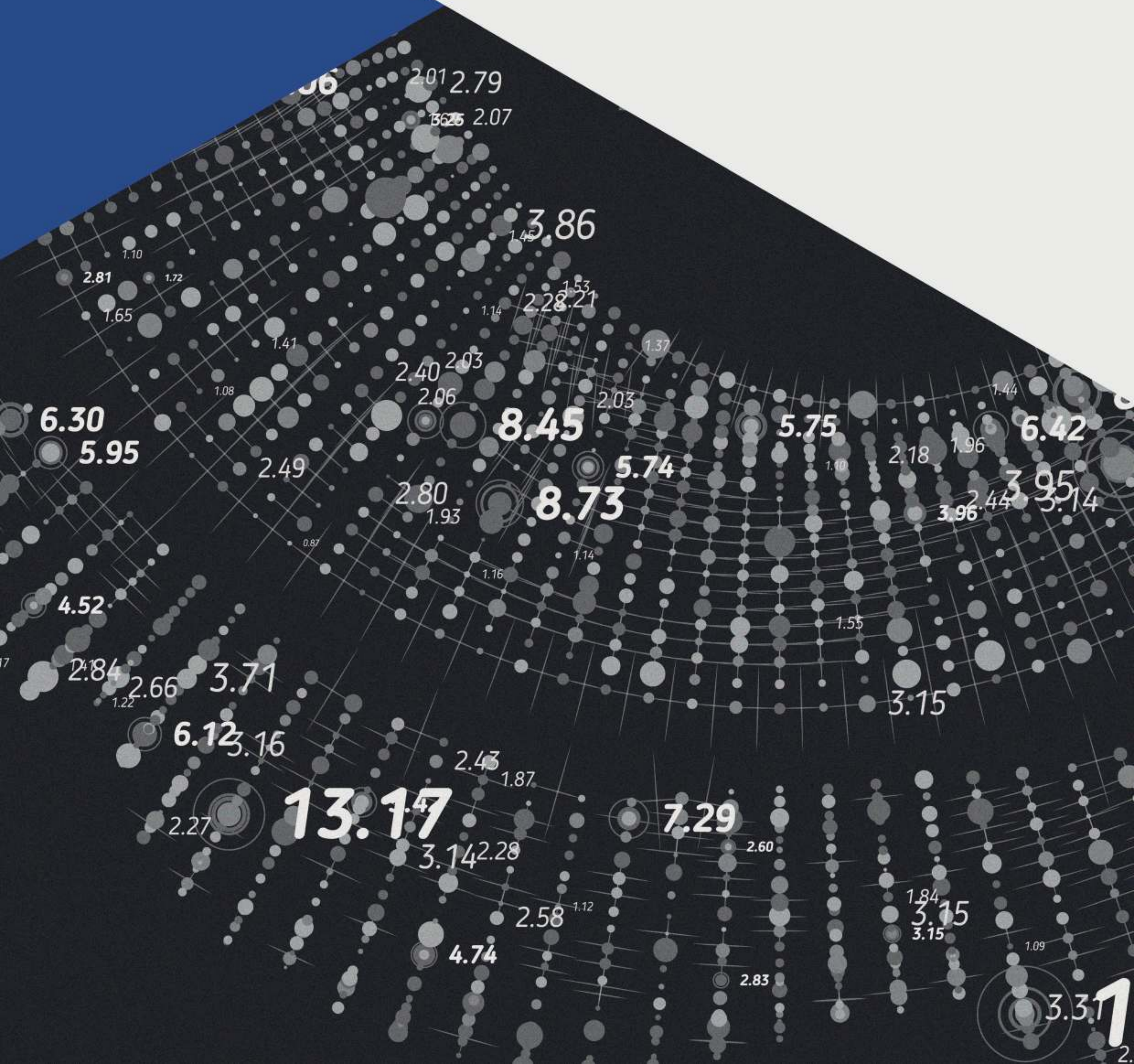
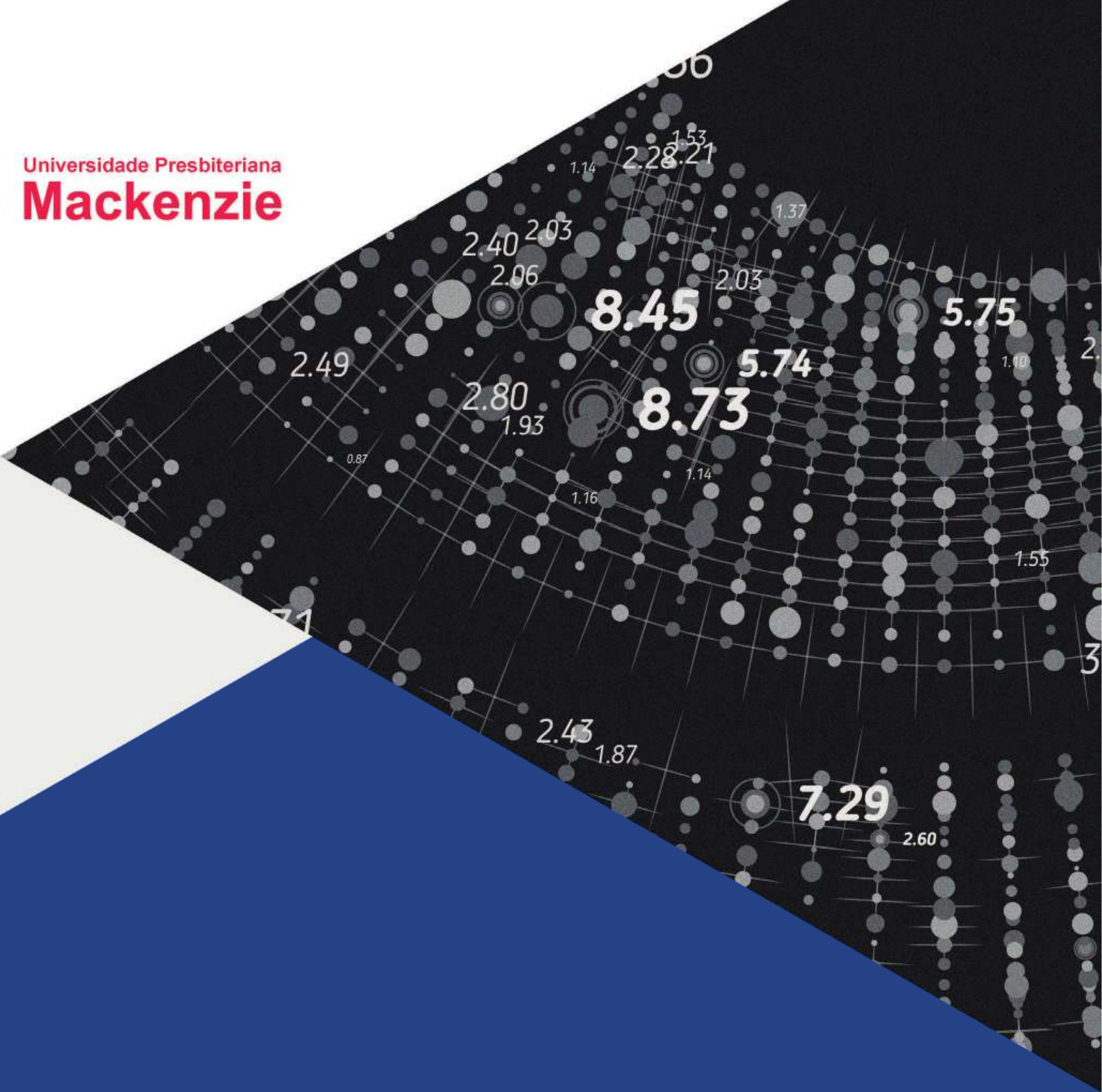
Universidade Presbiteriana
Mackenzie

CIÊNCIAS DE DADOS (BIG DATA ANALYTICS)

BIG DATA Analytics: Análise Estatística



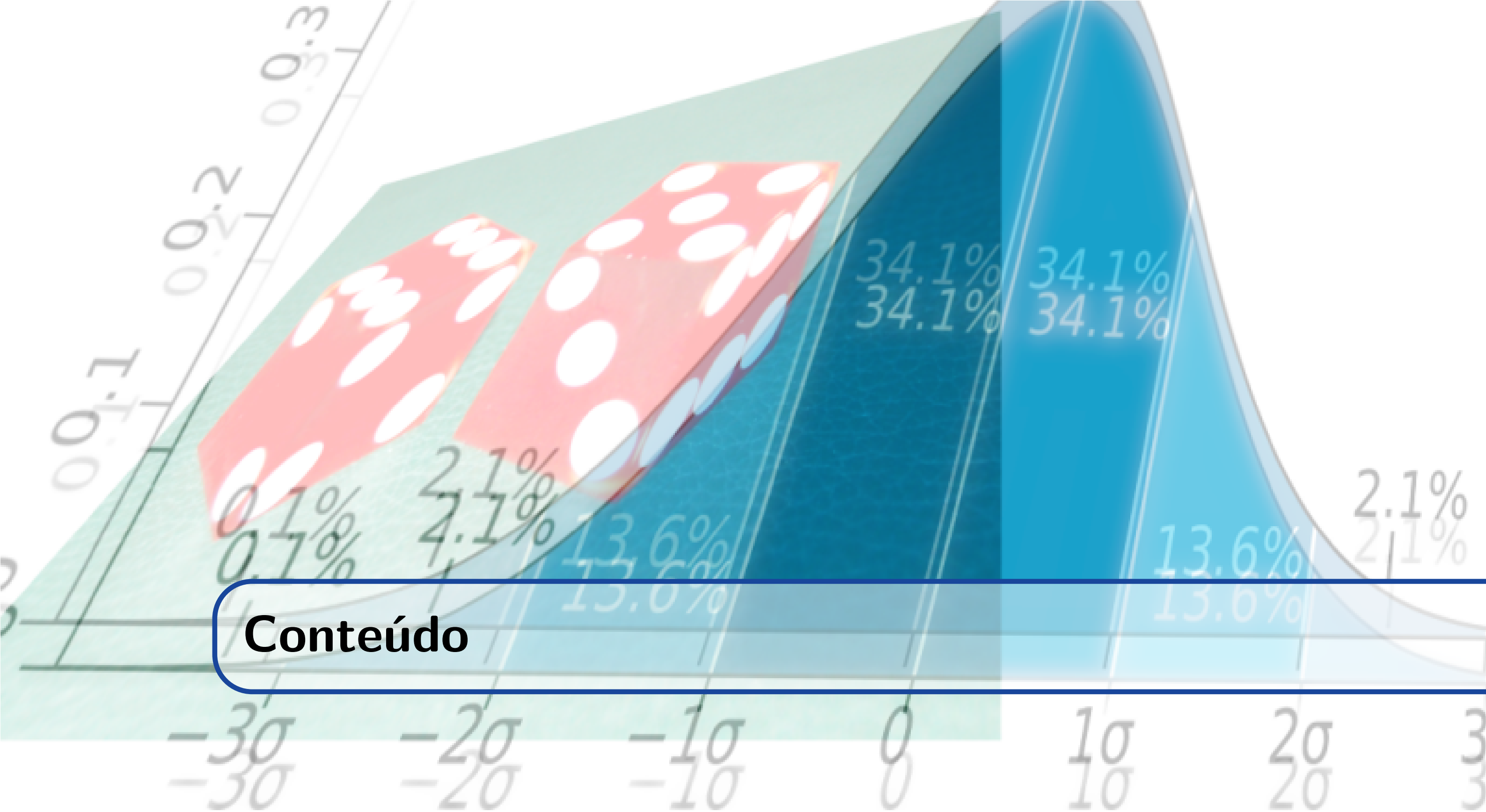
Universidade Presbiteriana Mackenzie



Trilha de Aprendizagem 07

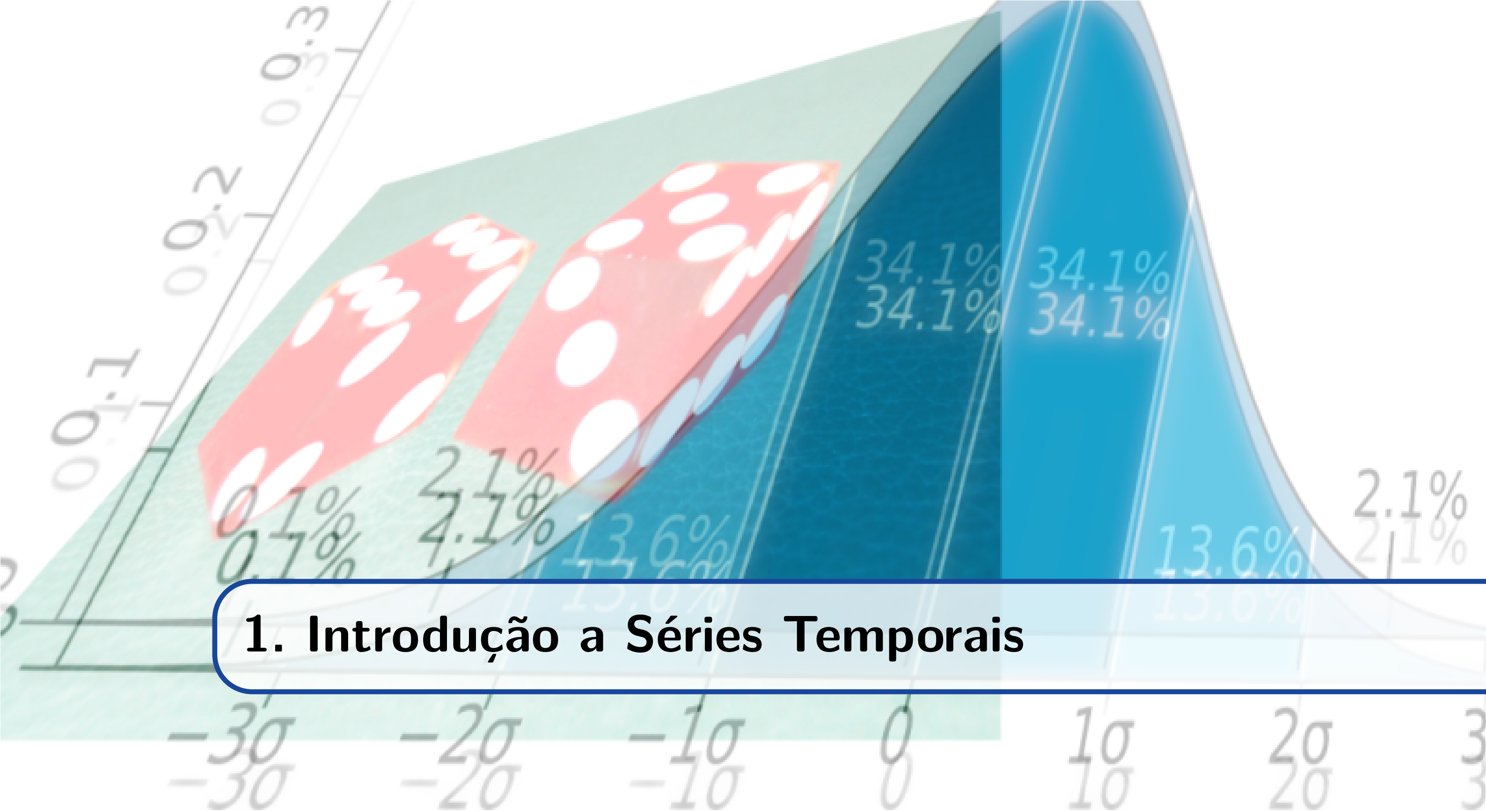
Introdução a Séries Temporais

Mário Olímpio de Menezes



Conteúdo

Introdução a Séries Temporais



1. Introdução a Séries Temporais

Nesta Trilha vamos estudar as Séries Temporais. Um *série temporal* é uma coleção de observações (medidas) tomadas em sequência, ao longo de um período. Por causa da natureza intrínseca deste tipo de fenômeno, as observações vizinhas são dependentes e o objetivo é analisar e entender esta dependência.

Outra característica importante das séries temporais, é que as observações devem ser consideradas e analisadas na ordem correta, especialmente na construção da série temporal no **R**.

Um dos maiores desafios da Análise de Séries Temporais é que a maior parte dos procedimentos estatísticos foi desenvolvida para analisar observações independentes; assim, técnicas específicas são necessárias para a análise de séries temporais.

Iniciaremos os estudos desta Trilha apresentando os conceitos básicos de séries temporais e sua manipulação no **R**. Em seguida apresentaremos algumas técnicas de modelagem, partindo de abordagens mais simples até chegarmos no modelo ARIMA. Não temos a pretenção de esgotar o assunto, e por isso, ficarão de fora do nosso estudo, estudos mais avançados onde se modela também a variabilidade da série temporal, com os modelos **ARCH** e **GARCH**. Outra técnica moderna que não abordaremos é a modelagem de séries temporais com redes neurais, especialmente, *Deep Learning*. Esta técnica tem apresentado bons resultados para a previsão de períodos futuros, independentemente dos mecanismos geradores da série.

Noções Básicas

Uma **Série temporal** é um conjunto de observações ordenadas no tempo; considera-se que o *tempo* pode ser substituiído por outra variável como espaço, profundidade, . . .

Neste conjunto, as observações vizinhas são dependentes. O estudo de séries temporais envolve: modelagens, análise dessa dependência, e outras técnicas específicas a séries temporais.

Dados de séries temporais surgem em várias áreas do conhecimento, sendo aplicadas por exemplo em:

- **Economia**: preços diários de ações; taxa de desemprego.
- **Medicina**: níveis de eletrocardiograma ou eletroencefalograma.
- **Epidemiologia**: casos semanais de sarampo; casos mensais de AIDS.
- **Metereologia**: temperatura diária; registro de marés, . . .

Formalmente, uma série temporal é o conjunto de observações

$$Y(t), t \in T$$

onde

- Y: variável de interesse,
- T: conjunto de índices.

Podemos classificar as séries temporais em:

1. Discreta: $T = \{t_1, t_2, ..., t_n\}$

   Ex: Exportações mensais de 1970 a 1980 $\{01/1970, 02/1970, ..., 11/1980, 12/1980\}$.

   Notação: Y_t

2. Contínua: $T = \{t : t_1 < t < t_2\}$

   Ex: Registro da maré do Rio durante 1 ano; $T = [0, 24]$ se a unidade de tempo é a hora.

   Notação :$Y(t)$

3. Multivariada: Observações são $Y_1(t), ..., Y_k(t), t \in T$.

   Ex: Vendas semanais $Y_1(t)$ e gastos com propaganda $Y_2(t)$.

Y pode também ser discreto ou contínuo. Muitas vezes, Y é discreto mas pode ser tratado como contínuo.

Ex.: número de casos notificados de AIDS.

Os casos mais simples são de séries univariadas, discretas e observadas em tempos equidistantes. Podemos identificar T com $\{1, 2, ..., n\}$

Objetivos de uma análise de séries temporais

Os principais objetivos de uma análise de séries temporais são:

i) Compreender o mecanismo gerador da série;

ii) Predizer o comportamento futuro da série.

Compreender o mecanismo gerador da série possibilita:

- Descrever efetivamente o comportametno da série;
- Encontrar periodicidades na série;
- Tentar obter razões para o comportamento da série (possivelmente através de variáveis auxiliares);
- Controlar a trajetória da série.

Predizer o futuro possibilita

- Fazer planos a longo, médio e curto prazos;
- Tomar decisões apropriadas.

Os objetivos **(i)** e **(ii)** rotineiramente estão interligados e com um modelo apropriado, eles são alcançáveis, a não ser nos raros casos de modelos determinísticos.

Todavia, lembramos que:

Futuro envolve incerteza $\Rightarrow$ previsões não são perfeitas.

O objetivo é reduzir ao máximo os erros de previsão.

Componentes de uma série temporal

- Em geral, pode identificar os seguintes padrões de comportamento na análise de séries temporais, utilizando-se um modelo clássico:
  - Uma tendência ao longo do tempo;
  - Um padrão sazonal, e
  - Uma componente aleatória (não identificada).
- Este padrões dificilmente aparecem isolados; em geral, surgem de forma combinada.
- A análise de séries temporais pode ser então encarada simplesmente como uma tentativa de decomposição nestas várias componenetes.

