

Introdução aos modelos lineares dinâmicos

James D Santos

2025-04-09

Table of contents

Prefácio

1 Manipulação de séries temporais univariadas no software R

1.1 Definição de séries temporais

Considere um fenômeno aleatório que é observado ao longo do tempo. A coleção de variáveis aleatórias indexadas no tempo, associadas a este fenômeno, é denominada série temporal. Neste caso, para cada instante de tempo t , há uma variável $X(t)$ associada.

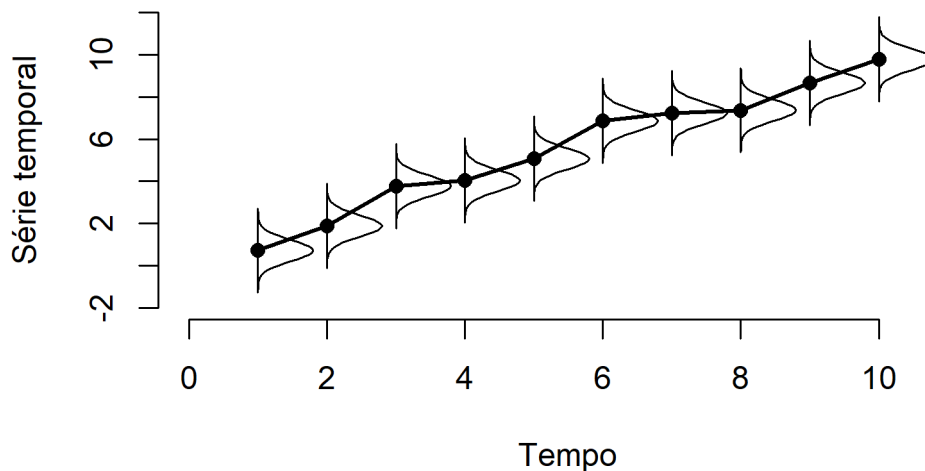


Figure 1.1: Figure 1 - Ilustração de uma série temporal

Alguns autores definem séries temporais simplesmente como valores observados ao longo do tempo. No entanto, essa definição não é útil para nós, uma vez que o tempo não necessariamente possui influência na variável, ou seja, é possível que a distribuição de $X(t)$ não dependa de t .

Importante: estamos interessados apenas em séries temporais nas quais o modelo de probabilidades depende do tempo t .

A partir deste momento, $X(t)$ será escrita como X_t e representará a variável aleatória associada ao tempo t e a versão minúscula x_t representará o valor observado.

1.2 A classe `ts` e a função `window`

Nessa seção vamos discutir a classe `ts` do R. Ela é particularmente útil para fazer gráficos de séries temporais. A função `ts` possui vários argumentos importantes:

- **data**: vetor ou matriz da série observada
- **frequency**: esse valor representa o número de observações por período. Vamos discutir essa particularidade em outro momento, mas para a maioria das séries, o período é representado por um ano e o valor de **frequency** está relacionado com quantas observações são necessárias para completar um ano. Por exemplo, se os dados são registrados mensalmente, temos **frequency=12**. Em caso de trimestres, **frequency=4**. O valor padrão é **frequency=1**.
- **deltat**: é o inverso do número de observações por período. Apenas um entre **frequency** e **deltat** deve ser dado.
- **start**: representa o tempo no qual a série começa. Pode ser representado por um único número ou por um vetor de dois números, com o segundo representando o momento dentro do período. Por exemplo:
 - se **frequency=12** (meses em um ano) então **start=c(1996,2)** implica que a primeira observação data de fevereiro de 1996.
 - se **frequency=4** (trimestres em um ano) então **start=c(1996,2)** implica que a primeira observação data do segundo trimestre de 1996.
- **end**: representa o tempo no qual a série termina. A sintaxe é a mesma do **start**
- **names**: é um vetor com o nomes das séries. É utilizado apenas quando há mais de uma série temporal.

Exemplo

Vamos ilustrar a construção de um objeto `ts` utilizando a tabela abaixo, que apresenta o número de nascidos vivos por mês na cidade de Manaus em 2021.

Mês	No. nascidos vivos
Janeiro	3043

Mês	No. nascidos vivos
Fevereiro	2902
Março	3166
Abril	3014
Maio	3095
Junho	2955
Julho	3087
Agosto	3141
Setembro	3129
Outubro	3096
Novembro	3191
Dezembro	3222

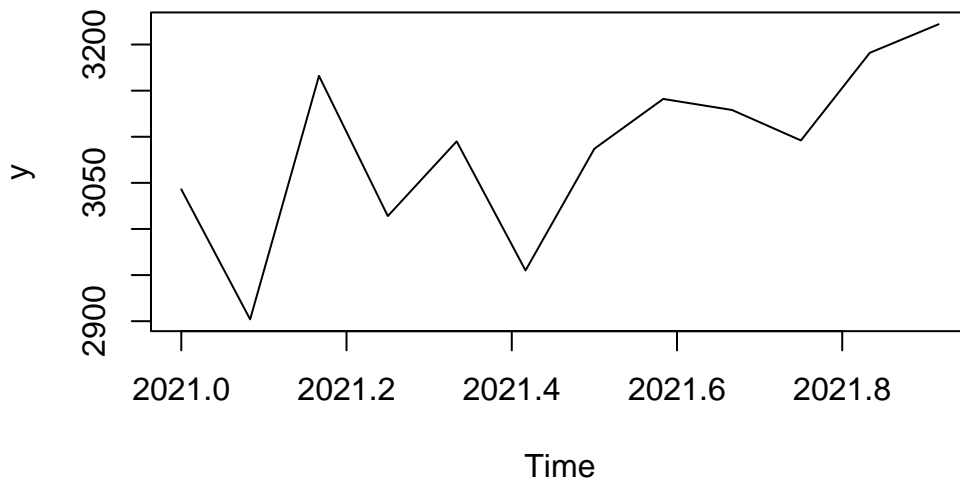
Vamos guarda a série no vetor `x` e construir o objeto `y` na classe `ts`.

```
x <- c(
  3043, 2902, 3166, 3014,
  3095, 2955, 3087, 3141,
  3129, 3096, 3191, 3222
)
y <- ts( x, start = c(2021,1), frequency = 12)
y
```

```
      Jan  Feb  Mar  Apr  May  Jun  Jul  Aug  Sep  Oct  Nov  Dec
2021 3043 2902 3166 3014 3095 2955 3087 3141 3129 3096 3191 3222
```

