(aes(y = Suavizado, color = "Suavização Médias Móveis")) +
  labs(title = "Suavização Médias Móveis",
       x = "Tempo",
       y = "Valor",
       color = "Legenda") +
  theme_minimal()
```



## Suavização Exponencial

```
# Aplicar suavização exponencial com alpha = 0.2
ses_manual <- suav_exp_manual(serie, alpha = 0.2)

# Visualizar os dados originais e suavizados
# Converter para data frame
df <- data.frame(
  Tempo = 1:length(serie),
  Original = as.numeric(serie),
  Suavizado = as.numeric(ses_manual)
)

# Plot usando ggplot2
ggplot(df, aes(x = Tempo)) +
  geom_line(aes(y = Original, color = "Original")) +
  geom_line(aes(y = Suavizado, color = "Suavização Exponencial")) +
  labs(title = "Suavização Exponencial",
       x = "Tempo",
       y = "Valor",
       color = "Legenda") +
  theme_minimal()
```