Exemplo de série temporal

As séries temporais mostradas na Figura 1 são:

- (esquerda) de dados quadrimestrais dos ganhos (em dólar) por ação da empresa Johnson & Johnson entre 1960 e 1980. São 84 observações: uma para cada quadrimestre, durante 21 anos.
- (direita) média mensal de números de manchas solares de 1749 a 1983, registrados pelo Observatório Federal da Suiça e pelo Observatório Astronômico de Tókio. Esta série é bem mais longa, com 2.820 observações — 1 por mês, durante 235 anos.

Para estas séries, podemos perguntar:

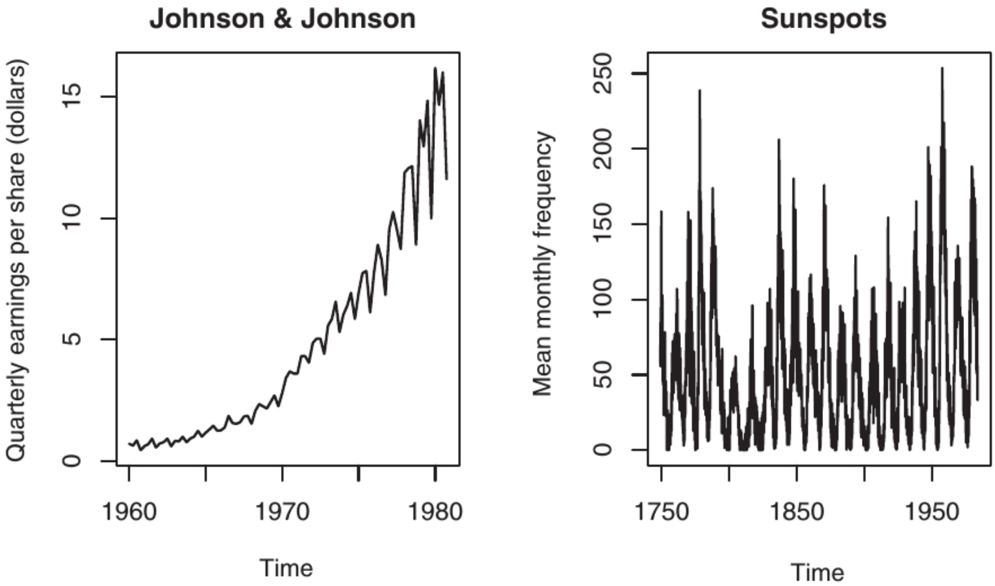
Johnson & Johnson

- Será que o preço das ações está mudando ao longo do tempo?
- Os efeitos quadrimestrais, com os preços das ações subindo e descendo, são do tipo regular ao longo do ano?
- Será que podemos prever quais serão os preços futuros e, se pudermos, com que acurácia?

Manchas Solares

- Quais modelos estatísticos descrevem melhor as atividades das manchas solares?

Figura 1: Exemplos de Séries Temporais



Fonte: KABACOFF (2015)

- Será que algum modelo se ajusta melhor aos dados do que outros?
- Será que o número de manchas solares em dado momento é previsível e, se for, em qual grau?

Estudo de Séries Temporais

A habilidade de prever preços de ações tem relevância prática, que pode ser, a garantia do seu investimento de aposentadoria. Já a previsão das tempestadas solares tem relevância, por exemplo, para a qualidade da recepção dos sinais de celulares.

Prever os valores futuros de uma série temporal, ou *forecasting*, é uma atividade humana fundamental. Os estudos de séries temporais tem importante aplicações no mundo real.

Técnicas

Há diferentes abordagens, que funcionam melhor em casos diferentes e com tipos diferentes de séries temporais. Por isso, vamos fazer uma *varredura* nos métodos e exemplos de séries temporais.

Exemplos de funções para análise de séries temporais no R

O Quadro 1 apresenta exemplos de funções disponíveis no **R** para a Análise de Séries Temporais.

Quadro 1: Exemplos de Funções para Análise de Séries Temporais no R

Função	Pacote	Uso
`ts()`	`stats`	Cria um objeto série temporal.
`plot()`	`graphics`	Plota uma série temporal.
`start()`	`stats`	Retorna o início de uma série temporal.
`end()`	`stats`	Retorna o final de uma série temporal.
`frequency()`	`stats`	Retorna o período de uma série temporal.
`cycle`	`stats`	Retorna as posições no ciclo de cada observação.
`window()`	`stats`	Subconjunto de uma série temporal.
`ma()`	`forecast`	Ajusta um modelo simples de média móvel.
`stl()`	`stats`	Decompõe a série temporal em componentes sazonal, tendência e irregular usando `loess`
`monthplot()`	`stats`	Plota a componente sazonal de uma série temporal
`seasonplot()`	`forecast`	Gera um gráfico sazonal
`HoltWinters()`	`stats`	Ajusta um modelo de suavização exponencial
`forecast()`	`forecast`	Prevê valores futuros de uma série temporal
`accuracy()`	`forecast`	Reporta as medidas de ajuste para uma série temporal
`ets()`	`forecast`	Ajusta um modelo de suavização exponencial. Inclue a habilidade de automatizar a seleção de um modelo.
`lag()`	`stats`	Retorna uma versão *retardada* de uma série temporal.
`Acf()`	`forecast`	Estima a função de autocorrelação.
`Pacf()`	`forecast`	Estima a função de autocorrelação parcial.
`diff()`	`base`	Retorna diferenças retardadas e repetidas (*iterated*)
`ndiffs()`	`forecast`	Determina o nível de diferenciação necessária para remover as tendência em uma série temporal.
`adf.test()`	`tseries`	Calcula um teste *Augmented Dickey-Fuller* de que uma série temporal seja estacionária
`arima()`	`stats`	Ajusta um modelo de média móvel auto regressiva
`Box.test()`	`stats`	Calcula um teste *Ljung-Box* de que os resíduos de uma série temporal sejam independentes.
`bds.test()`	`tseries`	Calcula o teste *BDS* de que uma série consista de variáveis aleatórias independentes, identicamente distribuídas.
`auto.arima()`	`forecast`	Automatiza a selação de um modelo ARIMA

Fonte: KABACOFF (2015). Traduzido e adaptado pelo autor.

Criando um objeto série temporal no R

Para trabalharmos com uma série temporal no **R**, temos que colocá-la em um *objeto série temporal* – uma estrutura do **R** que contém:

- as observações,
- os tempos de início e final da série e
- informação sobre sua periodicidade (por exemplo, mensal, quadrimestral ou anual)

Com os dados em um objeto série temporal podemos utilizar inúmeras funções para manipulá-lo, modelar e plotar os dados.

Um vetor de números ou uma coluna em um `data.frame` pode ser salvo como um objeto série temporal utilizando a função `ts()`. O formato é:

`myseries <- ts(data, start=, end=, frequency=)`
onde:

- `myseries` é o objeto série temporal
- `data` é um vetor numérico contendo as observações
- `start` especifica o tempo de início da série
- `end` especifica o tempo de fim da série
- `frequency` indica o número de observações por unidade de tempo:
  - `frequency=1` para dados anuais,
  - `frequency=12` para dados mensais, e
  - `frequency=4` para dados quadrimestrais)

O exemplo abaixo ilustra a criação de uma série temporal no **R** .

Criamos um *vetor* com os valores das observações colhidas ao longo do tempo e depois, utilizando o comando `ts` criamos nossa série temporal.

```
> # vetor com as observacoes
> sales <- c(18, 33, 41, 7, 34, 35, 24, 25, 24, 21, 25,
...      20, 22, 31, 40, 29, 25, 21, 22, 54, 31, 25, 26,
...      35)
```

```
> # cria um objeto serie temporal informamos ano, mes
> # de inicio e tambem a frequencia das observacoes
> # (mensais).
> tsales <- ts(sales, start = c(2003, 1), frequency = 12)
> tsales
     Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
2003  18  33  41   7  34  35  24  25  24  21  25  20
2004  22  31  40  29  25  21  22  54  31  25  26  35
> # obtem informacao sobre o objeto
> start(tsales)
[1] 2003    1
> end(tsales)
[1] 2004   12
> frequency(tsales)
[1] 12
> # pega subconjunto do objeto
> tsales.subset <- window(tsales, start = c(2003, 5),
...     end = c(2004, 6))
> tsales.subset
     Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
2003                  34  35  24  25  24  21  25  20
2004  22  31  40  29  25  21
```

Podemos também visualizar nossa série temporal com o comando `plot` do objeto `ts` conforme mostrado na Figura 2.

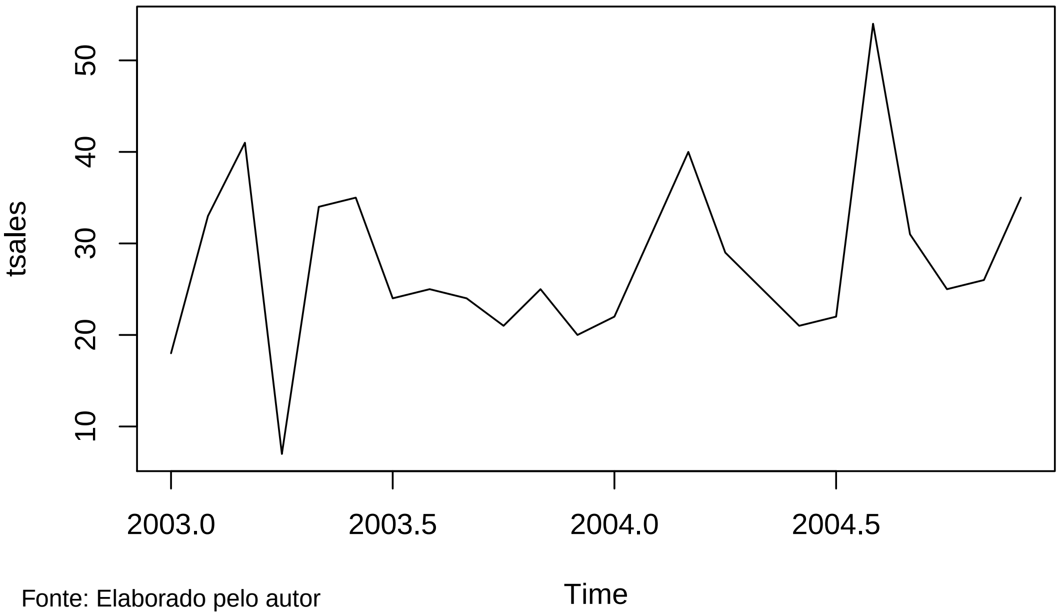
```
> plot(tsales)
> text(2003, -9, "Fonte: Elaborado pelo autor", xpd = NA,
...     cex = 0.8)
```

Exploração de Séries Temporais no R

Manipulando e Visualizando Séries Temporais

Vamos utilizar um conjunto de dados chamado `AirPassengers` que vem com a instalação base do **R**. Esta base descreve os totais mensais (em milhares) de passageiros aéreos internacionais entre 1949 e 1960.

Figura 2: Visualização da série temporal



Fonte: Elaborado pelo autor

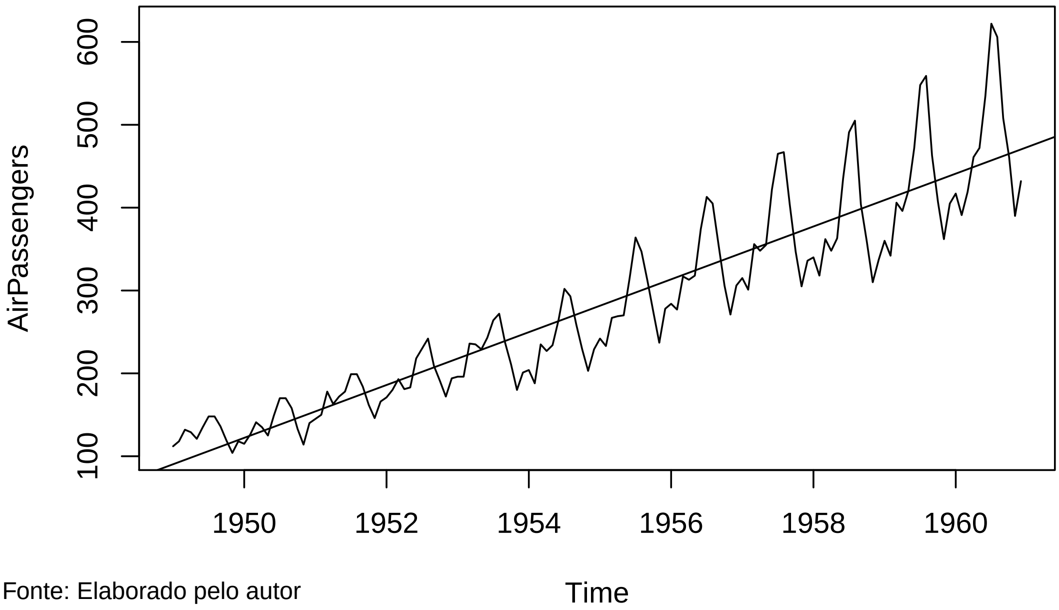
A seguir, alguns comandos do **R** para iniciar nossa exploração.

```
> data("AirPassengers")
> class(AirPassengers)
[1] "ts"
> start(AirPassengers)
[1] 1949    1
> end(AirPassengers)
[1] 1960   12
> frequency(AirPassengers)
[1] 12
> summary(AirPassengers)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  104.0   180.0   265.5   280.3   360.5   622.0 
```

Uma das primeiras coisas que devemos fazer na exploração de um conjunto de dados é a **visualização**. Adicionamos uma linha de regressão linear simples, como mostrado na Figura 3.

```
> plot(AirPassengers)
> abline(reg = lm(AirPassengers ~ time(AirPassengers)))
> text(1948.7, -65, "Fonte: Elaborado pelo autor", xpd = NA,
...      cex = 0.8)
```

Figura 3: Visualização da série temporal com uma linha de regressão



Fonte: Elaborado pelo autor

O comando `cycle` nos mostra as posições dentro do ciclo de cada observação. No nosso caso, como temos observações mensais, temos a indicação da posição de cada observação dentro de cada ano.

```
> cycle(AirPassengers)
     Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
1949   1   2   3   4   5   6   7   8   9  10  11  12
1950   1   2   3   4   5   6   7   8   9  10  11  12
1951   1   2   3   4   5   6   7   8   9  10  11  12
1952   1   2   3   4   5   6   7   8   9  10  11  12
1953   1   2   3   4   5   6   7   8   9  10  11  12
1954   1   2   3   4   5   6   7   8   9  10  11  12
1955   1   2   3   4   5   6   7   8   9  10  11  12
1956   1   2   3   4   5   6   7   8   9  10  11  12
1957   1   2   3   4   5   6   7   8   9  10  11  12
1958   1   2   3   4   5   6   7   8   9  10  11  12
1959   1   2   3   4   5   6   7   8   9  10  11  12
1960   1   2   3   4   5   6   7   8   9  10  11  12
```

A função `aggregate` quando aplicada a um objeto do tipo `time.series` do **R** retorna a função especificada no argumento `FUN` aplicada em blocos apropriados de comprimento `frequency(x)/nfrequency`. O argumento `nfrequency` (valor padrão 1) indica o novo número de observações por unidade de tempo; deve ser um divisor da frequencia da série original.

Assim, temos, `frequency(AirPassengers)` = 12. Utilizaremos o valor padrão de `nfrequency`=1.

Chamando a função `aggregate` na série `AirPassengers` com o parâmetro `FUN=mean`, teremos uma nova série temporal como uma nova frequência = 1. Os novos valores da série serão a média dos valores ano a ano.

```
> aggregate(AirPassengers, FUN = mean)
Time Series:
Start = 1949
End = 1960
Frequency = 1
 [1] 126.6667 139.6667 170.1667 197.0000 225.0000 238.9167 284.0000 328.2500 368.4167 381.0000 428.3333 476.1667
```

Podemos também obter os valores cumulativos ano a ano:

```
> aggregate(AirPassengers, FUN = sum)
Time Series:
Start = 1949
End = 1960
Frequency = 1
 [1] 1520 1676 2042 2364 2700 2867 3408 3939 4421 4572 5140 5714
```

Outra possibilidade, é obtermos os valores de sumários estatísticos mês a mês para toda a série.

Por exemplo, com a seguinte chamada do comando `aggregate` obteremos os valores médios para cada mês; por exemplo, o primeiro valor do comando abaixo mostra o valor médio de todos os Janeiros de todo o período da série temporal, e assim, subsequentemente, para todos os demais meses do ano.

```
> aggregate(AirPassengers ~ cycle(AirPassengers), FUN = mean)
   cycle(AirPassengers) AirPassengers
1                     1      241.7500
2                     2      235.0000
3                     3      270.1667
4                     4      267.0833
5                     5      271.8333
6                     6      311.6667
7                     7      351.3333
8                     8      351.0833
9                     9      302.4167
10                   10      266.5833
```

```
11                    11      232.8333
12                    12      261.8333
```

Também podemos obter as medianas de cada mês:

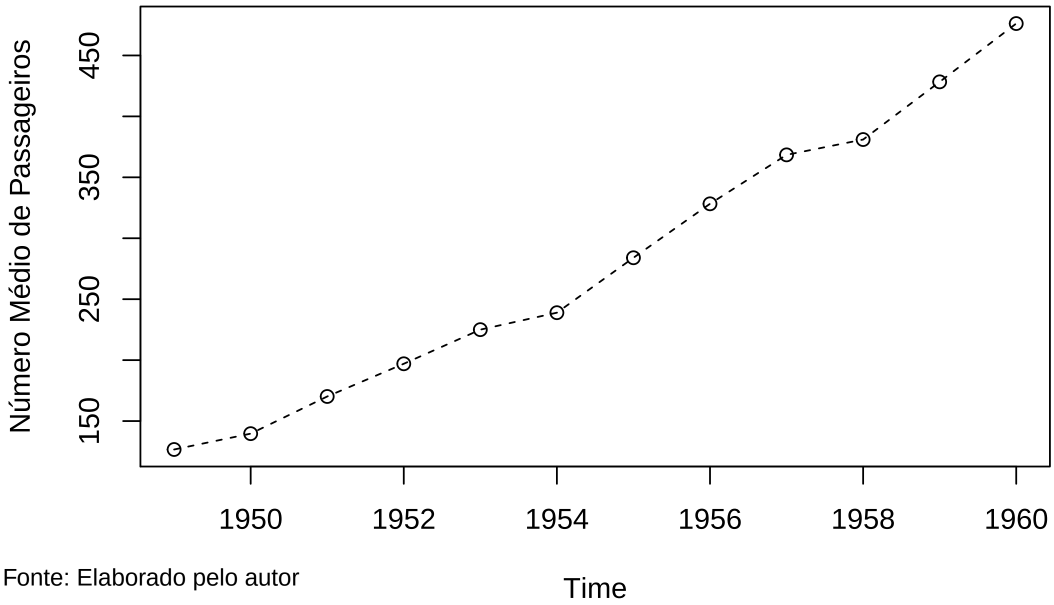
```
> aggregate(AirPassengers ~ cycle(AirPassengers), FUN = median)
   cycle(AirPassengers) AirPassengers
1                     1         223.0
2                     2         214.5
3                     3         251.5
4                     4         252.0
5                     5         252.0
6                     6         289.5
7                     7         333.0
8                     8         320.0
9                     9         285.5
10                   10         251.5
11                   11         220.0
12                   12         253.5
```

As variâncias dos meses:

```
> aggregate(AirPassengers ~ cycle(AirPassengers), FUN = var)
   cycle(AirPassengers) AirPassengers
1                     1     10207.659
2                     2      8031.636
3                     3     10112.152
4                     4     11529.356
5                     5     13165.242
6                     6     18014.970
7                     7     24594.788
8                     8     24268.447
9                     9     15364.629
10                   10     12264.447
11                   11      9060.333
12                   12     10628.333
```

Agora, visualizamos a nova série agregada, mostrando os valores médios ano a ano, como mostrado na Figura 4:

Figura 4: Série Temporal com valores médios anuais



Fonte: Elaborado pelo autor

```
> plot(aggregate(AirPassengers, FUN = mean), type = "o",
...      lty = 2, ylab = "Número Médio de Passageiros")
> text(1948.7, 20, "Fonte: Elaborado pelo autor", xpd = NA,
...      cex = 0.8)
```

Outra exploração interessante que podemos fazer é criar um gráfico do tipo `boxplot` para os valores anuais da nossa série temporal.