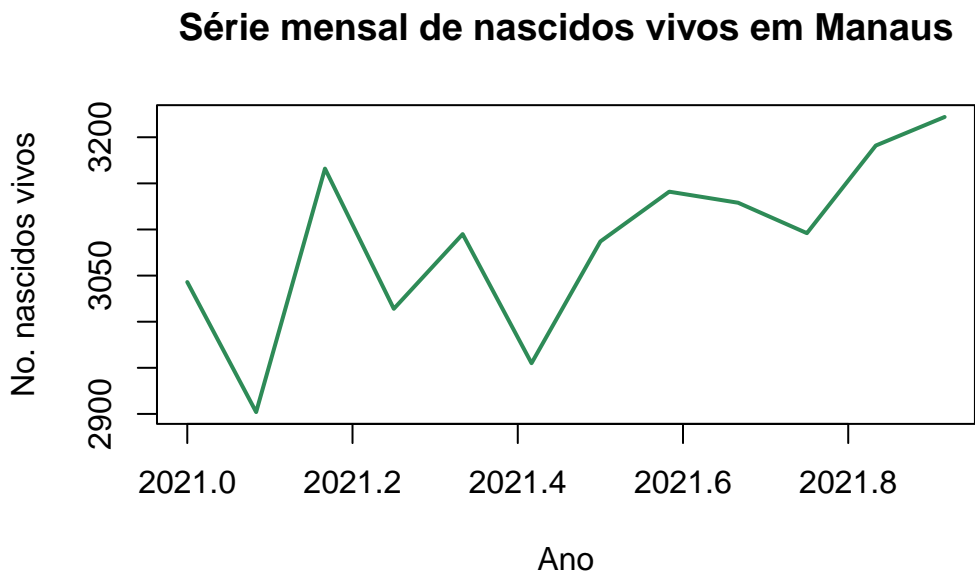
A função `plot` reconhece um objeto na classe `ts` e constrói um gráfico com o tempo devidamente marcado no eixo as abscissas.

```
plot(y)
```



Outros detalhes gráficos da função `plot` podem ser utilizados.

```
plot(y, ylab = 'No. nascidos vivos', lwd = 2, col = 'seagreen', xlab = 'Ano', main = 'Série mens
```



Fonte: Sistema de Informação sobre Nascidos Vivos/SUS

É possível extrair os argumentos de um `ts` já criado utilizando funções com os mesmos nomes dos respectivos argumentos. No exemplo acima, criamos o objeto denominado `y`. Abaixo, extraímos os argumentos deste objeto.

```
start(y)
```

```
[1] 2021    1
```

```
end(y)
```

```
[1] 2021   12
```

```
frequency(y)
```

```
[1] 12
```

```
deltat(y)
```

```
[1] 0.08333333
```

A função `window` é particularmente útil para selecionar um subconjunto da série temporal. Seus argumentos são os mesmos da função `ts`.

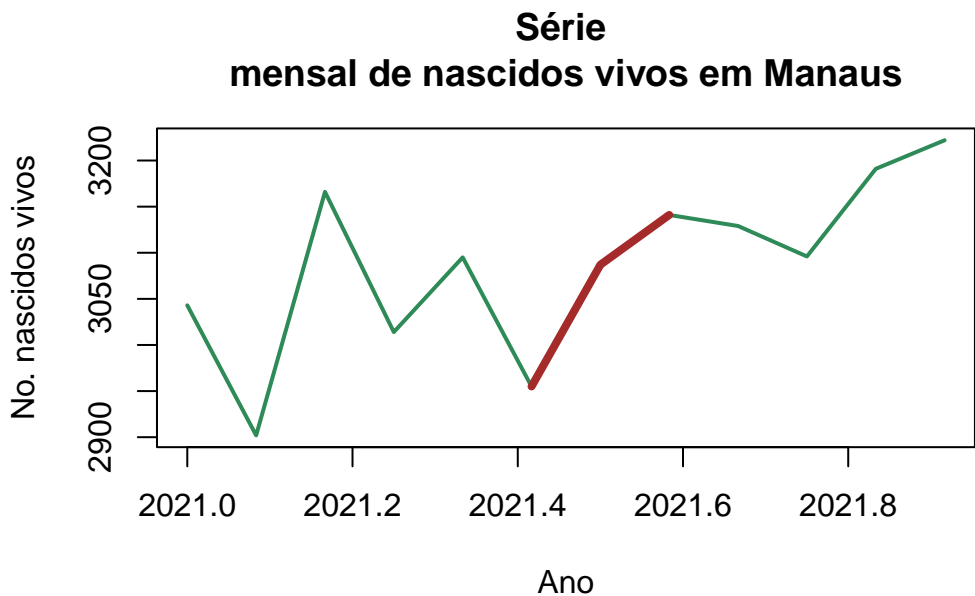
Exemplo Utilizando o mesmo conjunto de dados do exemplo anterior, vamos usar a função `window` para extrair apenas os nascimentos entre junho e agosto.

```
z <- window(y, start=c(2021,6), end = c(2021,8))
z
```

```
      Jun  Jul  Aug
2021 2955 3087 3141
```

Acima, `z` é um novo objeto `ts`. Podemos usar a função `lines` destacar a parte selecionada da série em um gráfico já existente. Abaixo, destacamos os dados selecionados em `z`.

```
plot(y, ylab = 'No. nascidos vivos', lwd = 2, col = 'seagreen', xlab = 'Ano', main = 'Série
mensal de nascidos vivos em Manaus', sub='Fonte: Sistema de Informação sobre Nascidos Vivos/')
lines(z, col = 'brown', lwd = 4 )
```

Fonte: Sistema de Informação sobre Nascidos Vivos/SUS

Exercício 1

A série abaixo representa o número de homicídios mensais no Amazonas, segundo causa básica de óbito, entre os anos 2000 e 2023.

```
require(gsheet)
url = 'https://docs.google.com/spreadsheets/d/1rtiy0Z1W3SRIWZJTR1RyBmxmnG011x005GNGrGLBr5A/e
hom = gsheets2tbl(url)
```

1. Construa um objeto do tipo `ts`
2. Faça um gráfico da série.
3. Crie uma janela para marcar o período entre o início da pandemia de COVID-19 (março de 2020) e o primeiro dia sem mortes por COVID-19 (julho de 2021)
4. Represente a janela acima no gráfico anterior. O que esse gráfico revela?

1.3 A classe `Date` e o pacote `lubridate`

Nessa seção discutimos a classe `Date`, responsável por operações com datas no R. São apresentadas as principais funções do pacote `base`. Em seguida, apresentamos o pacote `lubridate`, que oferece funções adicionais e uma sintaxe mais fluida.

No pacote `base`, as datas são objeto da classe `Date`. Abaixo, transformamos o texto que representa 3 de agosto de 1998 nessa classe.

```
# 3 de agosto de 1998 (formato americano)
x <- '1998/8/3'
y <- as.Date(x)
```

Existem diversas funções que interagem com objetos nessa classe:

- `weekdays`: Retorna o dia da semana.
- `months`: Retorna o nome do mês.
- `quarters`: Retorna o trimestre do ano (Q1, Q2, Q3 ou Q4).

Abaixo, ilustramos o uso dessas funções com a data 3 de agosto de 1998.

```
weekdays(y)
```

```
[1] "segunda-feira"
```

```
months(y)
```

```
[1] "agosto"
```

```
quarters(y)
```

```
[1] "Q3"
```

Outra vantagem desta classe é a possibilidade de calcular a diferença em dias entre duas datas, utilizando a função `-`. Abaixo mostramos a diferença entre 3 de agosto de 1998 e 3 de agosto de 1999.

```
z <- as.Date('1999-08-03')
z-y
```

```
Time difference of 365 days
```

Em certas aplicações, é necessário criar um vetor contendo datas em sequência. A função `seq` interage com objetos da classe `Date`, permitindo que o argumento `by` receba as strings `day`, `week`, `month`, `quarter` e `year`. Abaixo, criamos um vetor mensal que começa em 3 de agosto de 1998 e terminando em 3 de agosto de 1999.

```
inicio <- as.Date('1998-08-03')
fim <- as.Date('1999-08-03')
seq(inicio, fim, by='month')
```

```
[1] "1998-08-03" "1998-09-03" "1998-10-03" "1998-11-03" "1998-12-03"
[6] "1999-01-03" "1999-02-03" "1999-03-03" "1999-04-03" "1999-05-03"
[11] "1999-06-03" "1999-07-03" "1999-08-03"
```

Observe que a conversão da `string` para `Date` é realizada considerando o formato americano por padrão. É possível usar a função `as.Date` para ler qualquer formato, modificando o argumento `format`. No entanto, o pacote `lubridate` oferece funções mais simples para essa conversão:

- `ymd`: Converte `strings` no formato “ano, mês, dia”, como “2023-10-26”.
- `mdy`: Converte `strings` no formato “mês, dia, ano”, como “10-26-2023”.
- `dmy`: Converte `strings` no formato “dia, mês, ano”, como “26-10-2023”.

Abaixo, transformamos a data 3/8/1998 para o formato americano.

```
require(lubridate)
# 3 de agosto de 1998 (formato nacional)
x <- '3/8/1998'
dmy(x)
```

```
[1] "1998-08-03"
```

O `lubridate` também oferece a possibilidade de trabalhar com informações de tempo dentro de um dia, como horas, minutos e segundos. Por exemplo, a informação 15h30 de 3 de agosto de 1998 é lida como

```
x <- '3/8/1998 15:30:00'
dmy_hms(x)
```

```
[1] "1998-08-03 15:30:00 UTC"
```

O `lubridate` possui as funções `month` e `wday`, que funcionam de modo análogo às funções `months` e `weekdays`. Além disso, o `lubridate` traz uma série de funções adicionais como:

- `year`: retorna o ano de uma data
- `day`: retornam o dia de uma data (útil para o formato `xxx-xx-xx 00:00:00`)
- `hour`, `minute`, `second`: Retornam a hora, minuto e segundo de um objeto de data e tempo.

As funções de arredondamento de data também são úteis, especialmente para obter contagens mensais, anuais, etc. Elas são `floor_date` e `ceiling_date` e são responsáveis por arredondar uma data para o início ou o fim de um período, respectivamente. Abaixo, arredondamos a data 3 de agosto de 1998 para o começo do mês.

```
x <- dmy('03/08/1998')
floor_date(x, 'month')
```

```
[1] "1998-08-01"
```

Exemplo

A Força Aérea Brasileira (FAB), por meio do Centro de Investigação e Prevenção de Acidentes Aeronáuticos (CENIPA), possui um `dashboard` para explorar dados sobre incidentes e acidentes aéreos no Brasil. Um acidente é definido como uma ocorrência grave associada à operação de uma aeronave que resulta em lesões ou morte, dano estrutural da aeronave ou aeronave desaparecida. Os demais casos são classificados como incidentes. Os dados, atualizados em 11/08/2025, estão disponíveis para esse curso na `url` abaixo:

```
url <- 'https://docs.google.com/spreadsheets/d/1BjTXFMmTpcxKdRHCr5IIDJde9yr0s3oAAfyj2vdVuX8/'
aereo <- gsheets2tbl(url)
head(aereo)
```

```
# A tibble: 6 x 10
```

	Link	Data	Matrícula	Classificação	Tipo	Localidade	UF	Aeródromo	Operação
	<chr>	<chr>	<chr>	<chr>	<chr>	<chr>	<chr>	<chr>	<chr>
1	https~	05/0~	****	INCIDENTE	FALH~	RIO DE JA~	RJ	FAER	TÁXI AÉ~
2	https~	04/0~	****	INCIDENTE	FALH~	MARICÁ	RJ	NCAD	TÁXI AÉ~
3	https~	01/0~	****	INCIDENTE	FALH~	RIO DE JA~	RJ	SBJR	TÁXI AÉ~
4	https~	31/0~	****	INCIDENTE	FALH~	PALMAS	TO	SBPJ	REGULAR
5	https~	29/0~	****	INCIDENTE	FALH~	BOA VISTA	RR	FAER	TÁXI AÉ~
6	https~	29/0~	****	INCIDENTE	COLI~	RECIFE	PE	SBRF	REGULAR

```
# i 1 more variable: Status <chr>
```

A unidade amostral é o acidente/indicente. Estamos interessados em criar uma série temporal com o número de acidentes mensais. Abaixo, filtramos apenas os acidentes (coluna `Classificação`) e, em seguida, transformamos as datas em objetos do tipo `Date`.

```
acidentes <- aereo[ aereo$Classificação=='ACIDENTE', ]
datas <- dmy(acidentes$Data)
```

Agora, vamos arredondar todas as datas para o primeiro dia do mês. Em seguida, contaremos as frequências para cada mês/ano