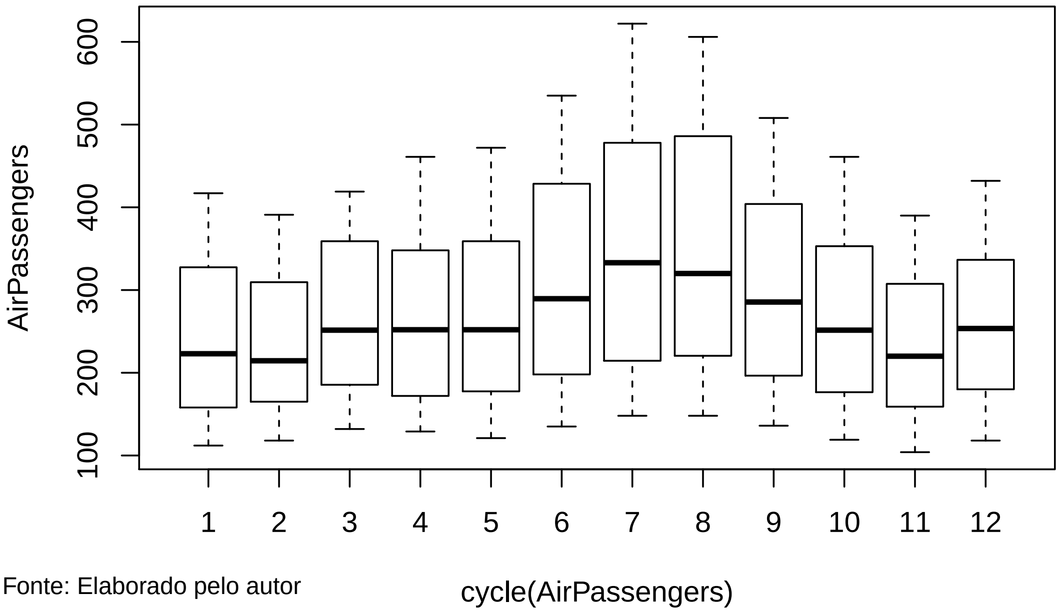
O comando abaixo nos dá este resultado. O gráfico da Figura 5 mostra, para cada mês, em todos os anos, a distribuição dos valores da série. Assim, podemos ver que os meses de Julho e Agosto apresentam os maiores valores da *mediana*

```
> boxplot(AirPassengers ~ cycle(AirPassengers))
> text(0.2, -60, "Fonte: Elaborado pelo autor",
...      xpd = NA, cex = 0.8)
```

Inferências importantes desta exploração

1. A tendência de ano a ano claramente mostra que o número de passageiros tem aumentado, sem falhar.

Figura 5: Distribuição dos valores da série temporal AirPassengers mês a mês



Fonte: Elaborado pelo autor

2. O valor da variância e da média em Julho e Agosto são muito maiores do que no resto dos meses.

3. Apesar de o valor da média em cada mês ser bem diferente, sua variância é pequena. Portanto, temos um forte efeito sazonal com um ciclo de 12 meses ou menos.

Exploração – sumarizando

A exploração é uma etapa muito importante na modelagem de séries temporais. Sem esta exploração, não sabemos se a série é estacionária ou não. A partir desta exploração temos alguns elementos para prosseguir com a modelagem

Suavização e decomposição sazonal

Uma maneira de se ganhar *insights* na análise de dados tradicional é através da exploração dos dados. Com séries temporais podemos suavizar a série visando clarificar sua tendência geral. Também podemos decompor a série para observar quaisquer efeitos sazonais. Um primeiro passo na investigação de uma série temporal é fazer o gráfico (como fizemos acima).

Séries temporais geralmente tem um comportamento irregular — ou componente de erro. Para discernir padrões nos dados, frequentemente plotamos um gráfico suavizado destas flutuações.

Um dos métodos mais simples de suavização de uma série de tempo é usar médias móveis.

Por exemplo, cada ponto de dado é substituído pela média daquela observação e uma observação anterior e outra posterior — isto é chamado de *média móvel central*, dada pela Equação (1).

$$S_t = (Y_{t-q} + \ldots + Y_t + \ldots + Y_{t+1})/(2q + 1) \tag{1}$$

onde:

- S_t é o valor suavizado no tempo t e $k = 2q$ é o número de observações que são consideradas no cálculo.
- o valor k é usualmente escolhido como um número ímpar (3 neste exemplo)

Forçosamente, quando utilizamos uma média móvel central perdemos as $(k-1)/2$ observações em cada ponto da série.

No **R**, utilizamos a função `ma` do pacote `forecast` para realizar uma suavização com média móvel em nossa série temporal. O parâmetro k é o segundo argumento passado para a função `ma`.

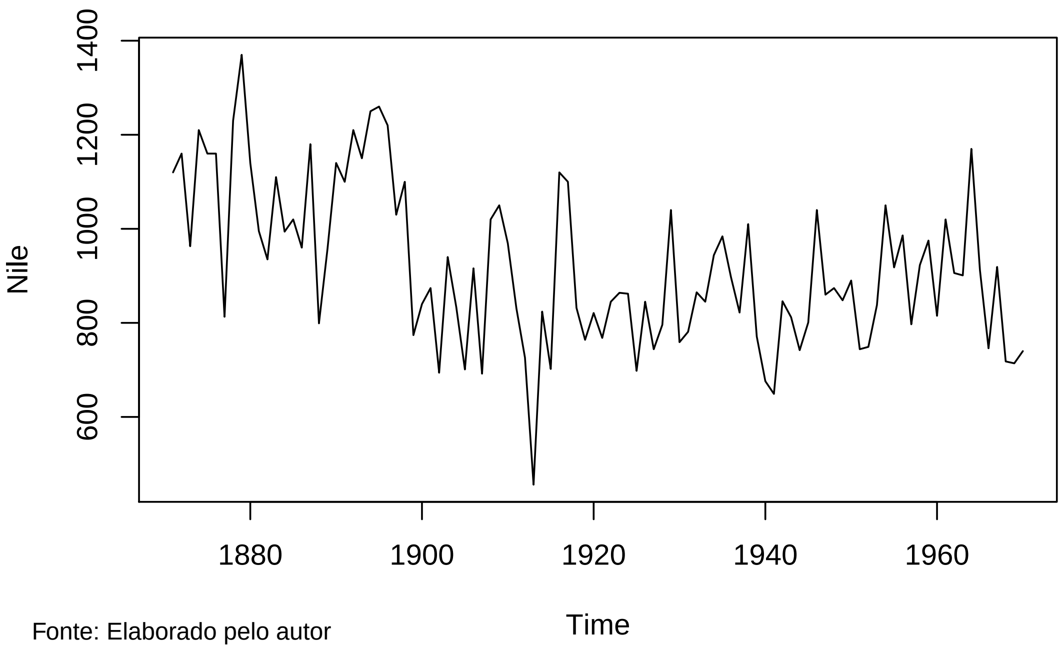
O exemplo abaixo, utiliza a série temporal `Nile` que apresenta medidas do fluxo anual do rio Nilo em *Aswan*, entre 1871–1970, em $10^8 m^3$, com uma mudança aparente por volta de 1898. Veja mais informações em `?Nile` no **R**, depois de carregar o pacote `forecast`.

```
> library("forecast")
> set.seed(123456)
> # opar <- par(no.readonly = TRUE) par(mfrow=c(2,2))
> ylim <- c(min(Nile), max(Nile))
> plot(Nile)
> text(1872, 140, "Fonte: Elaborado pelo autor", xpd = NA,
...      cex = 0.8)
```

```
> plot(ma(Nile, 3), ylim = ylim)
> text(1872, 140, "Fonte: Elaborado pelo autor", xpd = NA,
...      cex = 0.8)
```

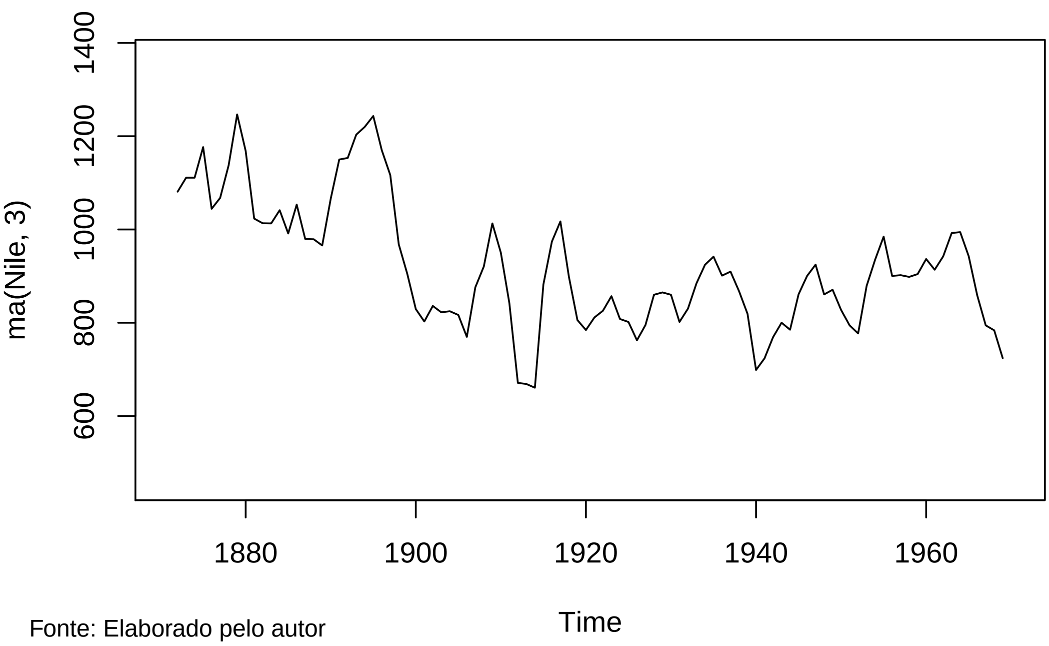
```
> plot(ma(Nile, 7), ylim = ylim)
> text(1872, 140, "Fonte: Elaborado pelo autor", xpd = NA,
...      cex = 0.8)
```

Figura 6: Série Temporal Nile original



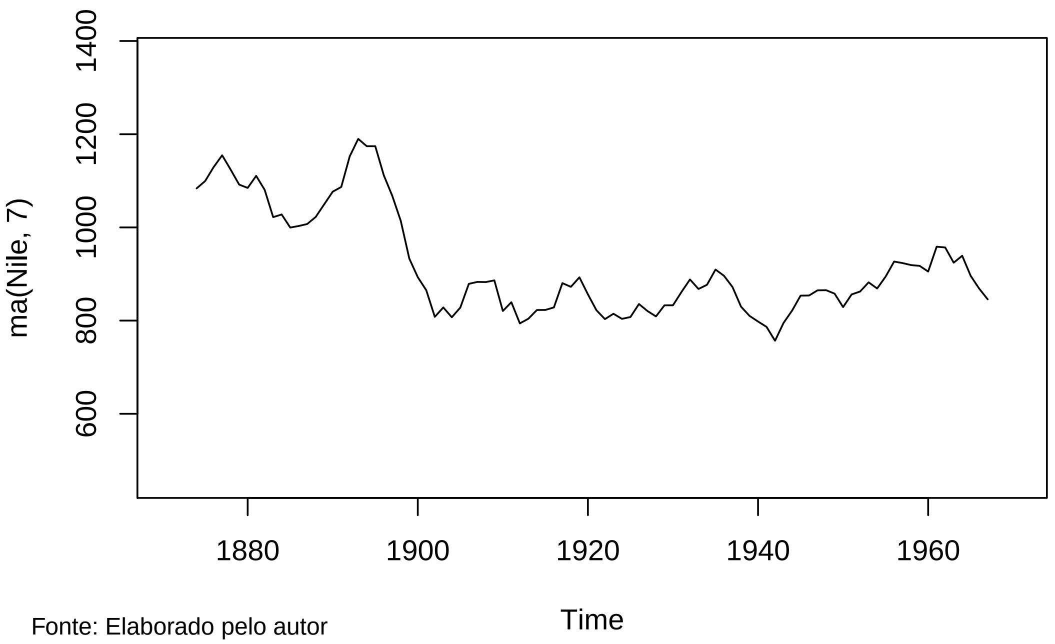
Fonte: Elaborado pelo autor

Figura 7: Série Temporal Nile - Média Móvel k=3



Fonte: Elaborado pelo autor

Figura 8: Série Temporal Nile - Média Móvel k=7



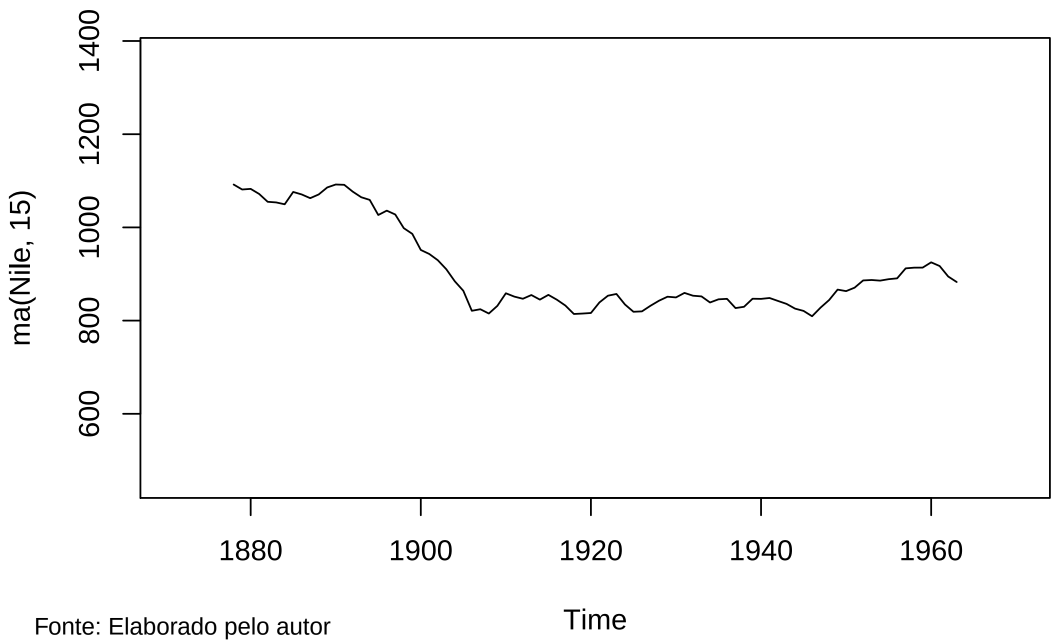

```
> plot(ma(Nile, 15), ylim = ylim)
> text(1872, 140, "Fonte: Elaborado pelo autor", xpd = NA,
...      cex = 0.8)
```

Observamos pelos gráficos acima que conforme k aumenta, o gráfico se torna mais suave. O desafio é achar o valor de k que destaque os padrões relevantes nos dados sem uma sub-suavização ou super-suavização. Isto é mais arte do que ciência.... por isso tentamos vários valores de k antes de decidir por um valor.

Dos gráficos mostrados, parece certo que houve uma diminuição no fluxo do rio entre 1892 e 1900. Outras mudanças podem ser interpretadas de várias maneiras também.

☞ Quando uma série temporal tem uma periodicidade maior do que um (isto é, tem um componente sazonal), devemos ir além da descrição de uma tendência geral. A decomposição sazonal pode ser utilizada para examinar tanto tendência geral como sazonal.

Figura 9: Série Temporal Nile - Média Móvel k=15



Fonte: Elaborado pelo autor

Decomposição sazonal

Séries de tempo que tem um aspecto sazonal (tais como dados mensais ou quadrimestrais) podem ser decompostos em:

- um componente de tendência
- um componente sazonal
- um componente irregular

O componente de tendência captura as mudanças no nível ao longo do tempo. O componente sazonal captura efeitos cíclicos devido a época do ano. O componente irregular (ou erro) captura aquelas influências não descritas por efeitos de tendência e sazonais.

A decomposição pode ser aditiva ou multiplicativa.

Modelo Aditivo

Neste modelo, os componentes são somados para dar os valores da série de tempo.

$$Y_t = \text{Tendência}_t + \text{Sazonal}_t + \text{Irregular}_t$$

Modelo Multiplicativo

Neste modelo, os componentes são multiplicados para dar os valores da série de tempo.

$$Y_t = \text{Tendência}_t \times \text{Sazonal}_t \times \text{Irregular}_t$$

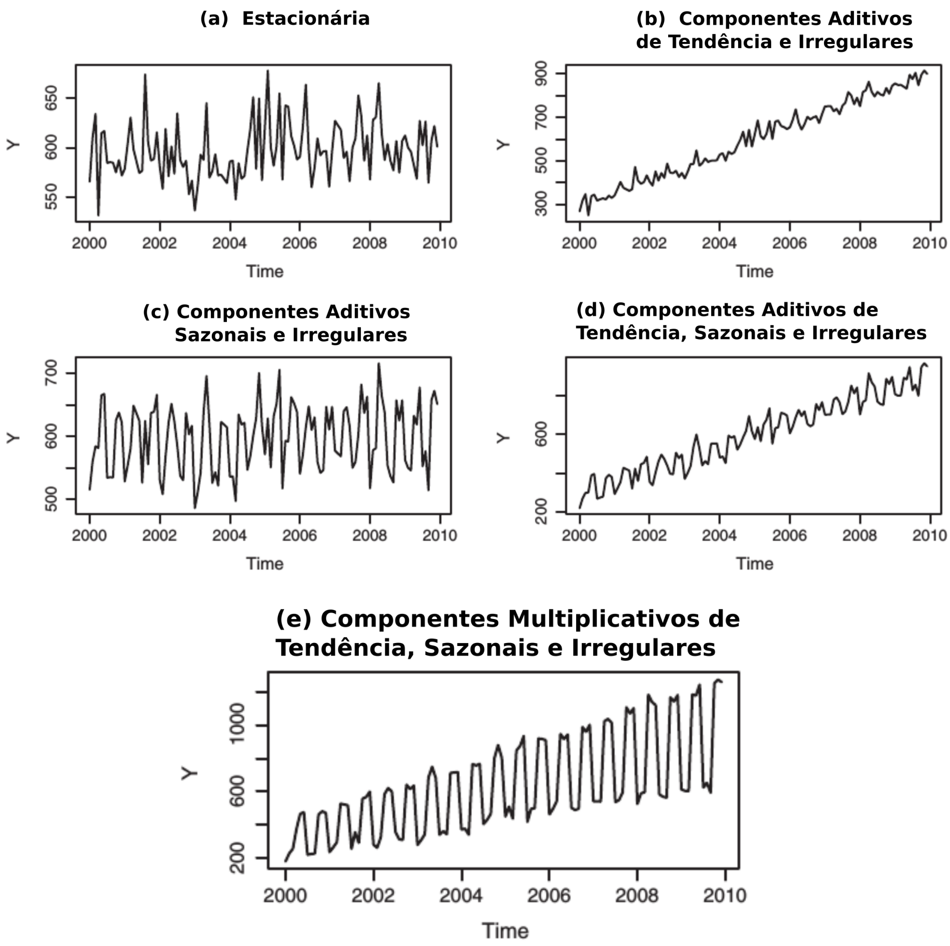
Os gráficos mostrados na Figura 10 ilustram estes casos.

Para cada gráfico, identificados pelas letras, observamos que:

- **a** – não há nem tendência nem componente sazonal – somente flutuação aleatória em torno de um nível.
- **b** – há uma tendência crescente ao longo do tempo, bem como flutuações aleatórias
- **c** – há efeitos sazonais e flutuações aleatórias, mas não há uma tendência que se afaste de uma linha reta.
- **d** – todos os três componentes estão presentes: tendência crescente, efeitos sazonais e flutuações aleatórias.
- **e** – do mesmo modo que no gráfico **d**, mas agora com uma combinação multiplicativa.
  - a variabilidade é proporcional ao nível: conforme o nível aumenta, aumenta a variabilidade.
  - esta amplificação (ou diminuição) baseado no nível atual da série sugere fortemente um modelo multiplicativo.

Uma maneira comum de decompor uma série nos componentes tendência, sazonal e irregular é pela suavização *loess*.

Figura 10: Exemplos de Séries Temporais



Fonte: KABACOFF (2015).

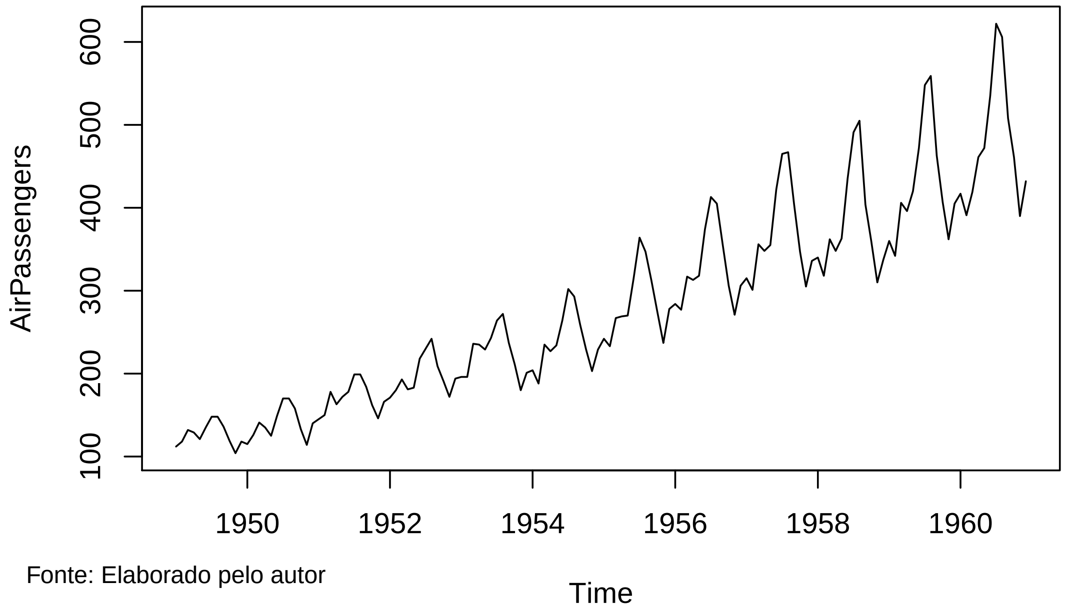
Suavização *loess* ou *lowess*

Esta técnica é um método de regressão, cujo nome significa *LOcal regrESSion* ou **LO**cally **W**eighted polynomial regre**SS**ion.

Nesta técnica em cada ponto do conjunto de dados, um polinônio de baixo grau é ajustado a um subconjunto dos dados. Os valores da variável explicativa próximos àquele ponto, cuja resposta está sendo estimada, recebem mais peso, e os pontos mais distantes, menos peso.

Podemos fazer isso no **R** com a função `stl()`:

Figura 11: Série temporal AirPassengers




```
> stl(ts, s.window = , t.window = )
```

A função `stl()` lida somente com modelos aditivos, mas isto não é um problema sério, já que modelos multiplicativos podem ser transformados em modelos aditivos utilizando uma transformação **log**:

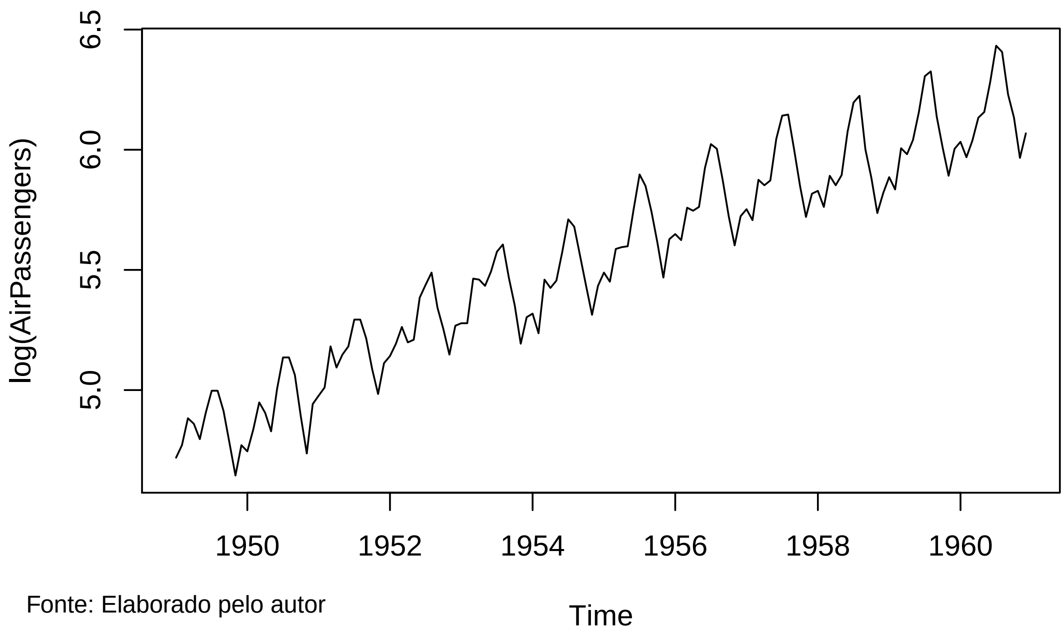
$$\log(Y_t) = \log(\text{Tendência}_t \times \text{Sazonal}_t \times \text{Irregular}_t)$$

$$\log(Y_t) = \log(\text{Tendência}_t) + \log(\text{Sazonal}_t) + \log(\text{Irregular}_t)$$

Após se ajustar o modelo aditivo à série *log-transformada*, os resultados podem ser *transformados de volta* para a escala original. Utilizaremos novamente a base de dados `AirPassengers` que vem com o instalação base do **R**. Um gráfico dos dados é mostrado na Figura 11.

```
> plot(AirPassengers)
> text(1949, -45, "Fonte: Elaborado pelo autor", xpd = NA,
...      cex = 0.8)
```

Figura 12: Série temporal AirPassengers – log



Como observamos no gráfico acima, a irregularidade (variabilidade) das observações aumenta ao longo do tempo, indicando um modelo multiplicativo da série.

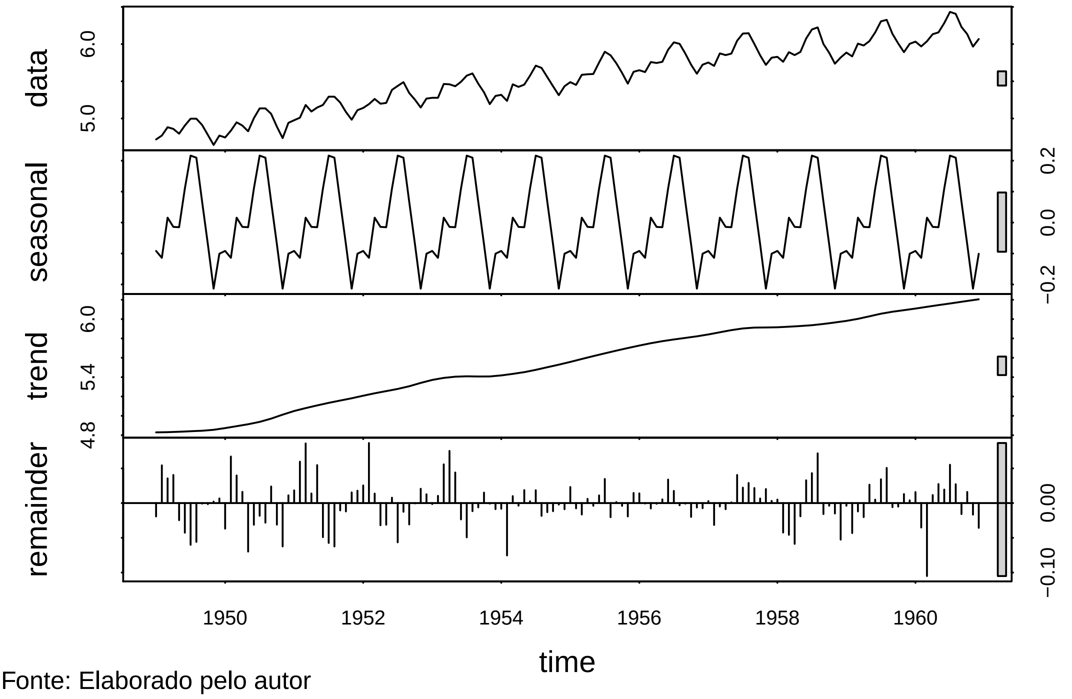
Então, primeiro fazemos a transformação `log` na série e plotamos novamente o gráfico para inspecionar, conforme mostrado na Figura 12.

```
> lAirPassengers <- log(AirPassengers)
> plot(lAirPassengers, ylab = "log(AirPassengers)")
> text(1949, 4.1, "Fonte: Elaborado pelo autor",
...      xpd = NA, cex = 0.8)
```

Agora, ajustamos o modelo com a função `stl` e depois utilizamos a função `plot` do objeto `stl` para visualizar graficamente o resultado do ajuste; no gráfico, mostrado na Figura 13 temos:

- o gráfico dos dados originais – no nosso caso, já transformados pela função `log` de 1949 a 1960;
- a componente sazonal, isto é, o gráfico que mostra a sazonalidade dos dados; os componentes sazonais foram limitados a permanecer os mesmos através de cada ano (utilizando a opção `s.window = "period"`).
- a componente de tendência, isto é, o gráfico que mostra a tendência presente nos dados; como é uma função de otimização local, a tendência parece com uma curva de suavização.

Figura 13: Ajuste do Modelo Aditivo - Série temporal AirPassengers-log



Fonte: Elaborado pelo autor

A tendência é monotonicamente crescente, e o efeito sazonal sugere mais passageiros no verão (talvez durante as férias).

- e por fim, a componenete de irregularidade, isto é, os resíduos do ajuste, o que não foi explicado pelo modelo.
- As barras cinzas à direita são os tamanhos das guias – cada barra representa a mesma magnitude. Isso é útil porque os valores do eixo y são diferentes para cada gráfico.

```
> fit <- stl(lAirPassengers, s.window = "period")
> plot(fit)
> text(1949, -0.102, "Fonte: Elaborado pelo autor",
...     xpd = NA, cex = 0.8)
```

O objeto retornado pela função `stl()` contém um componente chamado `time.series` que contém os valores da tendência, sazonal e irregular para cada observação. Neste caso, `fit$time.series` é baseado na série temporal com `log`. A expressão `exp(fit$time.series)` faz a conversão de volta à métrica original.

Os valores ajustados (transformados pela função `log`):

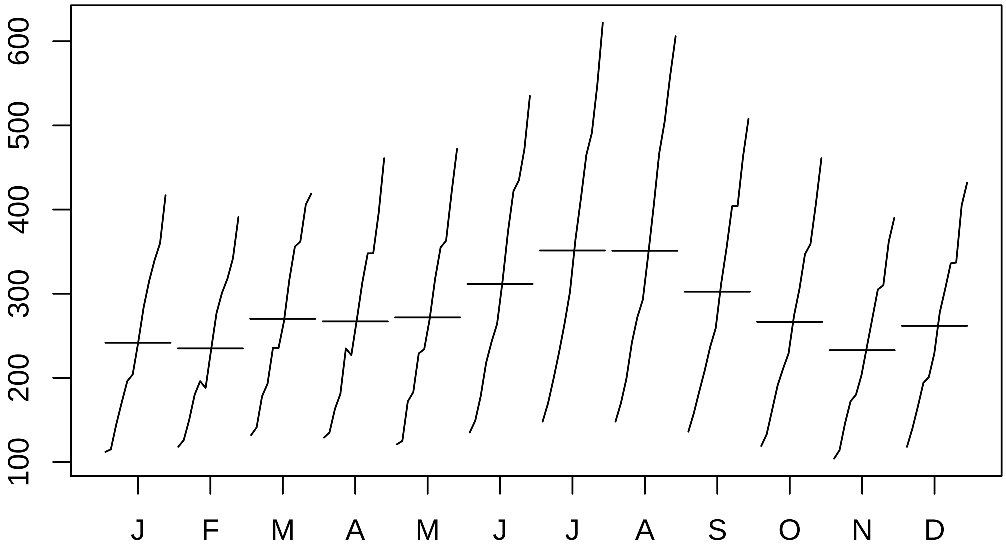
```
> options(max.print = 36, warn = -1, show.error.messages = FALSE)
> fit$time.series
            seasonal    trend     remainder
Jan 1949 -0.09164042 4.829389 -0.0192493585
Feb 1949 -0.11402828 4.830368  0.0543447685
Mar 1949  0.01586585 4.831348  0.0355884457
Apr 1949 -0.01402759 4.833377  0.0404632511
May 1949 -0.01502478 4.835406 -0.0245905300
Jun 1949  0.10978976 4.838166 -0.0426814256
Jul 1949  0.21640041 4.840927 -0.0601151688
Aug 1949  0.20960587 4.843469 -0.0558624690
Sep 1949  0.06747156 4.846011 -0.0008273977
Oct 1949 -0.07024836 4.850883 -0.0015112948
Nov 1949 -0.21352774 4.855756  0.0021630667
Dec 1949 -0.10063625 4.864586  0.0067346600
 [ reached getOption("max.print") -- omitted 132 rows ]
> options(max.print = 99999)
```

Os valores ajustados (originais, depois de aplicar a função inversa do `log`, a `exp`):

```
> options(max.print = 36, warn = -1, show.error.messages = FALSE)
> exp(fit$time.series)
          seasonal    trend remainder
Jan 1949 0.9124332 125.1344 0.9809347
Feb 1949 0.8922327 125.2571 1.0558486
Mar 1949 1.0159924 125.3798 1.0362293
Apr 1949 0.9860703 125.6345 1.0412930
May 1949 0.9850875 125.8897 0.9757094
Jun 1949 1.1160434 126.2377 0.9582166
Jul 1949 1.2415994 126.5866 0.9416561
Aug 1949 1.2331919 126.9088 0.9456692
Sep 1949 1.0697998 127.2318 0.9991729
Oct 1949 0.9321623 127.8533 0.9984898
Nov 1949 0.8077298 128.4777 1.0021654
Dec 1949 0.9042619 129.6173 1.0067574
 [ reached getOption("max.print") -- omitted 132 rows ]
> options(max.print = 99999)
```

Um exame dos efeitos sazonais sugere que o número de passageiros aumentou por 24% em Julho(um multiplicador de 1.24) e diminuiu por 20% em Novembro (com um multiplicador de

Figura 14: Distribuição dos valores da série AirPassengers, mês a mês



Fonte: Elaborado pelo autor

0.80).

Dois gráficos adicionais ajudam a visualiar uma decomposição sazonal. Eles são criados pela função `monthplot()` que vem com o pacote básico do **R** e a função `seasonplot()` provida pelo pacote `forecast`.

A Figura 14 mostra a distribuição dos valores, mês a mês, utilizando a função `monthplot()`, e a Figura 15 mostra a evolução, ano a ano, utilizando a função `seasonplot()`.

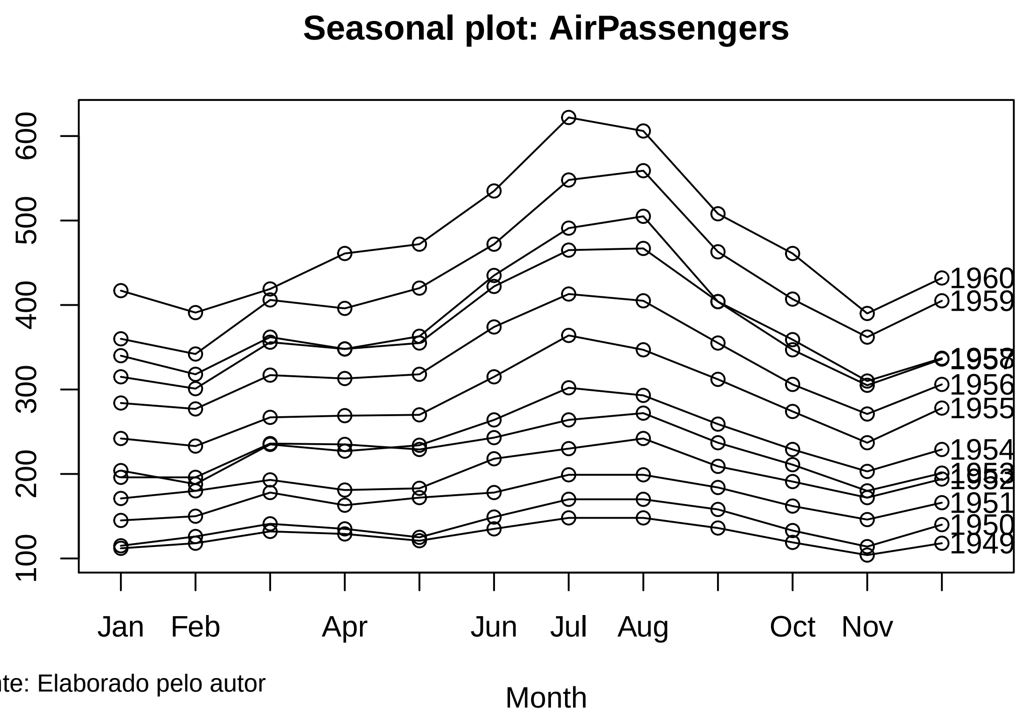
```
> monthplot(AirPassengers, xlab = "", ylab = "")
> text(0.5, -40, "Fonte: Elaborado pelo autor",
...      xpd = NA, cex = 0.8)
```

```
> seasonplot(AirPassengers, year.labels = "TRUE")
> text(0.9, -50, "Fonte: Elaborado pelo autor",
...      xpd = NA, cex = 0.8)
```

O primeiro gráfico, Figura 14, mostra a subsérie para cada mês:

- todos os valores de Janeiro conectados, todos os valores de Fevereiro conectados, e assim por diante.
- a média de cada subsérie é mostrada também.

Figura 15: Evolução, ano a ano, dos valores da série AirPassengers



Fonte: Elaborado pelo autor

- deste gráfico parece que temos uma tendência de aumento mês após mês de uma maneira aparentemente uniforme.
- adicionalmente, o maior o número de passageiros ocorre em Julho e Agosto.

O gráfico sazonal, Figura 15, mostra as subséries por ano. Deste gráfico vemos, novamente, um padrão similar, com aumento de passageiros a cada ano, e o mesmo padrão sazonal. Apesar de termos descrito a série, não fizemos nenhuma predição de valores futuros.

! Vamos examinar agora como fazer previsão de valores futuros de séries temporais. Iniciaremos por abordagens mais simples e evoluiremos ao longo da trilha.

Modelo de previsão exponencial

Modelos exponenciais são algumas das abordagens mais populares para previsão de valores futuros de séries temporais. Eles são mais simples do que outros tipos de modelos, mas podem dar bons resultados para previsões de curto prazo em uma ampla gama de aplicações. A diferença entre eles está nos componentes das séries temporais que são modelados.

Um modelo exponencial simples (também chamado *modelo exponencial único*) ajusta uma série temporal que tem apenas um nível constante e um componente irregular i mas não tem nem uma tendência nem uma componente sazonal.

Um modelo exponencial duplo (também chamado *suavização exponencial Holt*) ajusta uma série temporal com ambos, uma tendência e um nível.

Finalmente, um modelo exponencial triplo (também chamado *suavização exponencial Holt-Winters*) ajusta uma série temporal com um componente de nível, um componente de tendência e um componente de sazonalidade.

Modelos exponenciais podem ser ajustados tanto com a função `HoltWinters()`, parte da instalação base do **R**, como com a função `ets()` que vem com o pacote `forecast`.

A função `ets()` tem mais opções e é geralmente mais poderosa. Nós a utilizaremos.

O formato da função `ets()` é:

```
> ets(ts, model = "ZZZ")
```

onde ts é a série temporal e o modelo é especificado por três letras:

- a primeia letra denota o tipo de erro,
- a segunda letra denota o tipo de tendência,
- a terceira letra denota o tipo de sazonalidade.

As letras permitidas são:

- **A** para aditivo
- **M** para multiplicativo
- **N** para nenhum
- **Z** para selecionado automaticamente.

O Quadro 2 mostra modelos e parâmetros comuns:

Quadro 2: Tipos de Modelos e parâmetros da função `ets`

Tipo	Parâmetros ajustados	Funções
`simple`	`level`	`ets(ts, model="ANN")` `ses(ts)`
`double`	`level, slope`	`ets(ts, model="AAN")` `holt(ts)`
`triple`	`level,slope,seasonal`	`ets(ts, model="AAA")` `hw(ts)`

Fonte: KABACOFF (2015). Traduzido e adaptado pelo autor.

As funções `ses()`, `holt()` e `hw()` são conveniências para a função `ets()` com os defaults pré-especificados.