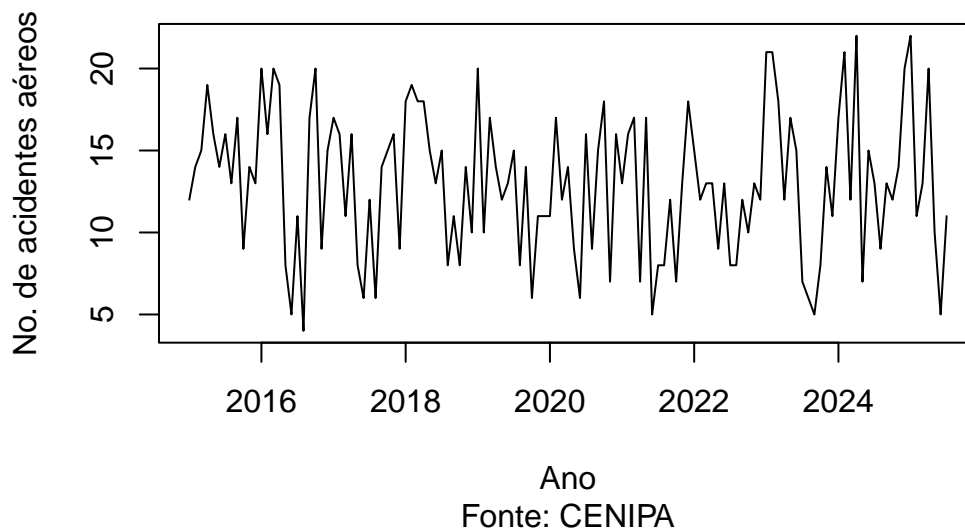
```
mes_ano <- floor_date(datas, 'month')
contagem <- table(mes_ano)
contagem
```

```
mes_ano
2015-01-01 2015-02-01 2015-03-01 2015-04-01 2015-05-01 2015-06-01 2015-07-01
          12          14          15          19          16          14          16
2015-08-01 2015-09-01 2015-10-01 2015-11-01 2015-12-01 2016-01-01 2016-02-01
          13          17           9          14          13          20          16
2016-03-01 2016-04-01 2016-05-01 2016-06-01 2016-07-01 2016-08-01 2016-09-01
          20          19           8           5          11           4          17
2016-10-01 2016-11-01 2016-12-01 2017-01-01 2017-02-01 2017-03-01 2017-04-01
          20           9          15          17          16          11          16
2017-05-01 2017-06-01 2017-07-01 2017-08-01 2017-09-01 2017-10-01 2017-11-01
           8           6          12           6          14          15          16
2017-12-01 2018-01-01 2018-02-01 2018-03-01 2018-04-01 2018-05-01 2018-06-01
           9          18          19          18          18          15          13
2018-07-01 2018-08-01 2018-09-01 2018-10-01 2018-11-01 2018-12-01 2019-01-01
          15           8          11           8          14          10          20
2019-02-01 2019-03-01 2019-04-01 2019-05-01 2019-06-01 2019-07-01 2019-08-01
          10          17          14          12          13          15           8
2019-09-01 2019-10-01 2019-11-01 2019-12-01 2020-01-01 2020-02-01 2020-03-01
          14           6          11          11          11          17          12
2020-04-01 2020-05-01 2020-06-01 2020-07-01 2020-08-01 2020-09-01 2020-10-01
          14           9           6          16           9          15          18
2020-11-01 2020-12-01 2021-01-01 2021-02-01 2021-03-01 2021-04-01 2021-05-01
           7          16          13          16          17           7          17
2021-06-01 2021-07-01 2021-08-01 2021-09-01 2021-10-01 2021-11-01 2021-12-01
           5           8           8          12           7          13          18
2022-01-01 2022-02-01 2022-03-01 2022-04-01 2022-05-01 2022-06-01 2022-07-01
          15          12          13          13           9          13           8
2022-08-01 2022-09-01 2022-10-01 2022-11-01 2022-12-01 2023-01-01 2023-02-01
```

8	12	10	13	12	21	21
2023-03-01	2023-04-01	2023-05-01	2023-06-01	2023-07-01	2023-08-01	2023-09-01
18	12	17	15	7	6	5
2023-10-01	2023-11-01	2023-12-01	2024-01-01	2024-02-01	2024-03-01	2024-04-01
8	14	11	17	21	12	22
2024-05-01	2024-06-01	2024-07-01	2024-08-01	2024-09-01	2024-10-01	2024-11-01
7	15	13	9	13	12	14
2024-12-01	2025-01-01	2025-02-01	2025-03-01	2025-04-01	2025-05-01	2025-06-01
20	22	11	13	20	10	5
2025-07-01						
11						

É sempre importante checar o resultado em `contagem`, para verificar se não há algum mês ausente. Como não o caso aqui, vamos construir um objeto do tipo `ts` e fazer o gráfico da série temporal.

```
serie <- ts(contagem, start=c(2015,1), frequency = 12 )
plot(serie, ylab = 'No. de acidentes aéreos', xlab = 'Ano', sub= 'Fonte: CENIPA' )
```



Exercício 2

A série abaixo contém as datas dos óbitos maternos no Brasil a partir de 2010.

```
url <- 'https://drive.google.com/uc?authuser=0&id=1tYFFT9L2iopKmBDUI3P8qNIRa0nMYj7d&export=d'
```

Crie uma série temporal com o número de óbitos mensal e faça um gráfico. Crie uma janela para destacar no gráfico o período da pandemia de COVID-19 (março de 2020 até julho de 2021).

2 Sinal, ruído e correlograma

2.1 Sinal e ruído

Em geral, a série temporal possui componentes de dois tipos: sinal e ruído. O primeiro é uma função do tempo geralmente relacionado com a média da série em dado instante de tempo, enquanto que o segundo está relacionado com a variância. Podemos assumir que essa relação é aditiva:

$$X_t = \text{sinal}(t) + \varepsilon_t$$

onde ε_t é o ruído.

Em alguns casos essa relação é multiplicativa, ou seja,

$$X_t = \exp\{\text{sinal}(t) + \varepsilon_t\},$$

e, nesses casos, aplicamos o logaritmo na série para que as componentes se tornem aditivas.

Os sinais mais importantes são:

- Tendência: um comportamento de subida ou descida que pode ser observado no médio/longo prazo. Uma série temporal com tendência costuma ser bem representada por polinômios, ou seja

$$x_t = \sum_{j=0}^q \beta_j t^j + \varepsilon_t.$$

- Sazonalidade: é um padrão de oscilação que ocorre em um período fixo e conhecido. Cheias de rios e quantidade de chuva são exemplos de padrões sazonais. Em termos gerais, o padrão sazonal pode ser representado por um harmônico, ou seja,

$$x_t = A \cos\left(\phi + \frac{2\pi}{p}t\right) + \varepsilon_t$$

onde p é o período sazonal, A é amplitude da onda e ϕ a fase.

- Ciclos: é um padrão oscilatório sem período fixo, como o ciclo de recessão de uma economia. Modelar ciclos é um pouco mais complexo, mas veremos posteriormente que é possível escrever séries deste tipo como

$$x_t = A_t \cos(\phi_t + 2\pi\lambda t) + \varepsilon_t,$$

onde a amplitude e a fase variam no tempo e o período $1/\lambda$ é desconhecido.

Os padrões oscilatórios (sazonalidade e ciclos) descritos acima variam em torno de zero. Caso seja necessário, um parâmetro μ constante pode ser adicionado para caracterizar uma oscilação em torno de uma média.

É possível que uma série possua todos os sinais acima, sendo escrita como

$$x_t = \sum_{j=0}^q \beta_j t^j + A \cos(\phi + 2\pi\lambda t) + A_t \cos(\phi_t + 2\pi\lambda t) + \varepsilon_t$$