Suaviação exponencial simples

A suavização exponencial simples utiliza uma média ponderada dos valores existentes de uma série de tempo para fazer uma previsão de curto prazo de valores futuros. Os pesos são escolhidos de modo que observações tenham um impacto decrescente exponencialmente nas médias conforme se volta no tempo.

O modelo de suavização exponencial simples assume que uma observação na série de tempo pode ser descrita por:

$$Y_t = \text{Nível} + \text{Irregular}_t$$

A previsão no tempo Y_{t+1} (chamado de *previsão um passo à frente*) é escrita como:

$$Y_{t+1} = c_0 Y_t + c_1 Y_{t-1} + c_2 Y_{t-2} + c_3 Y_{t-3} + \ldots$$

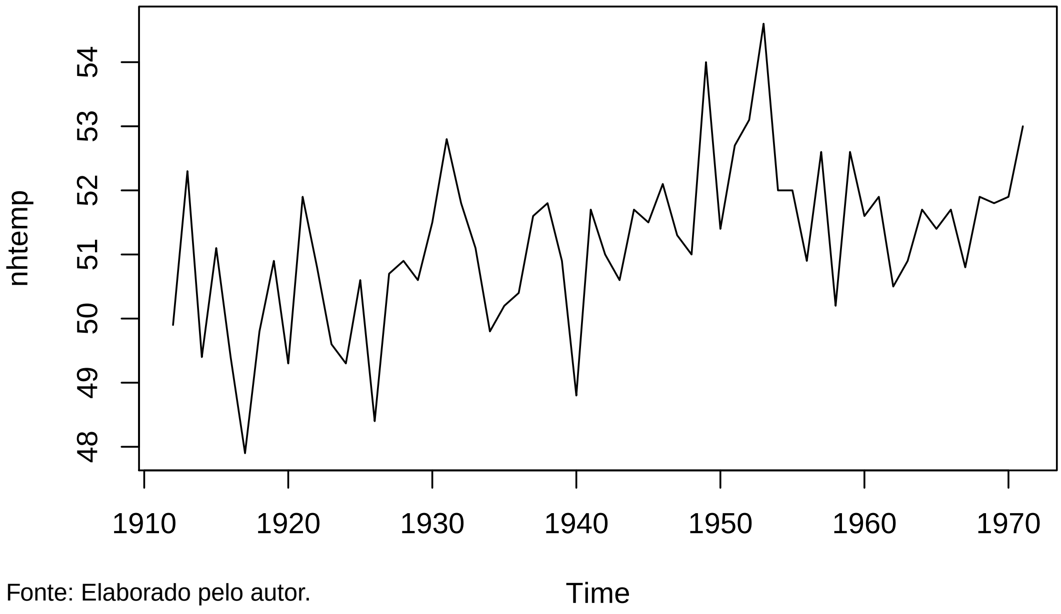
onde:

- $c_i = \alpha(1-\alpha)^i,\ i = 0, 1, 2, \ldots$ e $0 \leq \alpha \leq 1$.

A soma dos pesos c_i é igual um.

A previsão um passo à frente pode ser vista como uma média ponderada do valor atual e todos os valores passados da série temporal. O parâmetro alfa (α) controla a taxa de decaimento para os pesos.

Figura 16: Temperatura média anual New Haven



Fonte: Elaborado pelo autor.

- Quanto mais próximo de 1 alfa estiver, mais peso é dado a observações recentes.
- Quanto mais próximo de 0 alfa estiver, mais peso é dado às observações passadas.

O valor real de alfa é usualmente escolhido pelo computador para otimizar um critério de ajuste. Um critério comum de ajuste é a soma dos quadrados dos erros entre os valores real e predito. Vamos ver um exemplo. . . .para ilustrar estas ideias.

A série de tempo `nhtemp` contém o valor da média anual da temperatura, em graus Fahrenheit, em New Haven, Connecticut, de 1912 a 1971. Esta série temporal está disponível na instalação base do **R**.

A Figura 16 mostra gráfico desta série temporal.

```
> plot(nhtemp)
> text(1911, 45.7, "Fonte: Elaborado pelo autor.",
...      xpd = NA, cex = 0.8)
```

Não há uma tendência óbvia, e os dados anuais não parecem ter uma componente sazonal, *assim o modelo exponencial simples* é um início razoável.

A seguir, temos o código para se fazer uma previsão um passo à frente utilizando a função `ses()`.

```
> fit <- ets(nhtemp, model = "ANN")
> fit
ETS(A,N,N)

Call:
 ets(y = nhtemp, model = "ANN")

  Smoothing parameters:
    alpha = 0.1819

  Initial states:
    l = 50.2762

  sigma:  1.1455

     AIC     AICc      BIC
265.9298 266.3584 272.2129
```

```
> (previsao <- forecast(fit, 1))
     Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
1972       51.87031 50.40226 53.33835 49.62512 54.11549
> accuracy(fit)
                    ME     RMSE       MAE       MPE   MAPE      MASE          ACF1
Training set 0.1460657 1.126268 0.8951225 0.2419373 1.7489 0.7512408 -0.006441923
```

A Figura 17 mostra as previsões para o ano de 1972 feita com o modelo ajustado.

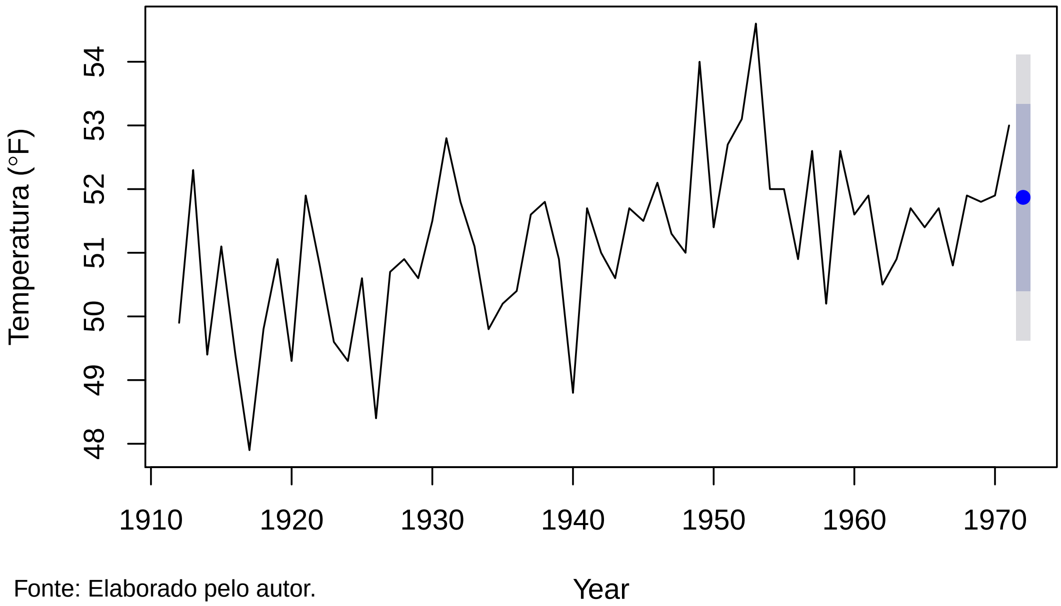
```
> plot(previsao, xlab = "Year", ylab = expression(paste("Temperatura (",
...     degree * F, ")")), main = "")
> text(1911, 45.7, "Fonte: Elaborado pelo autor.", xpd = NA,
...     cex = 0.8)
```

A chamada `ets(mode="ANN")` ajusta o modelo exponencial simples.

A indica que os erros são aditivos, e **NN** indicam que não há componentes de tendência e sazonal.

O valor relativamente baixo de alfa 0.182 indica que tanto valores distantes como observações recentes estão sendo considerados na previsão.

Figura 17: Temperatura Média Anual em New Haven Previsão para 1972



Fonte: Elaborado pelo autor.

Este valor é automaticamente escolhido para maximizar o ajuste do modelo aos dados.

A função `forecast()` é utilizada para prever a série temporal k passos no futuro. O formato é `forecast(fit,k)`. A previsão *um passo à frente* para esta série é 51.87^{o}F, com um intervalo de confiança de 95% de (49.6^{o}F,54.1^{o}F)

O pacote `forecast` também provê a função `accuracy()` que mostra as medidas de acurácia mais populares para previsaõ de séries temporais. Estas medidas e suas respectivas definições são mostradas no Quadro 3.

Quadro 3: Medidas de Acurácia providas pela função `accuracy` do pacote `forecast`

Medida	Abreviação	Definição
Erro médio	ME	mean(e_t)
Raiz quadrada do erro quadrado médio	RMSE	$\sqrt{\text{mean}(e_t^2)}$
Erro absoluto médio	MAE	mean(abs(e_t))
Erro percentual médio	MPE	mean$(100 \times e_t/Y_t)$
Erro percentual absoluto médio	MAPE	mean(abs$(100 \times e_t/Y_t)$)
Erro padronizado absoluto médio	MASE	mean(abs(q_t))**

** $q_t = \dfrac{e_t}{(1/(T-1) \times \sum_{t=2}^{T}(|y_t - y_{t-1}|))}$, onde T é número de observações

Fonte: KABACOFF (2015). Traduzido e adaptado pelo autor.

O *erro médio* (ME) e o *erro percentual médio* (MPE) podem não ser muito úteis, porque os erros negativos e positivos podem se cancelar. O RMSE dá a *raiz quadrada do erro quadrado médio*, que para o modelo ajustado anteriormente é 1.13^{o}F.

O *erro percentual absoluto médio* (MAPE) nos dá o erro como um percentual dos valores da série temporal; ele não tem unidade e pode ser utilizado para comparar a acurácia das predições entre séries temporais. Entretanto, ele assume uma escala de medida com um ponto zero verdadeiro (por exemplo, número de passageiros por dia); como a escala de Fahrenheit não tem um zero verdadeiro, não podemos usar esta medida (MAPE) aqui.

O *erro padronizado absoluto médio* (MASE) é uma medida de acurária recente e é utilizada para comparar previsões entre séries temporais de diferente escalas.

Dentre estas medidas, o RMSE é certamente o mais conhecido e frequentemente citado.

Suavização exponencial Holt e Holt-Winters

Suavização Holt

A suavização exponencial simples, vista anteriormente, assume a ausência de tendência ou componente sazonal; mas nem sempre os dados reais tem este comportamento.

Vamos abordar agora a suavização exponencial Holt, que pode ajustar uma série temporal que tenha um nível (valor inicial) e uma tendência (inclinação). O modelo para uma observação no tempo t é:

$$Y_t = \text{nível} + \text{slope} \times t + \text{irregular}_t$$

Na suavização de Holt, temos:

- um parâmetro de suavização alfa, que controla o decaimento exponencial para o nível,
- um parâmetro de suavização beta, que controla o decaimento exponencial para a inclinação.

Novamente, cada parâmetro varia de 0 a 1, com valores maiores dando mais importância a observações recentes.

Suavização Holt-Winters

A suavização Holt-Winters pode ser utilizada para ajustar uma série temporal que tem um nível, uma tendência e um componente de sazonalidade. O modelo é:

$$Y_t = \text{nível} + \text{slope} \times t + s_t + \text{irregular}_t$$

Onde:

- s_t representa a influência sazonal no tempo t.

Além dos parâmetros alfa e beta, um parâmetro de suavização gama controla o decaimento exponencial da componente sazonal.

Como os outros, este parâmetro gama também varia de 0 a 1, com valores maiores dando mais importância às observações recentes.

Vamos utilizar a base de dados `AirPassengers` novamente e fazer uma suavização Holt-Winters. Utilizando a transformação logaritmica, podemos considerar componentes aditivas no nosso modelo.

```
> fit <- ets(log(AirPassengers), model = "AAA")
> fit
ETS(A,A,A)

Call:
 ets(y = log(AirPassengers), model = "AAA")

  Smoothing parameters:
    alpha = 0.6975
    beta  = 0.0031
    gamma = 1e-04

  Initial states:
    l = 4.7925
    b = 0.0111
    s = -0.1045 -0.2206 -0.0787 0.0562 0.2049 0.2149
           0.1146 -0.0081 -0.0059 0.0225 -0.1113 -0.0841

  sigma:  0.0383
```

```
      AIC      AICc       BIC 
-207.1694 -202.3123 -156.6826 
```

```
> accuracy(fit)
                       ME       RMSE        MAE         MPE      MAPE      MASE       ACF1
Training set -0.001830684 0.03606976 0.02770885 -0.03435608 0.5079142 0.2289192 0.05590461
```

A função `forecast()` produz uma previsão para os próximos cinco meses

```
> pred <- forecast(fit, 5)
> pred
         Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
Jan 1961       6.109335 6.060306 6.158365 6.034351 6.184319
Feb 1961       6.092542 6.032679 6.152405 6.000989 6.184094
Mar 1961       6.236626 6.167535 6.305718 6.130960 6.342292
Apr 1961       6.218531 6.141239 6.295823 6.100323 6.336738
May 1961       6.226734 6.141971 6.311498 6.097100 6.356369
```

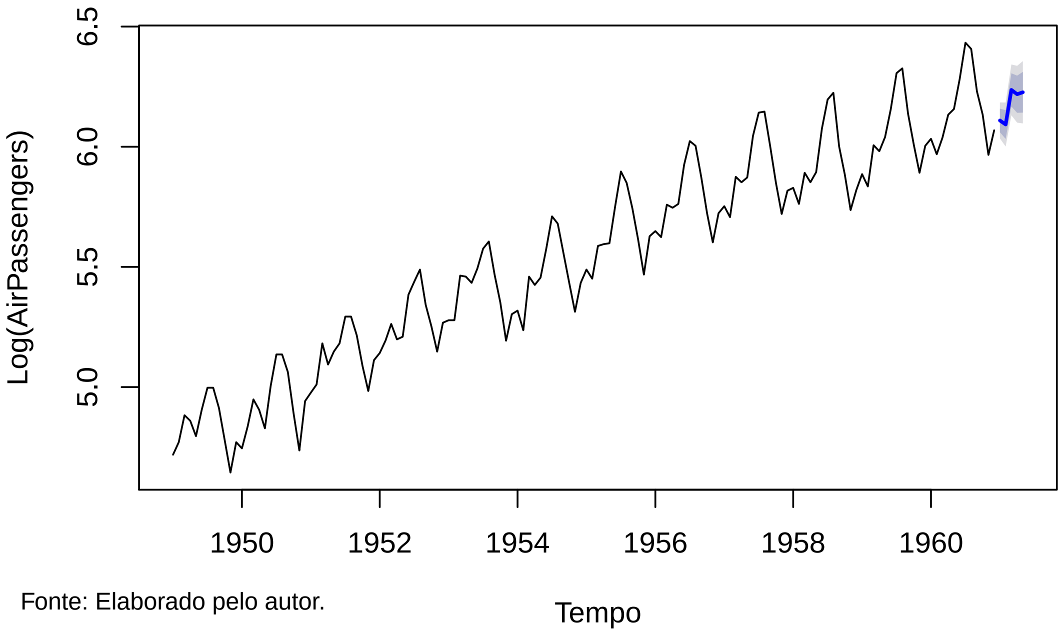
A Figura 18 mostra o gráfico que apresenta a previsão para os próximos cinco meses

```
> plot(pred, ylab = "Log(AirPassengers)", xlab = "Tempo",
...      main = "")
> text(1949, 4.1, "Fonte: Elaborado pelo autor.", xpd = NA,
...      cex = 0.8)
```

Como as previsões estão em uma escala `log`, usamos a exponenciação para obtê-las na sua métrica original.

```
> pred$mean <- exp(pred$mean)
> pred$lower <- exp(pred$lower)
> pred$upper <- exp(pred$upper)
> p <- cbind(pred$mean, pred$lower, pred$upper)
> dimnames(p)[[2]] <- c("mean", "Lo 80", "Lo 95", "Hi 80",
...      "Hi 95")
> p
             mean    Lo 80    Lo 95    Hi 80    Hi 95
Jan 1961 450.0395 428.5065 417.5279 472.6544 485.0826
Feb 1961 442.5448 416.8301 403.8280 469.8459 484.9735
Mar 1961 511.1312 477.0088 459.8775 547.6945 568.0971
Apr 1961 501.9652 464.6289 446.0019 542.3017 564.9506
May 1961 506.1001 464.9691 444.5667 550.8694 576.1504
```

Figura 18: Previsão das Viagens Aéreas – 5 meses de 1961



Fonte: Elaborado pelo autor.

Os parâmetros de suavização obtidos são:

- nível = 0.7,
- tendência = 0.003058
- componente sazonal 1.13×10^{-4}

O valor baixo para a tendência (0.003058), não significa que não haja uma inclinação; indica apenas que a inclinação estimada para as observações iniciais não precisa ser atualizada.

As previsões obtidas indicam 509.696 passageiros em Março, com o intervalo de confiança de 95% variando de 463.099 a 560.981.

A função `ets()` e previsão automática

A função `ets()` tem algumas habilidades adicionais. Podemos utilizá-la para ajustar modelos exponenciais que tenham componentes multiplicativos, adicionar um componente de amortecimento e realizar previsão automatizadas.

Ajustante um modelo multiplicativo

Para ajustar um modelo multiplicativo aos dados `AirPassengers` podemos utilizar as seguintes chamadas na função `ets()`:

- `ets(AirPassengers, model = "MAM")` ou o equivalente
- `ets(AirPassengers, seasonal="multiplicative")`

Há uma função direta quando utilizamos o modelo Holt-Winter – `hw`, e ao chamarmos esta função, indicamos o modo da componente sazonal (aditivo ou multiplicativo)

A componente de tendência permanece aditiva, mas as componentes irregular e sazonal são tratadas como multiplicativas. Como estamos utilizando um modelo sazonal neste caso, a acurácia estatística e os valores previstos estarão na métrica original (milhares de passageiros) – uma boa vantagem.

Ajustando um componente de amortecimento

Previsões em séries temporais frequentemente assumem que a tendência vai continuar para sempre (mercado de imóveis, lembram?). Para contrapor esta previsão, um componente de amortecimento força a tendência a uma assíntota horizontal após um período de tempo. Em muitos casos, um modelo de mercado com amortecimento faz previsões mais realistas.

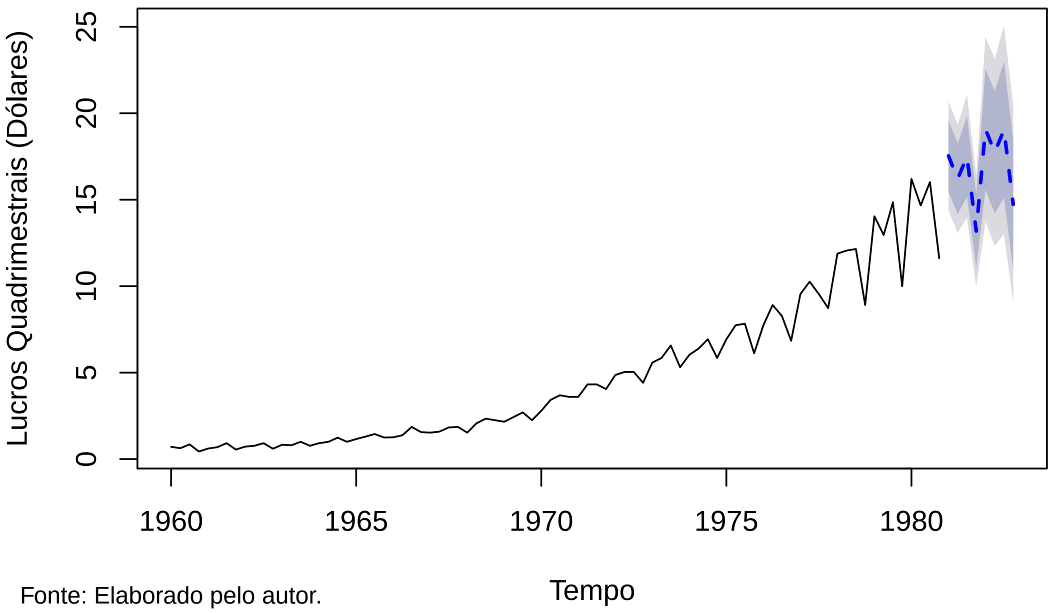
Ajuste automatizado

A função `ets()` também pode ser invocada para automaticamente selecionar o modelo que melhor se ajusta aos dados.