O ruído é a parte aleatória da composição e possui média zero. Isso implica que, para uma série com sinal $f(t)$, teremos

$$E(x_t) = E(f(t)) + E(\varepsilon_t) = f(t)$$

e

$$Var(x_t) = Var(f(t) + \varepsilon_t) = Var(\varepsilon_t)$$

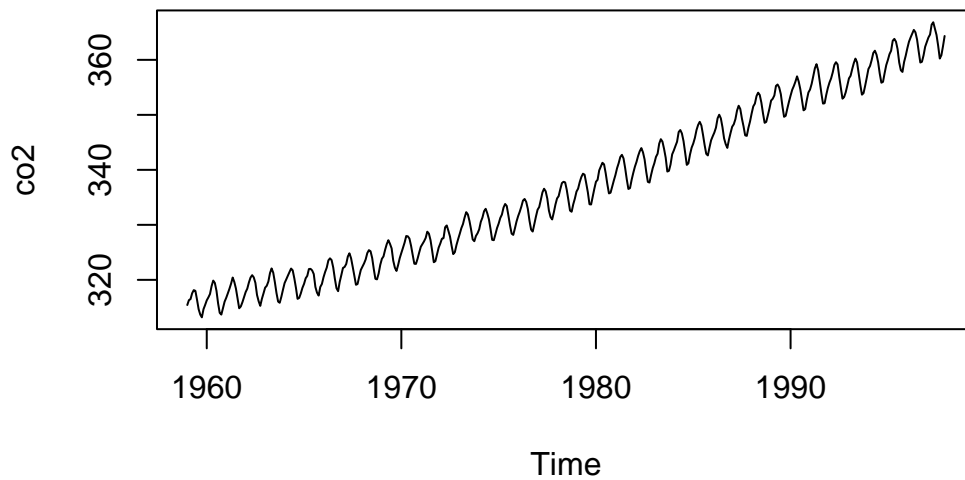
Os ruídos mais importantes são:

- Ruído branco: possuem variância constante e são não correlacionados, ou seja $Cov(\varepsilon_t, \varepsilon_s) = 0$ para todo $t \neq s$.
- Média móvel de ordem q : sejam $\dots, \varepsilon_{-2}, \varepsilon_{-1}, \varepsilon_0, \varepsilon_1, \dots$ ruídos brancos. O ruído de média móvel de ordem q é dado por

$$\eta_t = \varepsilon_t + \sum_{j=1}^q \theta_j \varepsilon_{t-j}.$$

A série a seguir representa uma série mensal de concentração de CO_2 na atmosfera em Mauna Loa, expressa em partes por milhão. Note o comportamento da tendência e da sazonalidade.

```
plot(co2)
```

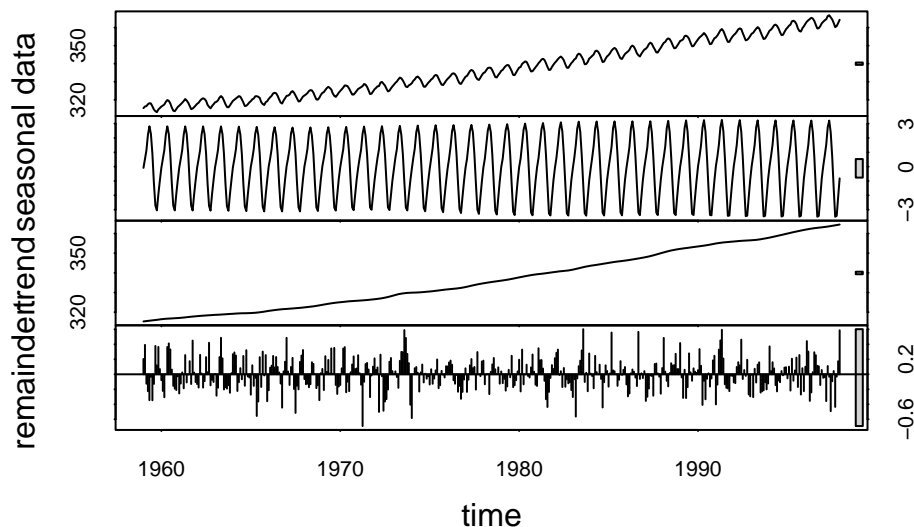


Existem diversas ferramentas que estimam a tendência e a sazonalidade. Dentre estas, destaca-se o STL, por sua robustez. Abaixo, ilustramos a decomposição da série `co2`. O termo **remainder**, também conhecido como resíduo, é calculado como

$$e_t = x_t - \hat{f}(t),$$

onde $\hat{f}(t)$ é o sinal (tendência+sazonalidade) ajustado. Se não houver qualquer outro sinal a ser ajustado, então e_t é uma estimativa para ε_t .

```
# p é o período do padrão sazonal  
p = 12  
plot(stl(co2,p))
```



Exercício. Considere a série `AirPassengers`, disponível no pacote `datasets`.

1. Faça um gráfico da série temporal e procure determinar suas componentes.
2. A função `decompose` estima a sazonalidade assumindo a média das observações correspondentes ao mesmo fator sazonal (por exemplo, todos os janeiros). Faça um gráfico do resultado do `decompose` para esta série e discuta os resultados (em especial o termo `random`, que é o resíduo da decomposição).
3. Repita o item 2, mas utilizando a função `stl`. Quais são as principais diferenças?
4. Aplique o logaritmo da série e verifique o seu gráfico. Quais são as semelhanças e diferenças deste gráfico com o obtido no item 1?

Exercício. Na série abaixo temos a taxa de desemprego mensal no Brasil entre março de 2002 e dezembro de 2015. Analise as componentes desta série.

```
url <- 'https://www.dropbox.com/s/rmgymzsic99qawd/desemprego.csv?dl=1'
banco <- read.csv(url, sep = ';', h = F)
```

2.2 A função de autocorrelação

Considere inicialmente uma amostra aleatória X_1, \dots, X_n (ou seja, todas as variáveis são independentes e possuem a mesma distribuição). Sejam

$$A_h = \{X_1, \dots, X_{n-h}\}$$

e

$$B_h = \{X_h, \dots, X_n\}.$$

Então, a correlação entre A_h e B_h é nula.

Deste modo, um meio de verificar se a coleção observada é uma série temporal é observar a correlação amostral entre

$$a_h = \{x_1, \dots, x_{n-h}\}$$

e

$$b_h = \{x_h, \dots, x_n\},$$

para diferentes valores de h .

A função $r(h)$ que representa a correlação amostral entre a_h e b_h é denominada **autocorrelação**. O valor h é denominado **defasagem** (do inglês, *lag*).

Propriedades

- $r(0) = 1$
- $-1 \leq r(h) \leq 1$

Correlograma O gráfico $(h, r(h))$ é denominado correlograma, ou gráfico da função de autocorrelação.

2.2.1 O correlograma de uma amostra aleatória

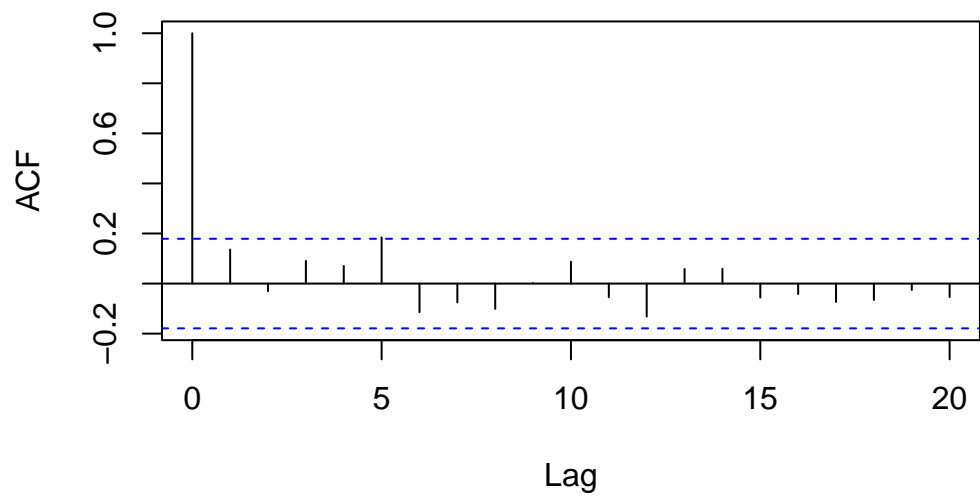
Quando a amostra é aleatória, a função de autocorrelação é nula para qualquer defasagem diferente de 0. Deste modo, o correlograma deve apresentar valores próximos de zero.

Para entender o que próximo de zero significa, o limites do intervalo de confiança para o coeficiente de correlação sobre a hipótese de que esta é nula são colocados no gráfico.

Abaixo ilustramos um correlograma para uma amostra de variáveis aleatórias independentes com distribuição normal padrão.

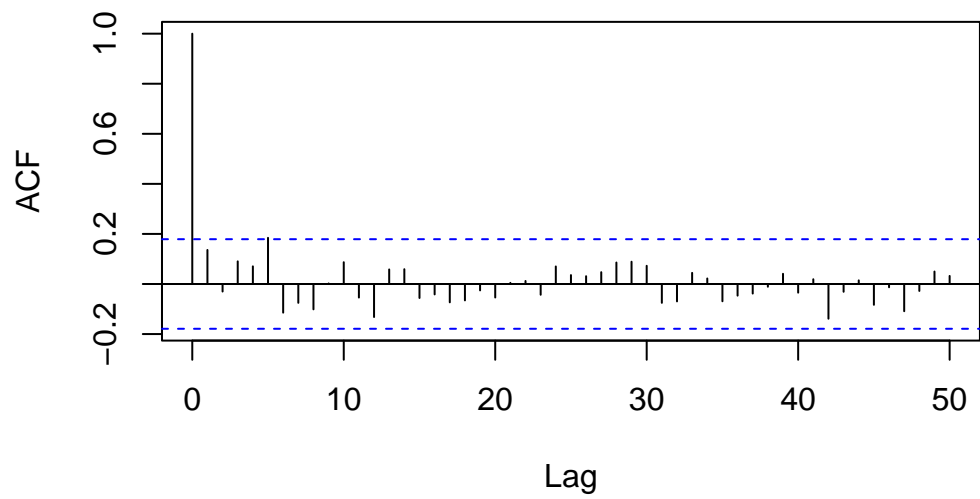
```
x <- rnorm(120)
# correlograma
acf(x)
```

Series x



```
# o mesmo correlograma com uma defasagem maior  
acf(x, lag = 50)
```

Series x



2.2.2 O correlograma com a componente de tendência

Quando uma série exibe tendência, o correlograma exibe um decaimento lento e persistente.

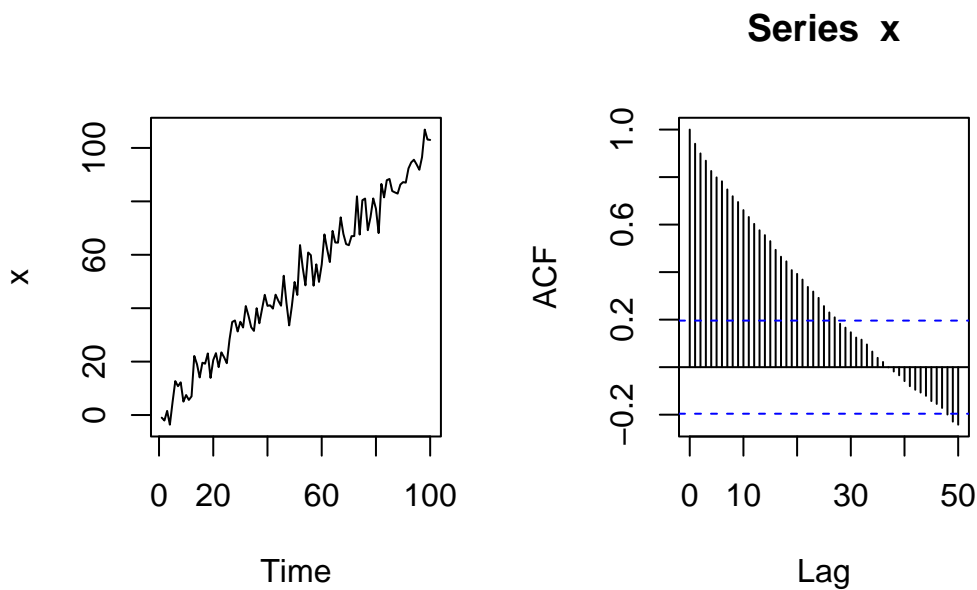
Considere, por exemplo, a série

$$x_t = t + \varepsilon_t,$$

onde $\varepsilon_t \sim \text{Normal}(0, 5^2)$. Abaixo simulamos essa série e apresentamos o respectivo correlograma

```
x <- rnorm(100, 1:100, 5)

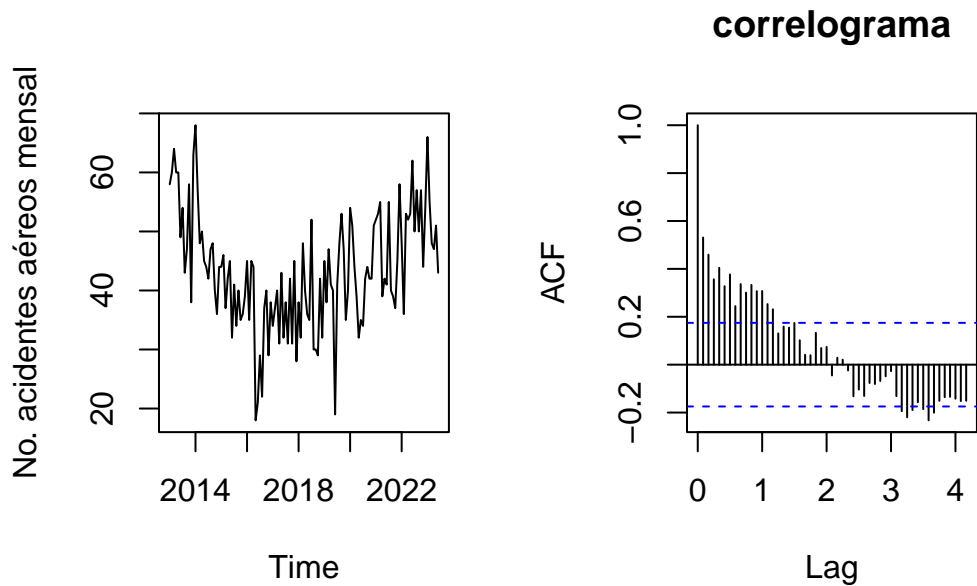
oo <- par( mfrow=c(1,2))
ts.plot(x)
acf(x, lag = 50)
```



```
par(oo)
```