Previsão automática exponencial com `ets()`

```
> fit <- ets(JohnsonJohnson)
> fit
ETS(M,A,A)

Call:
 ets(y = JohnsonJohnson)
```

Figura 19: Previsões para Johnson & Johnson




```
  Smoothing parameters:
    alpha = 0.2776
    beta  = 0.0636
    gamma = 0.5867

  Initial states:
    l = 0.6276
    b = 0.0165
    s = -0.2293 0.1913 -0.0074 0.0454

  sigma:  0.0921

     AIC     AICc      BIC
163.6392 166.0716 185.5165
```

A Figura 19 mostra o gráficos das previsões.

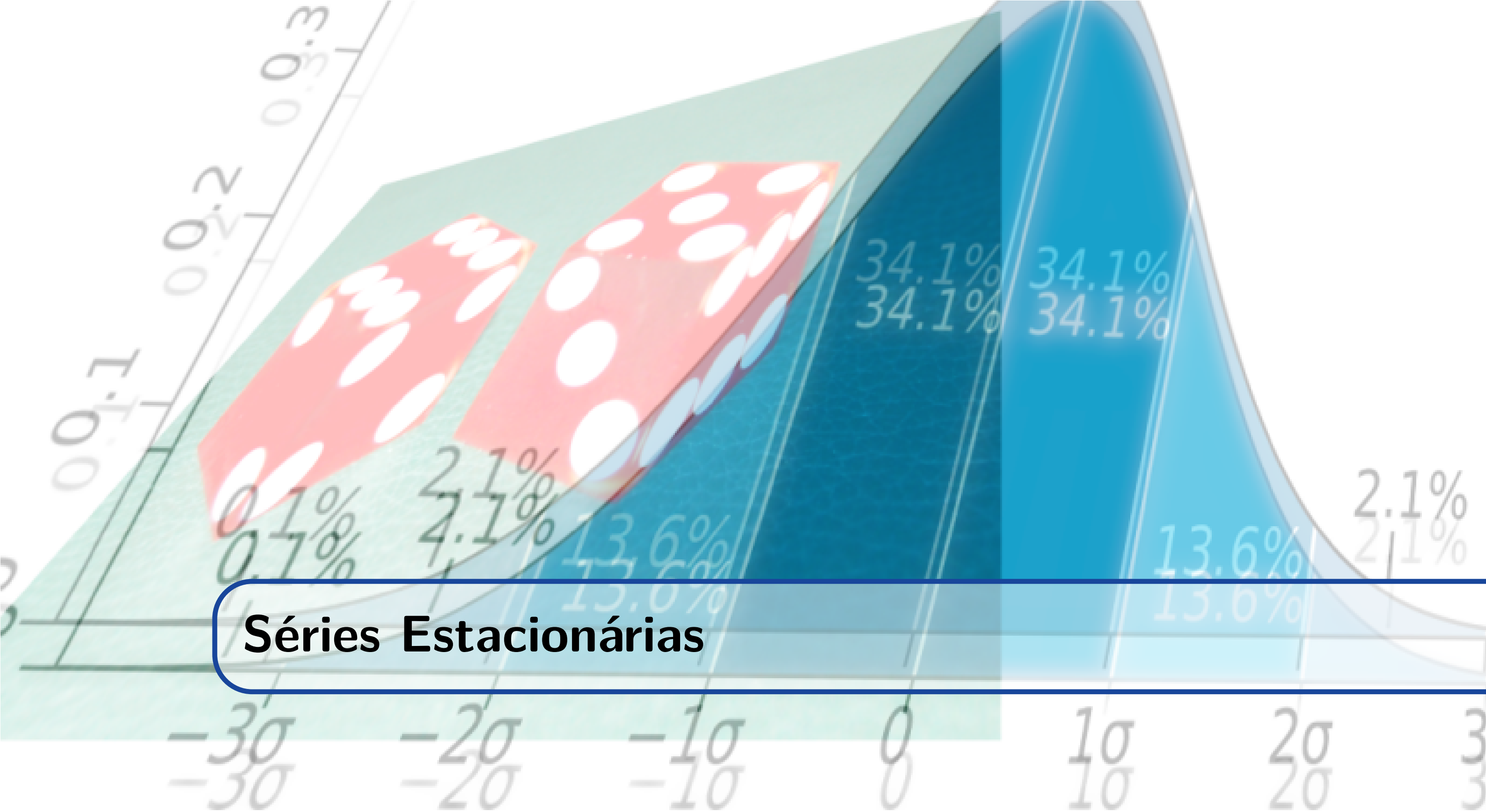
```
> plot(forecast(fit), main = "", ylab = "Lucros Quadrimestrais (Dólares)",
...      xlab = "Tempo", flty = 2)
> text(1960, -8, "Fonte: Elaborado pelo autor.", xpd = NA,
...      cex = 0.8)
```

Como não especificamos um modelo, o software realiza uma busca em uma grande gama de modelos para encontrar aquele que minimiza o critério de ajuste (*log-likelihood* por default).

O modelo selecionado tem uma componente de tendência multiplicativa, uma componente sazonal aditiva e uma componente de erro multiplicativa (`ETS(M,A,M)`).

O gráfico, juntamente com a previsão para os próximos 8 quadrimestres (quando não informamos o número de períodos de previsão, se o objeto tem frequência maior que um, o padrão é $2\times$`frequency(object)`, ou seja, $2 \times 4 = 8$) é mostrado na Figura 19. O parâmetro `flty` define o tipo da linha para a previsão – tracejada, neste caso.

II Modelagem de Séries Temporais

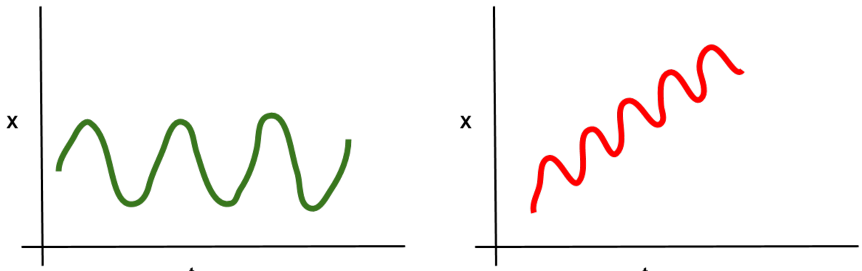


Séries Estacionárias

Há três critérios básicos para uma série ser classificada como série estacionária:

1. A média da série não deve ser uma função do tempo, mas deve, ao invés, ser constante. Na Figura 20, o gráfico da esquerda satisfaz esta condição enquanto que o da direita (vermelho) tem uma média dependente do tempo.

Figura 20: Exemplo de Série Estacionária e Não Estacionária

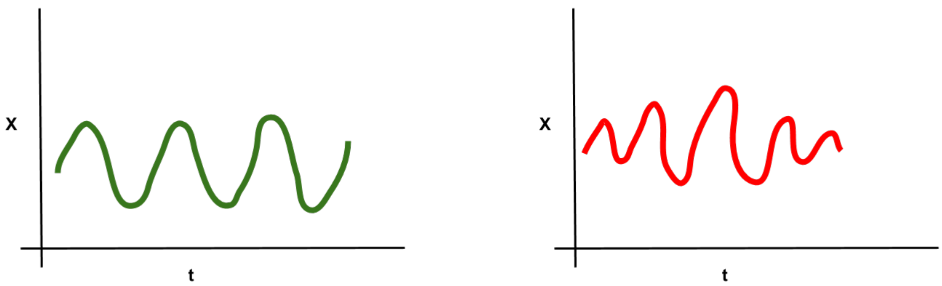


(a) Série Estacionária (b) Série Não Estacionária

Fonte: Elaborado pelo autor

2. A variância da série **não** deve ser uma função do tempo. Esta propriedade é conhecida como homocesdasticidade. A Figura 21 mostra o que é o que não é uma série estacionária (Perceba a dispersão variável da distribuição no gráfico da direita).

Figura 21: Exemplo de Série Estacionária e Não Estacionária

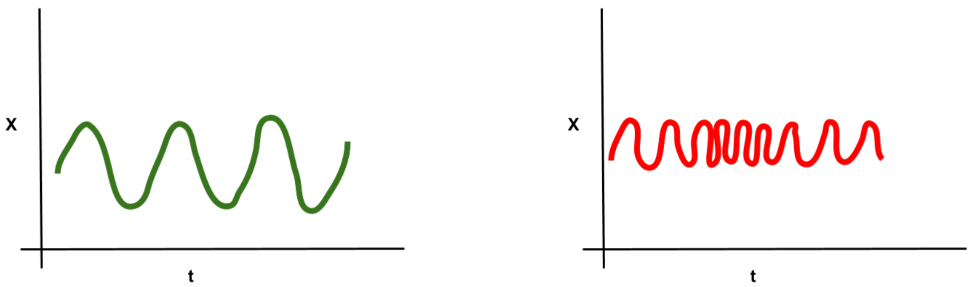


(a) Série Estacionária (b) Série Não Estacionária - variância não constante

Fonte: Elaborado pelo autor

3. A covariância do i-th termo e do $(i+m)$ termo não deve ser uma função do tempo. Na Figura 22 percebemos que a dispersão se torna mais próxima conforme o tempo aumenta. Portanto, a covariância não é constante com o tempo para a série **vermelha**.

Figura 22: Exemplo de Série Estacionária e Não Estacionária



(a) Série Estacionária (b) Série Não Estacionária - covariância função do tempo

Fonte: Elaborado pelo autor

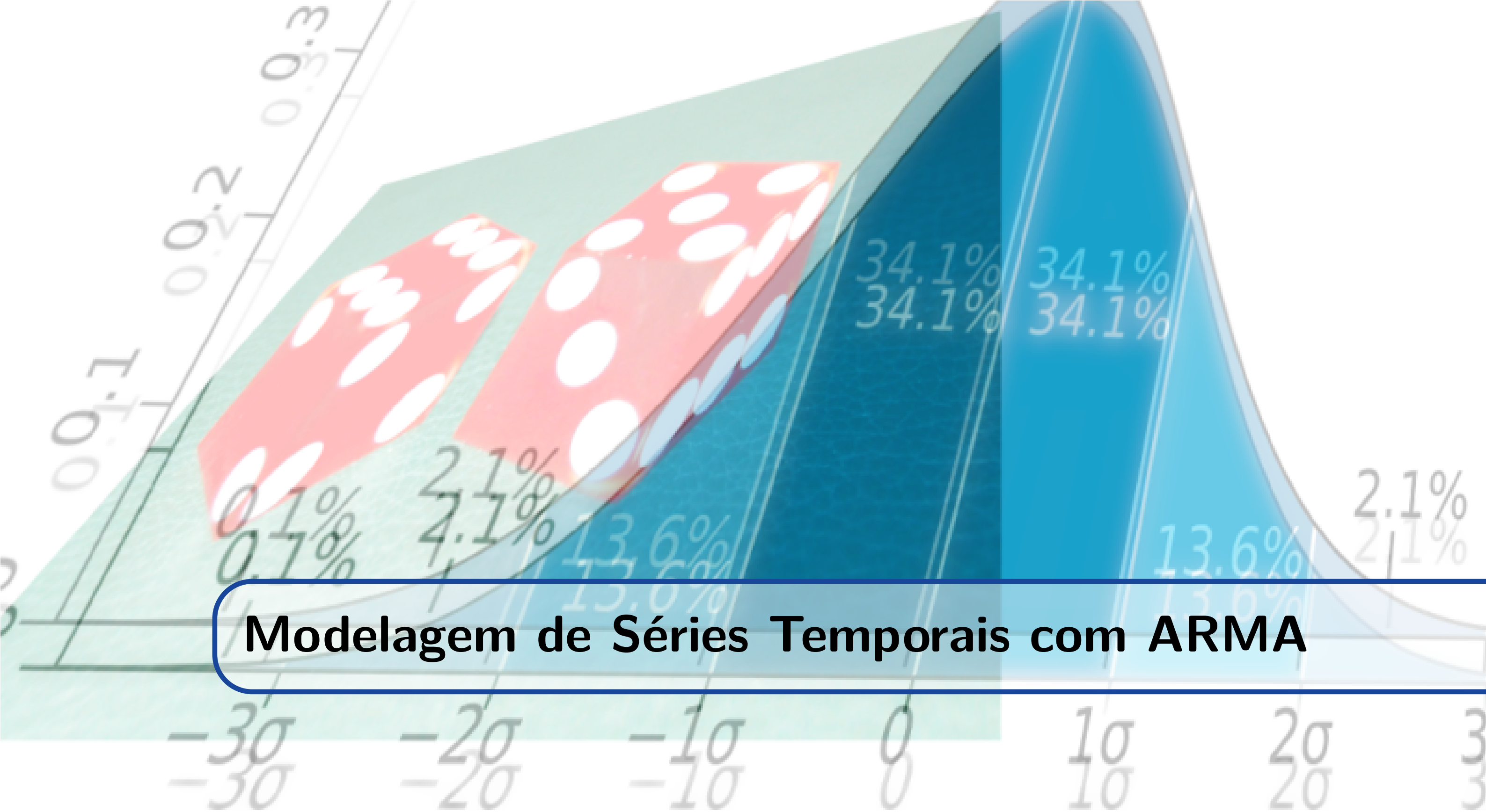
!

Por que nos preocupamos com a estacionariedade de uma série temporal?

A razão é que **a menos que nossa série temporal seja estacionária, não podemos construir um modelo de série temporal**.

Neste caso, quando o critério estacionário é violado, o primeiro requisito é estacionarizar a série temporal e então tentar algum modelo estocástico para predizer esta série temporal.

Há várias maneiras de se trazer esta estacionariedade. Alguns deles são *Detrending*, Diferenciando, etc.



Modelagem de Séries Temporais com ARMA

Os modelos ARMA são comumente utilizados na modelagem de séries temporais. Em um modelo ARMA, **AR** significa **auto-regressão** e **MA** significa **moving average** (média móvel).

Estes modelos são aplicáveis a séries estacionárias. Se temos uma série não estacionária, nossa primeira tarefa é estacionarizá-la (através de diferenças/transformações) e só então escolher um modelo dentre os disponíveis para séries temporais (SRIVASTAVA (2015)).

Vamos explicar cada um destes dois modelos (AR e MA) individualmente.

Modelo Auto-Regressivo

Exemplo 1

O PIB (GDP) atual de um país (dado por $x(t)$) é dependente do PID do último ano, i.e. $x(t-1)$.

A hipótese é de que o custo total de produção de produtos e serviços em um país em um ano fiscal (conhecido como PIB) é depedente do arranjo de fábricas/serviços no ano anterior e das novas indústrias, fábricas, serviços do ano atual. Mas o componente primário do PIB é o anterior.

Podemos escrever formalmente isso como:

$$x(t) = alpha * (x - 1) + error(t)$$

Esta equação é conhecida como *formulação AR(1)*. O numeral um (1) denota que a próxima instância somente depende da instância anterior.

O *alpha* é o coeficiente que procuramos para minimizar a função erro.

Notamos que $x(t-1)$ está ligado a $x(t-2)$ do mesmo modo. Portanto, qualquer impacto em $x(t)$ vai desaparecer gradualmente no futuro.

Exemplo 2

Por exemplo, digamos que $x(t)$ seja o número de garrafas de suco vendidas em uma cidade em um dia particular. Durante os invernos, poucos vendedores compraram garrafas de suco. De repente, em um dia particular, a temperatura aumentou e a demanda por garrafas de suco disparou a 1000.

Contudo, após alguns dias, a clima se torna frio novamente. Contudo, algumas pessoas (*digamos* 50%) continuam tomando suco durante os dias frios.

Nos dias seguintes, esta proporção diminui para 25% (ou seja, 50% de 50%) e então, gradualmente, para um número pequeno após um número significante de dias.

O gráfico mostrado na Figura 23 ilustra a propriedade inercial da série AR.

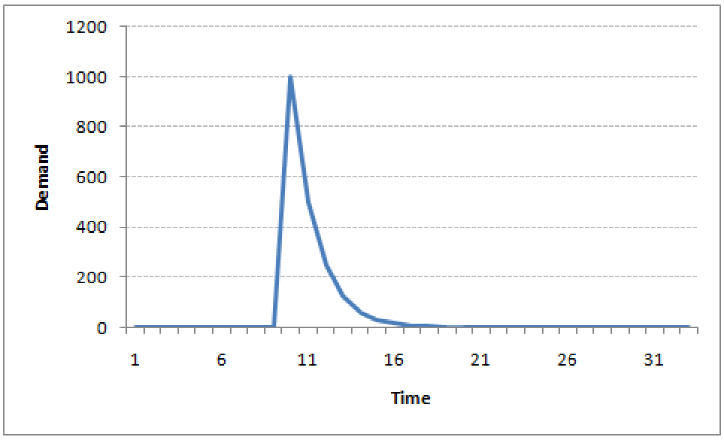
Modelo de Média Móvel – *Moving Average*

Exemplo

Um produtor produz um certo tipo de sacola, que ficou prontamente disponível no mercado. Sendo um mercado competitivo, a venda da sacola permaneceu em zero por muitos dias.

Assim, um dia ele fez alguns experimentos com o projeto e produziu um tipo diferente de sacola. Este tipo de sacola não esteve amplamente disponível no mercado.

Figura 23: Série com propriedade inercial - Modelo Auto-Regressivo



Fonte: SRIVASTAVA (2015).

Com isso, ele foi capaz de vender o estoque inteiro de 1000 sacolas (vamos chamar isso de $x(t)$).

A demanda foi tão alta que as sacolas sumiram das lojas. Como resultado, por volta de 100 infelizes compradores não puderam adquirir esta sacola. Vamos chamar este *gap* de "erro" naquele ponto de tempo.

Com o tempo, a sacola perdeu o fator novidade, mas alguns poucos clientes restaram que ainda ficaram sem a sacola no dia anterior.

Uma formulação simples deste cenário é:

$$x(t) = beta * error(t-1) + error(t)$$

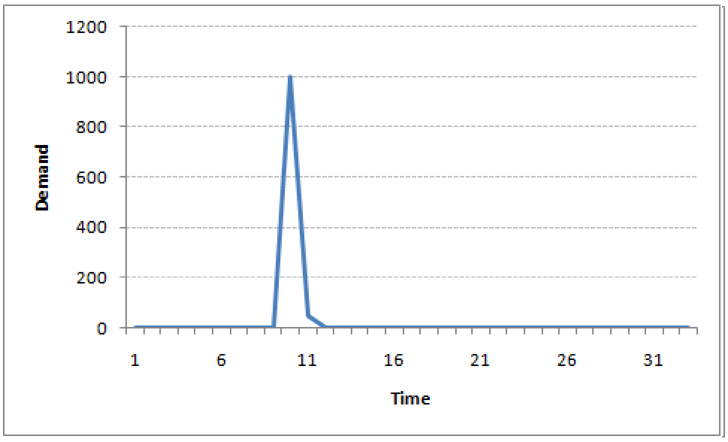
A Figura 24 apresenta um gráfico que mostra este efeito.

Diferenças entre os Modelos AR e MA

Percebeu a diferença entre os modelos **MA** e **AR**?

- No Modelo MA, o ruído/impacto rapidamente desaparece com o tempo.
- O Modelo AR tem um efeito muito mais duradouro do impacto.

Figura 24: Série do Modelo *Moving Average*, rápido desaparecimento do pulso



Fonte: SRIVASTAVA (2015).

A diferença primária entre os modelos AR e MA está baseada na correlação entre os objetos de série temporais em direntes pontos no tempo. A correlação entre $x(t)$ e $x(t-1)$ para $n >$ ordem de MA é sempre zero!

Isto segue diretamente do fato que a covariância entre $x(t)$ e $x(t-1)$ é zero para os modelos MA. Contudo a correlação de $x(t)$ e $x(t-1)$ gradualmente declina com n se tornando maior no modelo AR. Esta diferença é explorada independentemente de se ter um modelo AR ou um modelo MA. O gráfico da correlação nos dá a ordem do Modelo MA.

Explorando gráficos ACF e PACF

Uma vez que temos séries estacionárias, temos que responder duas questões primárias:

- Q1. É um processo AR ou MA?
- Q2. Qual ordem do processo AR ou MA precisamos usar?

A dica para resolver estas equações foi discutida anteriormente. Vamos esclarecer.

A primeira questão pode ser respondida utilizando o gráfico da Correlação Total (também conhecido como Função de Auto-Correlação / ACF). ACF é um gráfico da correlação total entre diferentes funções de intervalos (*lag functions*).

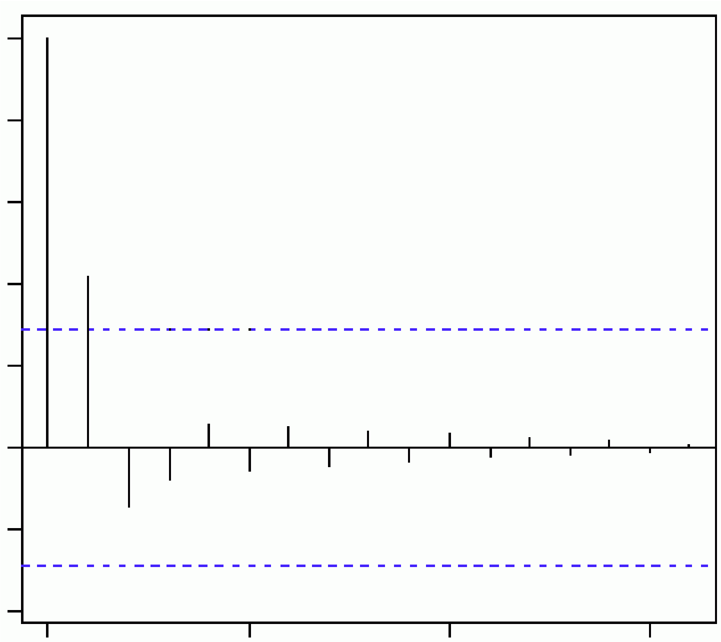
Por exemplo, no problema do PIB, o PIB em um ponto de tempo é $x(t)$. Estamos interessados na correlação de $x(t)$ com $x(t-1), x(t-2), \ldots$ — vamos refletir sobre isso.

Processo de Média Móvel (MA)

Em uma série de médias móveis de lag n, nós não teremos nenhuma correlação entre $x(t)$ e $x(t-1)$. Portanto, o gráfico de correlação total será interrompido (*cut off*) no n-th lag. Assim, fica simples encontrar o *lag* para uma série MA.

Veja a ilustração deste efeito no gráfico de ACF mostrado na Figura 25.

Figura 25: Efeito de Auto-Correlação total interrompido rapidamente –**ACF**

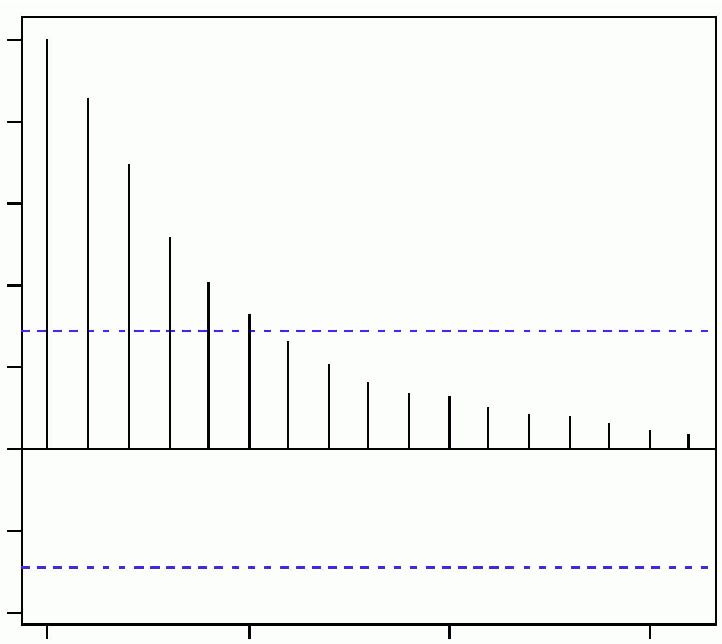


Fonte: SRIVASTAVA (2015).

Claramente, o gráfico mostrado na Figura 25 tem um corte na curva de ACF após o segundo lag, o que significa que provavelmente é um processo MA(2).

Para uma série MA, a função de correlação parcial (PACF) vai gradualmente diminuindo sem um valor de corte. A Figura 26 ilustra esta situação.

Figura 26: Efeito de Auto-Correlação Parcial com declínio gradual – **PACF**



Fonte: SRIVASTAVA (2015).

Processo Auto-Regressivo (AR)

Assim, como fazemos para uma série AR?

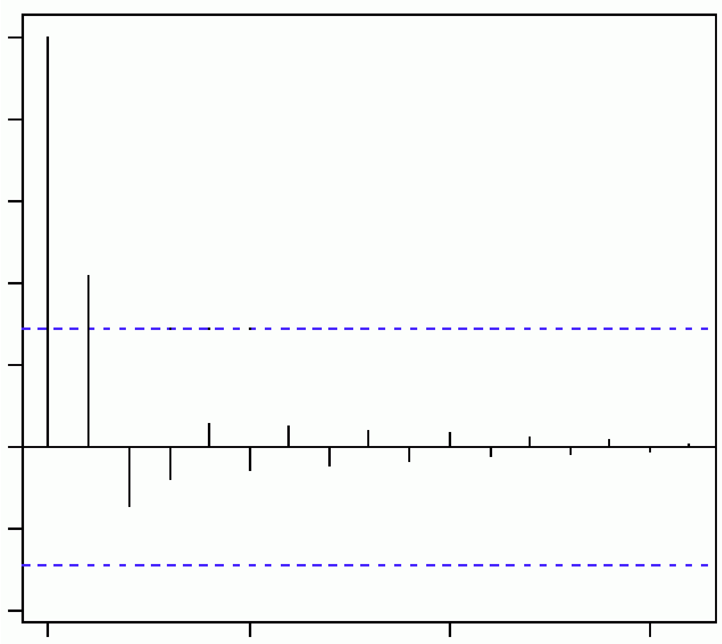
Aqui está a segunda dica:

- Se encontramos a correlação parcial de cada lag, ele vai ser cortado após o grau da série AR.

Por exemplo, se temos uma série AR(1), se excluirmos o efeito do primeiro lag ($x(t-1)$), nosso segundo lag ($x(t-2)$) é independente de $x(t)$. Assim, a função de correlação parcial (PACF) vai diminuir bruscamente após o primeiro lag, enquanto a função de correlação total (ACF) terá um declínio lento.

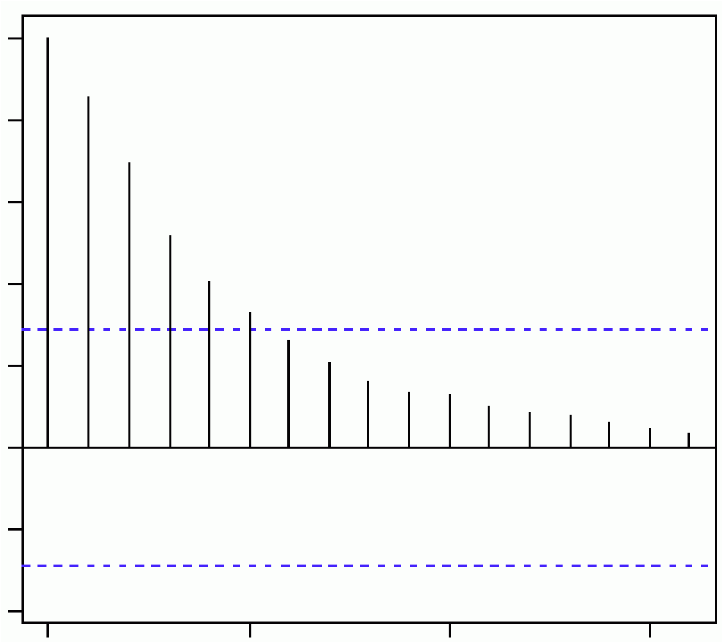
Os gráficos mostrados nas Figuras 27 e 28 ilustram estes conceitos; a linha azul nos gráficos mostra valores significantemente diferente de zero. Observamos um corte na curva de PACF após o segundo lag, o que significa que provavelmente é um processo AR(2).

Figura 27: Efeito de Auto-Correlação Parcial com rápido declínio – **PACF**

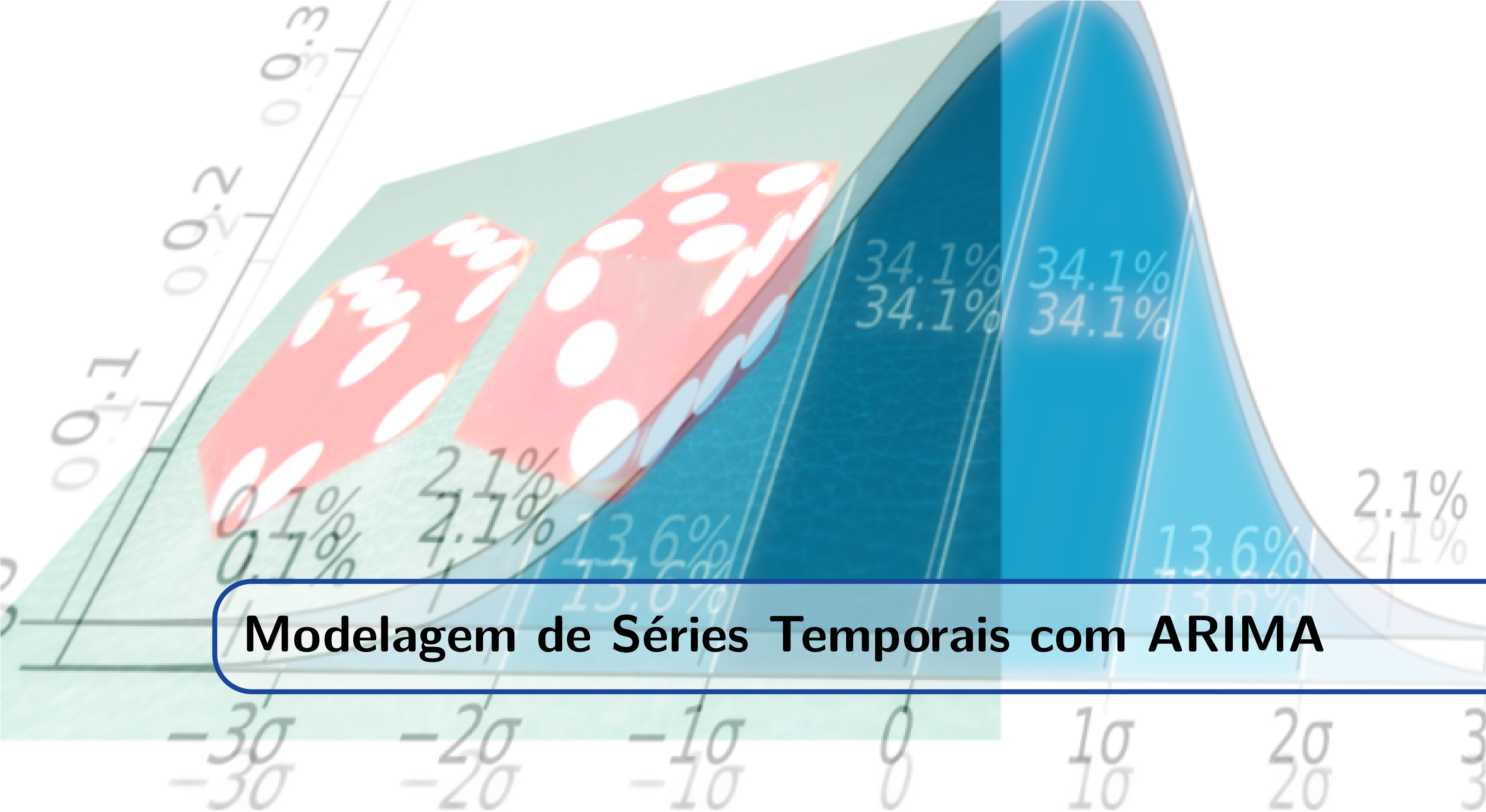


Fonte: SRIVASTAVA (2015).

Figura 28: Efeito de Auto-Correlação Total com declínio gradual – **ACF**



Fonte: SRIVASTAVA (2015).



Modelagem de Séries Temporais com ARIMA

A Modelagem de Séries Temporais com ARIMA pode ser entendida, ou abordada, a partir das seguintes etapas:

1. **Visualizar a Série Temporal**

2. **Estacionarizar a Série**

3. **Encontrar os Parâmetros Ótimos**

4. **Construir o Modelo ARIMA**

5. **Fazer Predições**

Passo 1 – Visualizar a Série Temporal

É essencial para analisar as tendências antes de se construir qualquer tipo de modelo da série temporal.

Os detalhes que estamos interessados são qualquer tipo de tendência, sazonalidade, ou comportamento aleatório nas séries.

Passo 2 – Estacionarizar a Série

Uma vez que conhecemos os padrões, tendências, ciclos e sazonalidade, podemos verificar se a série é estacionária ou não. Um teste popular é o **Dickey–Fuller**

Se a série não é estacionária, temos três técnicas comumente utilizadas para torná-la uma série estacionária.

1. **Detrending** (removendo tendência). O objetivo é simplesmente remover o componente de tendência da série temporal. Por exemplo, a equação de uma série pode ser:

$$x(t) = (mean + trend * t) + error$$

Simplesmente removemos a parte nos parênteses e construimos um modelo para o restante.

2. **Diferenciando**. Esta é a técnica comumente utilizada para remover a não estacionariedade. Aqui, tentamos modelar as diferenças dos termos e não os termos propriamente. Por exemplo:

$$x(t) - x(t-1) = ARMA(p, q)$$

   Esta diferenciação é a **Integração** no modelo AR(I)MA.

Agora temos três parâmetros:

- **p**: AR
- **d**: I
- **q**: MA

3. **Sazonalidade**. Sazonalidade pode ser facilmente incorporada no modelo ARIMA diretamente, como veremos brevemente a seguir.

Passo 3 – Encontrar os Parâmetros Ótimos

O parâmetros p, d, q podem ser encontrados utilizando os gráficos ACF e PACF.

Uma abordagem adicional pode ser, se tanto ACF como PACF diminuem gradualmente, isto indica que precisamos fazer a série temporal estacionária e introduzir um valor para d.

Passo 4 – Construir o Modelo ARIMA

Com os parâmetros em mãos, podemos tentar construir o modelo ARIMA.

Os valores encontrados acima podem ser uma estimativa aproximada e nós podemos ter que explorar mais combinações de (p,d,q).

Aquele com os menores indicadores de BIC e AIC poderão ser nossa escolha.

Também se pode tentar alguns modelos com um componente sazonal, caso se perceba alguma sazonalidade nos gráficos de ACF/PACF.

Passo 5 – Fazer Predições

Com o modelo ARIMA final, estamos prontos então para fazer predições de futuros pontos de tempo. Também podemos visualizar as tendências para fazer validação cruzada e ver se o modelo funciona bem.

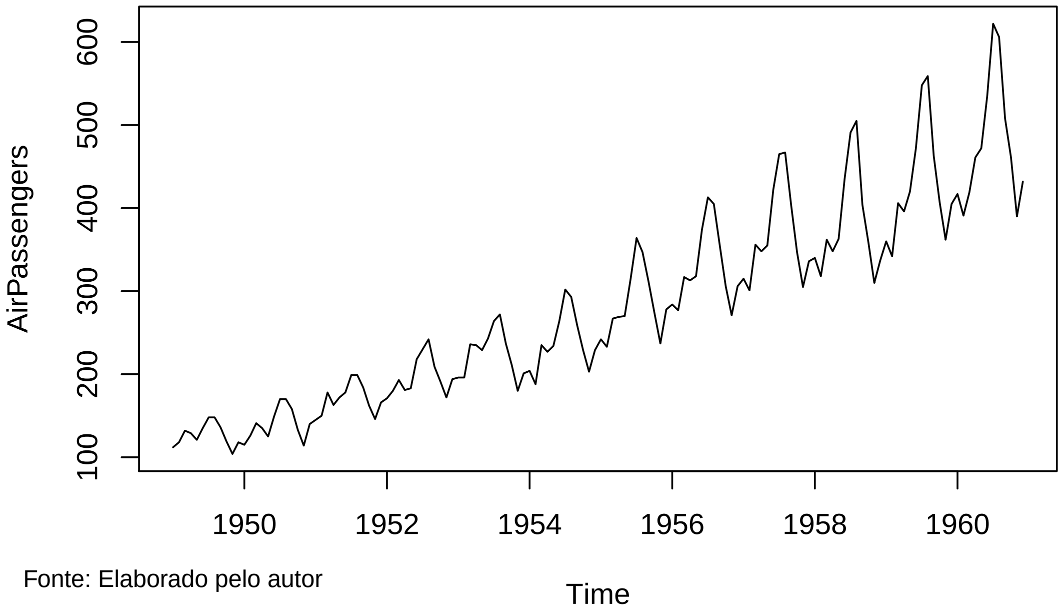
Aplicando o Modelo ARIMA

Vamos retomar a série AirPassengers; a Figura 11 mostra a série.

```
> plot(AirPassengers)
> text(1949, -49, "Fonte: Elaborado pelo autor", xpd = NA,
...     cex = 0.8)
```

Algumas observações

Figura 29: Série temporal AirPassengers



Fonte: Elaborado pelo autor

1. Como já detectamos anteriormente, parece haver uma componente de tendência que faz o número de passageiros crescer ano após ano.
2. Da mesma forma, parece haver uma componente sazonal que tem um ciclo menor do que 12 meses.
3. A variância nos dados se mantem aumentando com o tempo.

Precisamos abordar estes pontos antes de testar a série estacionária.

Primeiro, precisamos remover a desigualdade da variância. Fazemos isso, utilizando o $\log$ da série.

Segundo, precisamos abordar a componente de tendência. Fazemos isso, pegando a diferença da série.

Aplicando

Utilizamos o teste de Dickey-Fuller

```
> library(tseries)
> adf.test(diff(log(AirPassengers)), alternative = "stationary",
```

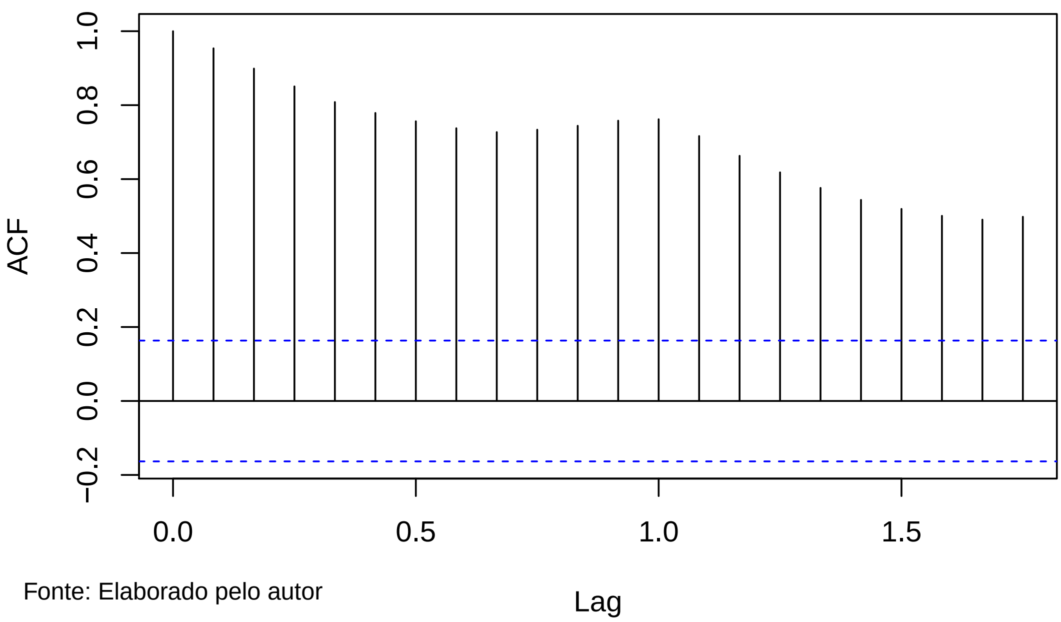
```
...     k = 0)

    Augmented Dickey-Fuller Test

data:  diff(log(AirPassengers))
Dickey-Fuller = -9.6003, Lag order = 0, p-value = 0.01
alternative hypothesis: stationary
```

Podemos ver que nossa série é estacionária o bastante para fazermos qualquer tipo de modelagem de série temporal.