Observe as similaridades do correlograma acima com o observado para a série de acidentes aéreos mensais vista anteriormente.

```
oo <- par( mfrow=c(1,2))
ts.plot( fab_mes , ylab = 'No. acidentes aéreos mensal' )
acf(fab_mes , lag = 50, main ='correlograma')
```



```
par(oo)
```

2.2.3 O correlograma com a componente de sazonalidade - sinal harmônico

O sinal sazonal é caracterizado por um comportamento periódico. Existem dois comportamentos sazonais típicos. O primeiro é baseado na função harmônica:

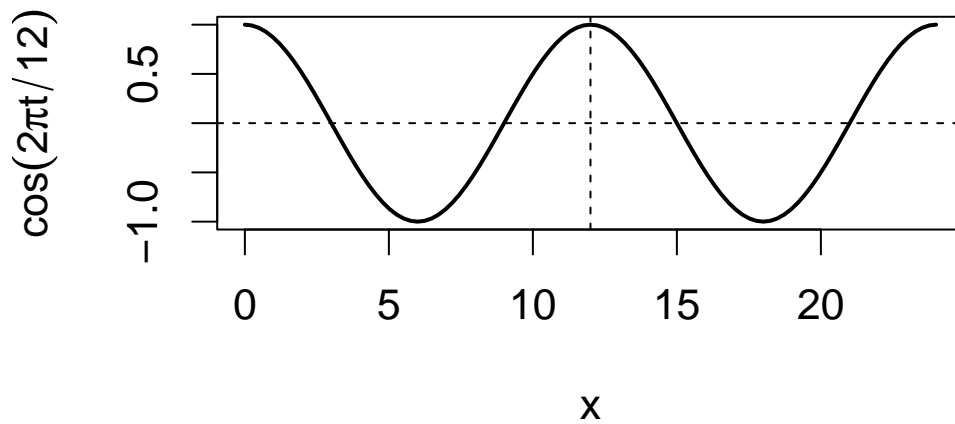
$$\text{sinal}(t) = A \cos\left(\frac{2\pi}{p}t + \phi\right)$$

Neste tipo de sinal, há um comportamento em forma de onda já estabelecido. Eis algumas informações importantes:

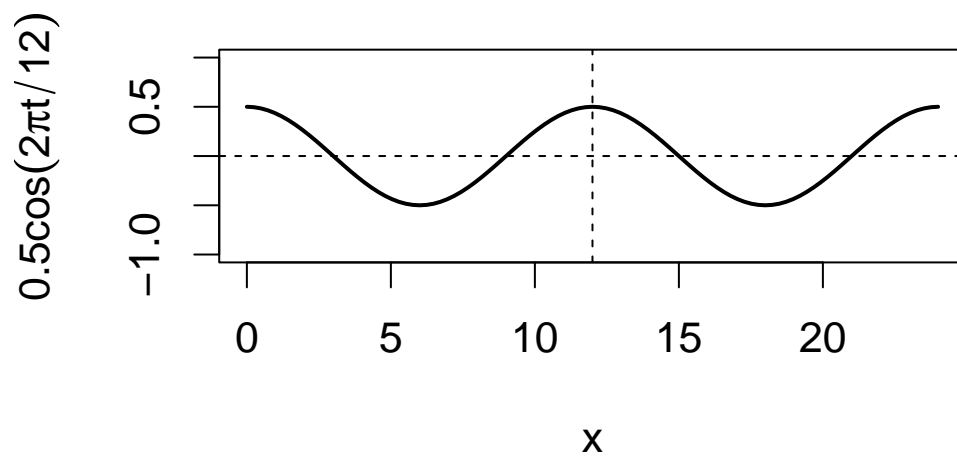
- O valor p , denominado período, equivale ao tempo que demora para o padrão se repetir.
- A é denominado amplitude e representa o maior/menor valor que este sinal pode atingir.
- Por último, ϕ é denominado fase, e serve basicamente para deslocar a onda.

Abaixo seguem alguns exemplos de harmônicos, todos com período 12:

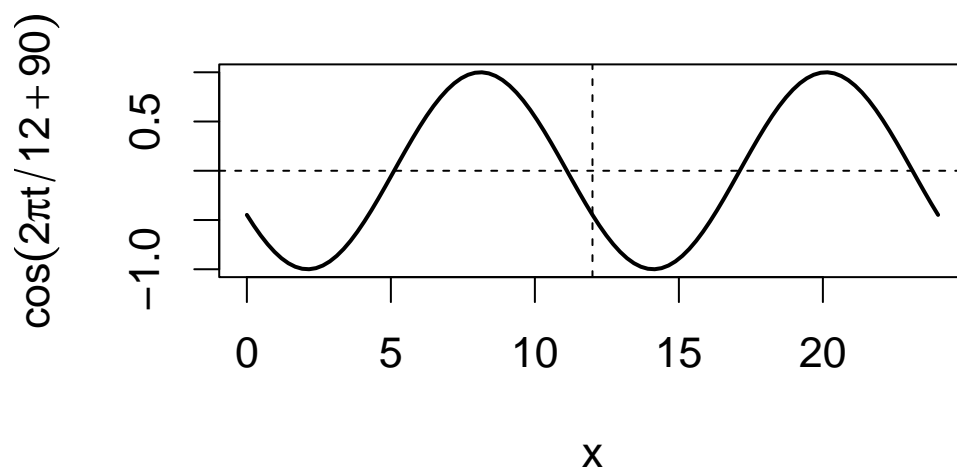
```
oo <- par( cex = 1.3)
curve( cos( x* 2*pi/12), 0,24, lwd = 2, ylab = expression( cos( 2*pi*t/12 )))
abline(h = 0, lty = 2 )
abline(v=12, lty = 2)
```



```
curve( .5*cos( x* 2*pi/12), 0,24, lwd = 2, ylab = expression( .5*cos( 2*pi*t/12 )), ylim = c
abline(h = 0, lty = 2 )
abline(v=12, lty = 2)
```

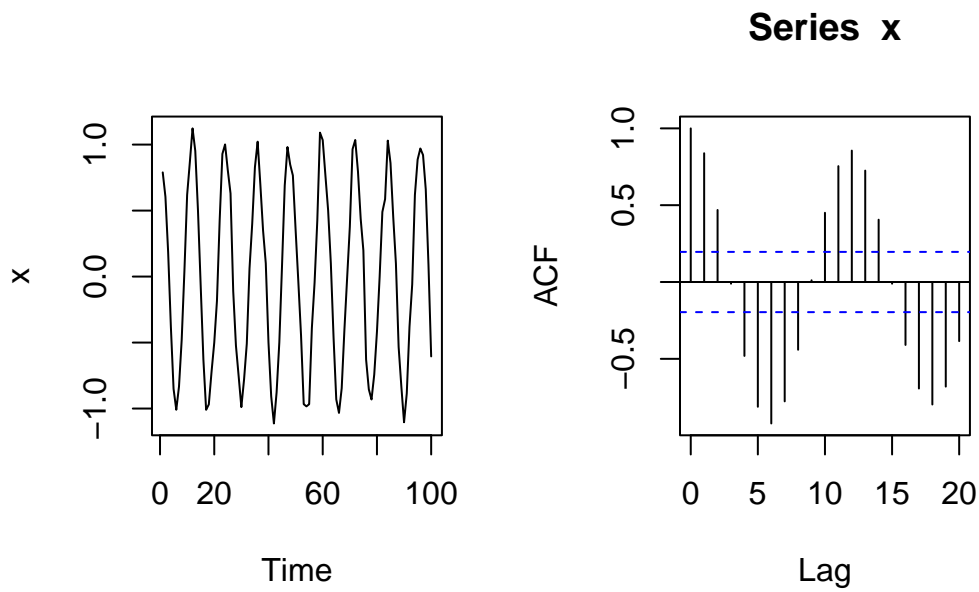
```
curve( cos( x* 2*pi/12+90), 0,24, lwd = 2, ylab = expression( cos( 2*pi*t/12 +90)), ylim = c
abline(h = 0, lty = 2 )
abline(v=12, lty = 2)
```



Abaixo simulamos uma série temporal com um sinal do tipo harmônico. Observe que o comportamento em forma de onda é aparente na função de autocorrelação.

```
x <- cos( 2*pi/12 * 1:100) + rnorm(100,0,.1)

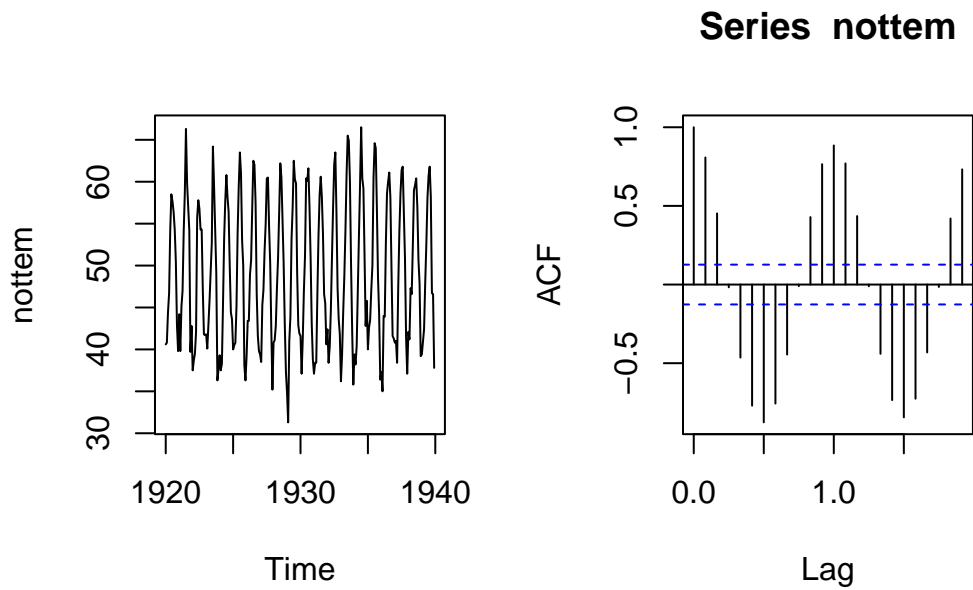
oo <- par( mfrow = c(1,2))
ts.plot(x)
acf(x)
```



```
par(oo)
```

Abaixo, apresentamos a temperatura mensal observada no Castelo de Nottingham, entre 1920-1939. Compare os resultados com os gráficos acima.

```
oo <- par( mfrow = c(1,2))
plot(nottem)
acf(nottem)
```



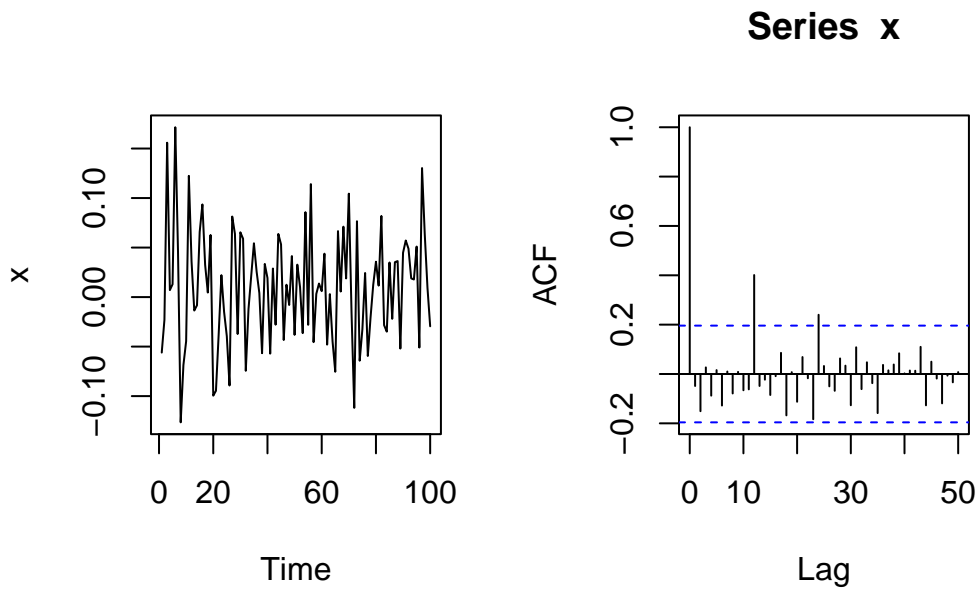
```
par(oo)
```

2.2.4 O correlograma com a componente de sazonalidade - sinal autorregressivo

Nesse tipo de sazonalidade, ainda há um período p , mas não há um sinal harmônico. O valor da série no tempo t é baseado no valor observado no tempo $t - p$.

Quando a sazonalidade possui essa característica, há uma autocorrelação marcante nos múltiplos de p . Observe a série simulada abaixo, com um período $p = 12$

```
set.seed(123)
oo <- par( mfrow = c(1,2))
x <- rnorm(12,0,.1)
for(i in 13:100) x[i] <- .6*x[i-12] + rnorm(1,0,.05)
ts.plot(x)
acf(x, lag = 50)
```



```
par(oo)
```

2.3 Exercícios

Exercício 1 Estude o comportamento da série `ldeaths`, que conta o número mensal de óbitos por doenças pulmonares no Reino Unido.

Exercício 2 Estude o comportamento da série do número de óbitos maternos mensais.

Exercício 3 Em 2017, um epidemiologista estava interessado na série de suicídios no Mato Grosso do Sul. O banco de dados utilizado é dado a seguir. Construa uma série mensal e estude seu comportamento

```
url <- 'https://drive.google.com/uc?authuser=0&id=1DMSgrQDl0636Lw0Y0MYJHJrgw_2uXntM&export=d'
```