O próximo passo é encontrar os parâmetros corretos para usarmos no modelo ARIMA. Já sabemos que a componente "d" é 1, já que foi necessário aplicar uma diferença (função `diff`) para tornar a série estacionária. Vamos usar os gráficos de ACF para a série, conforme mostra a Figura 30:

```
> acf(log(AirPassengers), main = "")
> text(0, -0.52, "Fonte: Elaborado pelo autor", xpd = NA,
...     cex = 0.8)
```

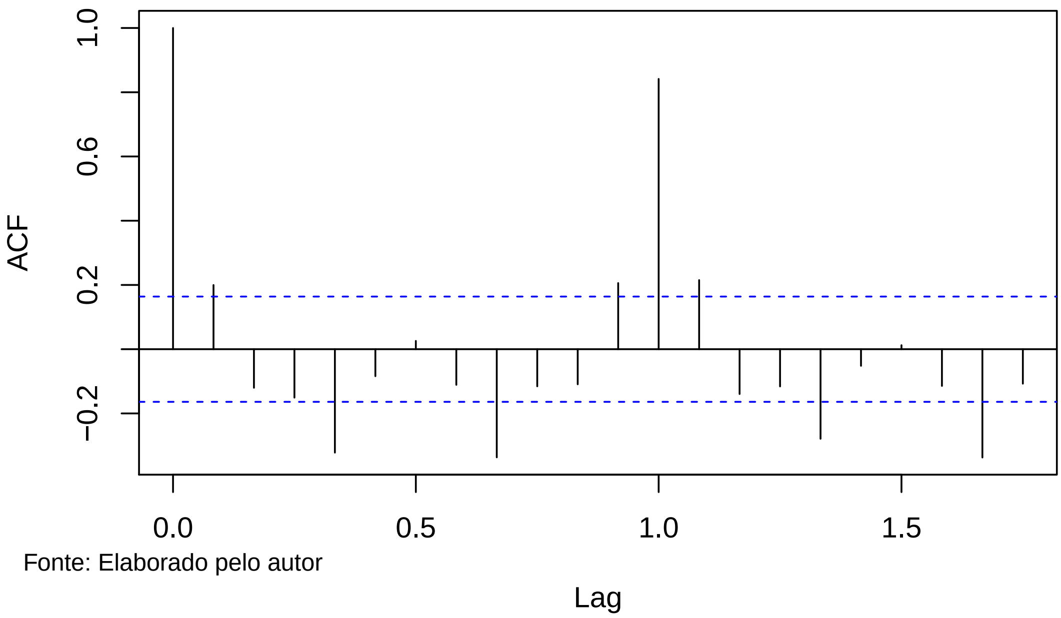
Figura 30: Gráfico ACF para a série Air Passengers



Claramente, o decaimento do gráfico de ACF mostrado na Figura 30 é bem lento, o que significa que tomando **apenas** o log da população nossa série não é estacionária. Mas, como dissemos, pretendemos fazer uma regressão sobre as **diferenças dos logs** ao invés de diretamente sobre o log. Vamos ver, então como os gráficos de ACF (Figura 31) e PACF (Figura 32) ficam com as diferenças dos logs e a regressão.

```
> acf(diff(log(AirPassengers)), main = "")
> text(0, -0.67, "Fonte: Elaborado pelo autor", xpd = NA,
...      cex = 0.8)
```

Figura 31: ACF da Diferenças dos logs da série Air Passengers



Fonte: Elaborado pelo autor

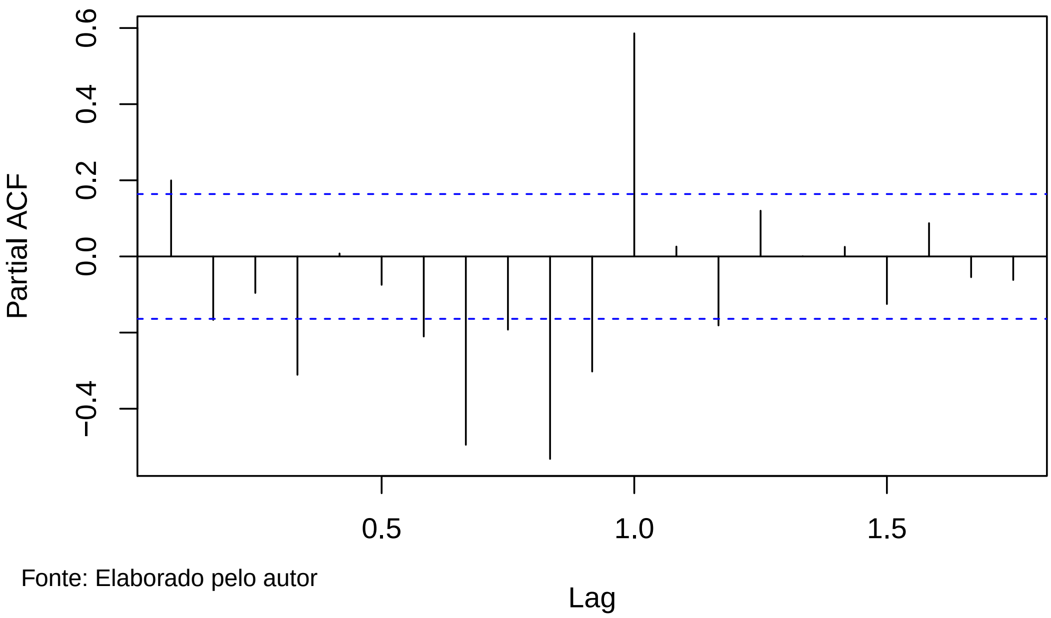
```
> pacf(diff(log(AirPassengers)), main = "")
> text(0.08, -0.85, "Fonte: Elaborado pelo autor", xpd = NA,
...      cex = 0.8)
```

Claramente, o gráfico de ACF corta após o primeiro lag. Portanto, entendemos que o valor de p dever ser 0 (não detectamos auto-correlação parcial), já que a curva de ACF é a que está tendo um corte – a curva de PACF não apresenta um padrão de corte, apresentando oscilações muito próximas ao valor significativo limite (linha tracejada azul). Já o valor de q poderia ser 1 ou 2.

Para saber qual destes, fazemos alguns testes de modelo e pegamos o valor que resulta nos menores AIC e BIC:

```
> AIC(arima(log(AirPassengers), c(0, 1, 1)), arima(log(AirPassengers),
...      c(0, 1, 2)))
                                     df       AIC
arima(log(AirPassengers), c(0, 1, 1))  2 -238.7254
arima(log(AirPassengers), c(0, 1, 2))  3 -240.8465
```

Figura 32: PACF da Diferenças dos logs da série Air Passengers



Fonte: Elaborado pelo autor

```
> BIC(arima(log(AirPassengers), c(0, 1, 1)), arima(log(AirPassengers),
...       c(0, 1, 2)))
                                    df       BIC
arima(log(AirPassengers), c(0, 1, 1))  2 -232.7997
arima(log(AirPassengers), c(0, 1, 2))  3 -231.9580
```

Pelas comparações acima, temos:

- AIC:
  - o melhor modelo é (0,1,2): AIC = -240.8465, com 3 graus de liberdade;
- BIC:
  - o melhor modelo é (0,1,1): BIC = -232.7997, com 2 graus de liberdade.

Considerando que os índices de comparação não convergiram para o mesmo modelo, precisaremos tomar uma decisão, que acaba sendo um pouco arbitrária – escolheremos o melhor modelo (menor AIC ou BIC) com o menor número de graus de liberdade:

- O resultado é (0,1,1) como (p,d,q)

O pacote `forecast` tem outra função que ajuda a verificar se nossa decisão de escolher o modelo (0,1,1) é a melhor ou não; o nome da função é `auto.arima`, e ela faz uma busca do melhor modelo ARIMA de acordo com um dos critérios AIC, AICc ou BIC.

```
> lAirPassengers = log(AirPassengers)
> auto.arima(lAirPassengers, stepwise = FALSE, approximation = FALSE)
Series: lAirPassengers
ARIMA(0,1,1)(0,1,1)[12]

Coefficients:
          ma1     sma1
      -0.4018  -0.5569
s.e.   0.0896   0.0731

sigma^2 estimated as 0.001371:  log likelihood=244.7
AIC=-483.4   AICc=-483.21   BIC=-474.77
```

Do resultado da função `auto.arima` concluimos que o melhor modelo é realmente (0,1,1).

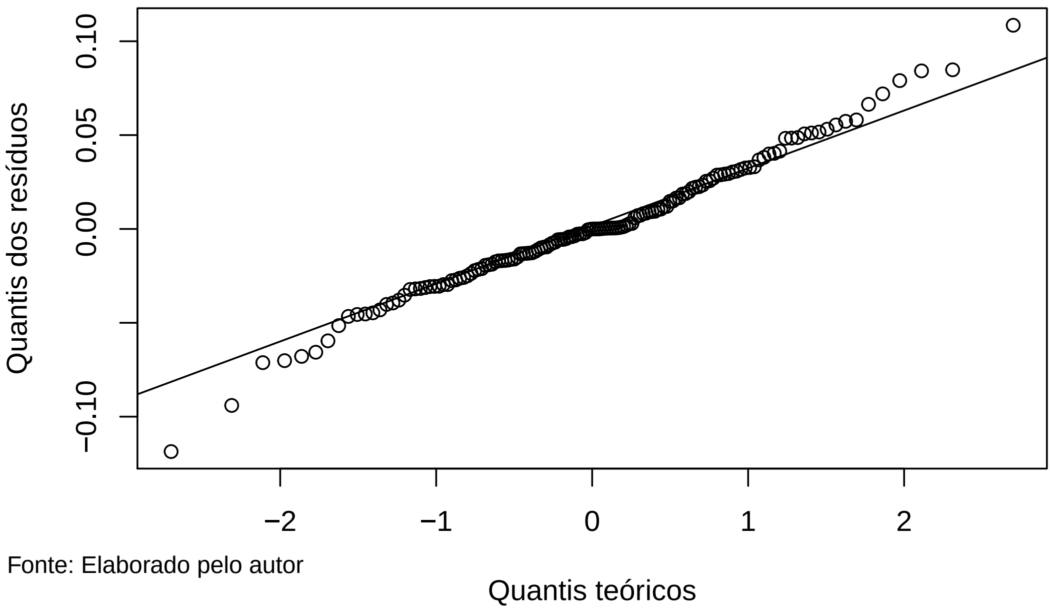
Agora podemos ajustar nosso modelo ARIMA e predizer o futuro em 10 anos. Vamos tambem tentar ajustar uma componente sazonal na formulação do modelo ARIMA e depois vamos visualizar os dados.

```
> (fit <- arima(log(AirPassengers), c(0, 1,
...     1), seasonal = list(order = c(0, 1, 1),
...     period = 12)))

Call:
arima(x = log(AirPassengers), order = c(0, 1, 1), seasonal = list(order = c(0,
    1, 1), period = 12))

Coefficients:
          ma1     sma1
      -0.4018  -0.5569
s.e.   0.0896   0.0731

sigma^2 estimated as 0.001348:  log likelihood = 244.7,  aic = -483.4
```

Antes de prosseguirmos para as predições, precisamos avaliar nosso modelo. Se ele for apropriado, os resíduos devem ter uma distribuição normal, com média zero, e as autocorrelações devem ser zero para todos os *lags*, isto é, resíduos devem ter distribuição normal e ser independentes. Podemos avaliar se estes critérios foram atingidos com os seguintes comandos.

```
> qqnorm(fit$residuals, main = "", xlab = "Quantis teóricos",
...      ylab = "Quantis dos resíduos")
> qqline(fit$residuals)
> text(-2.8, -0.18, "Fonte: Elaborado pelo autor", xpd = NA, cex = 0.8)
```

Figura 33: QQ-plot




```
> Box.test(fit$residuals, type = "Ljung-Box")

	Box-Ljung test

data:  fit$residuals
X-squared = 0.030651, df = 1, p-value = 0.861
```

Como já visto anteriormente, quando os dados seguem uma distribuição normal, eles devem se situar ao longo da linha do gráfico acima; neste caso, o resultado é satisfatório.

A função `Box.test` provê um teste para verificar se todas as autocorrelações são zero. O resultado, observando-se o valor do `p-value` do teste indica que não podemos rejeitar a hipótese nula de que os valores das autocorrelação sejam zero, o que indica que o modelo ARIMA parece ajustar bem os dados.

Além dos testes acima, também devemos analisar a acurácia do nosso modelo.

```
> accuracy(fit)
                       ME       RMSE        MAE        MPE      MAPE      MASE       ACF1
Training set 0.0005730622 0.03504883 0.02626034 0.01098898 0.4752815 0.2898953 0.01443892
```

Pelo resultado acima, podemo ver que o modelo obteve realmente bons resultados no conjunto de dados de treinamento. O RMSE, que é um valor amplamente utilizado para avaliar modelos, apresenta um valor de 0.035; o Erro Percentual Absoluto Médio (MAPE) apresenta um valor de 0.48%, que também é um valor muito bom.

Desta forma, podemos ter uma melhor confiança nas predições que serão feitas. Iniciamos listando os dados previstos e depois visualizaremos o gráfico das previsões.

Limitaremos a previsão numérica em 3 anos por questão de economia de espaço.

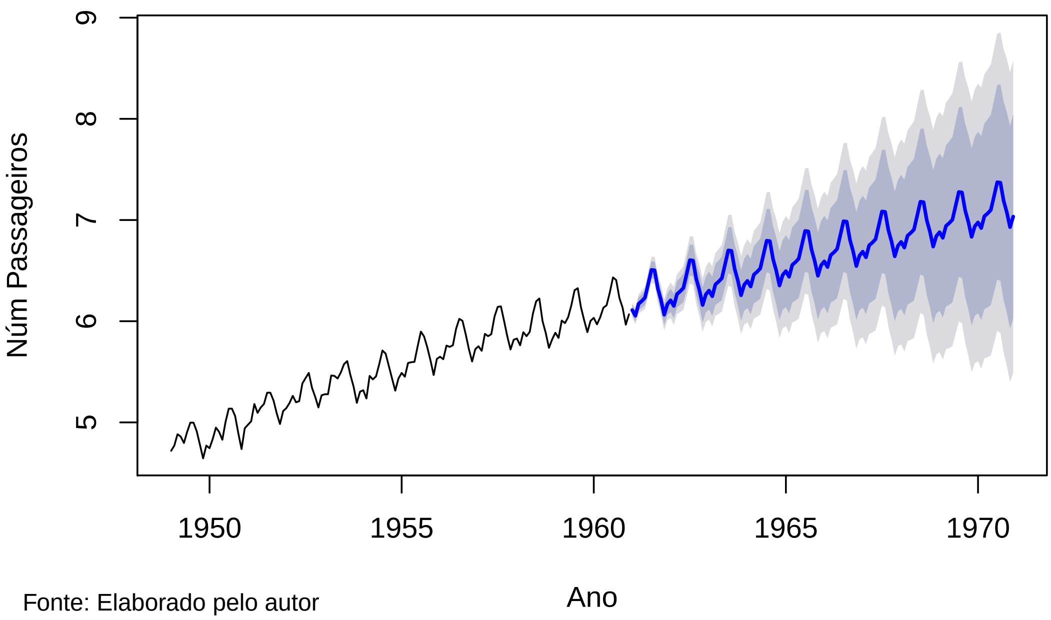
```
> forecast(fit, 3 * 12)
         Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
Jan 1961       6.110186 6.063133 6.157239 6.038224 6.182147
Feb 1961       6.053775 5.998947 6.108604 5.969922 6.137628
Mar 1961       6.171715 6.110084 6.233346 6.077459 6.265971
Apr 1961       6.199300 6.131547 6.267054 6.095680 6.302920
May 1961       6.232556 6.159189 6.305923 6.120351 6.344761
Jun 1961       6.368779 6.290198 6.447359 6.248600 6.488957
Jul 1961       6.507294 6.423825 6.590763 6.379639 6.634949
Aug 1961       6.502906 6.414820 6.590993 6.368189 6.637623
Sep 1961       6.324698 6.232224 6.417172 6.183271 6.466125
Oct 1961       6.209008 6.112346 6.305670 6.061176 6.356841
Nov 1961       6.063487 5.962811 6.164164 5.909516 6.217459
Dec 1961       6.168025 6.063488 6.272562 6.008149 6.327901
Jan 1962       6.206435 6.090987 6.321883 6.029872 6.382998
Feb 1962       6.150025 6.027640 6.272409 5.962854 6.337195
Mar 1962       6.267964 6.139016 6.396912 6.070755 6.465173
Apr 1962       6.295550 6.160356 6.430743 6.088789 6.502310
May 1962       6.328805 6.187643 6.469968 6.112916 6.544695
Jun 1962       6.465028 6.318138 6.611917 6.240380 6.689676
Jul 1962       6.603543 6.451142 6.755944 6.370465 6.836621
Aug 1962       6.599156 6.441435 6.756876 6.357943 6.840369
Sep 1962       6.420947 6.258081 6.583814 6.171865 6.670030
Oct 1962       6.305257 6.137403 6.473112 6.048546 6.561969
Nov 1962       6.159737 5.987038 6.332435 5.895617 6.423856
Dec 1962       6.264274 6.086864 6.441685 5.992948 6.535600
```

```
Jan 1963       6.302684 6.114929 6.490440 6.015537 6.589832
Feb 1963       6.246274 6.051158 6.441390 5.947870 6.544678
Mar 1963       6.364213 6.162005 6.566422 6.054962 6.673465
Apr 1963       6.391799 6.182738 6.600860 6.072068 6.711530
May 1963       6.425054 6.209359 6.640750 6.095176 6.754933
Jun 1963       6.561277 6.339145 6.783409 6.221555 6.900999
Jul 1963       6.699792 6.471405 6.928180 6.350504 7.049081
Aug 1963       6.695405 6.460929 6.929881 6.336805 7.054005
Sep 1963       6.517197 6.276786 6.757607 6.149521 6.884873
Oct 1963       6.401507 6.155305 6.647708 6.024974 6.778039
Nov 1963       6.255986 6.004126 6.507846 5.870800 6.641172
Dec 1963       6.360523 6.103130 6.617917 5.966874 6.754173
```

Agora vamos visualizar as predições (10 anos). Na Figura 34, os pontos em azul representam os valores preditos, e as faixas em cinza escuro e cinza claro representam os limites de confiança: 80% e 95%, respectivamente.

```
> plot(forecast(fit, 10 * 12), xlab = "Ano",
...      ylab = "Núm Passageiros", main = "")
> text(1949, 3.2, "Fonte: Elaborado pelo autor",
...      xpd = NA, cex = 0.8)
```

Figura 34: Previsões para o Modelo ARIMA(0,1,1) – 10 anos



Fonte: Elaborado pelo autor

Finalizando

Nesta Trilha estudamos as Séries Temporais, que são uma coleção de observações tomadas em sequência, ao longo de um período. Esta característica das séries temporais implica que as observações devem ser consideradas e analisadas na ordem correta. Outro ponto importante abordado, é a especificidade das séries temporais com relação aos procedimentos estatísticos tradicionais; estes procedimentos foram desenvolvidos, em sua grande maioria, para analisar observações independentes, assim, técnicas específicas são necessárias para a análise deste tipo de dado.

Nossa abordagem nos permitiu criar modelos para ajustar diversos tipos de séries temporais, caminhando em uma crescente de complexidade, até chegarmos ao modelo ARIMA.

Através dos modelos podemos fazer previsões de novos valores das séries temporais. A ideia de fazer previsões é uma das alternativas mais atrativas do trabalho com séries temporais.

Vimos nesta Trilha alguns dos muitos recursos que o **R** fornece para o trabalho com séries temporais, desde sua criação, manipulação, exploração e análise.

Como mencionamos, há outras técnicas, com abordagens diferentes, algumas mais avançadas, para a análise de séries temporais. Dentre as que mencionamos na Introdução, destacamos novamente aqui, os modelos **ARCH** e **GARCH**, além daquelas que empregam redes neurais, especialmente, *Deep Learning*.

Bibliografia

KABACOFF, R. I. **R in Action - Data Analysis and graphics with R**. 2nd. ed. Shelter Island, NY - USA: Manning Publications Co., 2015.

SRIVASTAVA, T. **A Complete Tutorial on Time Series Modeling in R.**Analytics Vidhya,, 2015. Disponível em: <<https://www.analyticsvidhya.com/blog/2015/12/complete-tutorial-time-series-modeling/>>