

# LISTA 1

## ANÁLISE DE SÉRIES TEMPORAIS

Tailine J. S. Nonato

2023-03-23

### Descrição da atividade

- Exercícios 1.1 a 1.6 do Cap.1 (pag.10) de Cryer & Chan (2008)

Antes de iniciar os exercícios propostos, carrega-se os pacotes necessários:

```
if (!require(pacman)) install.packages("pacman")
pacman::p_load(tidyverse,TSA)
```

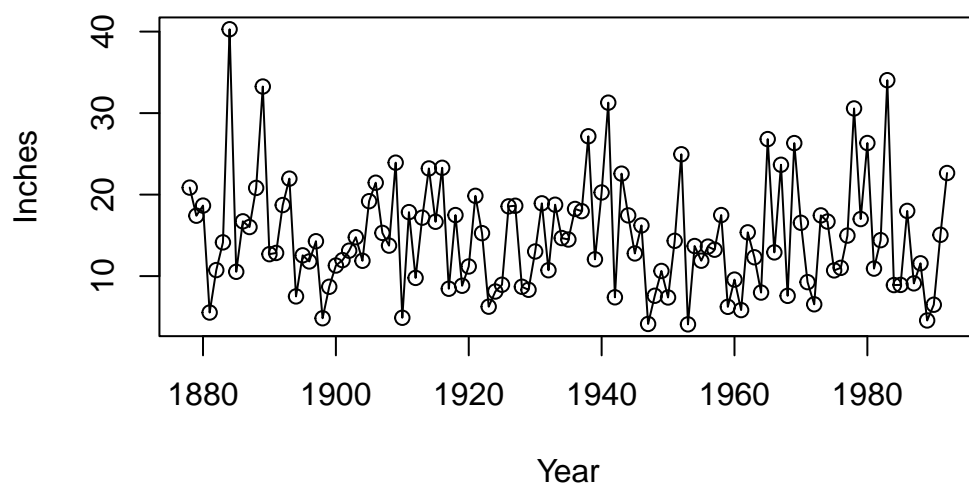
### Exercício 1.1

Use software to produce the time series plot shown in Exhibit 1.2, on page 2. The data are in the file named larain.

- Resposta:

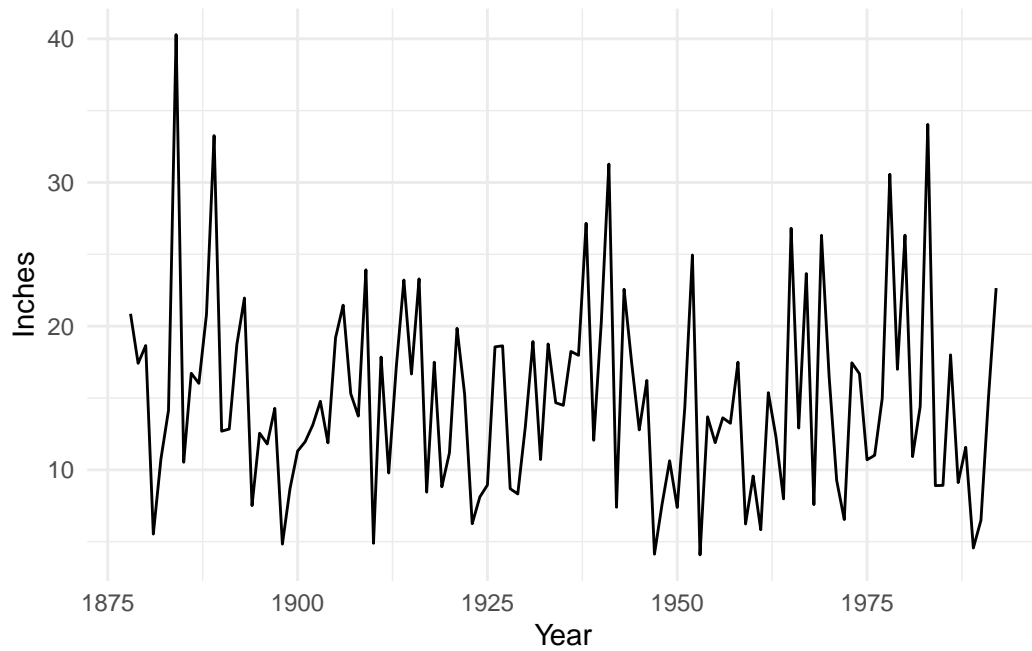
Em primeiro lugar, é necessário trazer as informações dadas no livro texto. O enunciado fala sobre reproduzir a série temporal de Exhibit 1.2, que é na verdade um Gráfico de Dispersão. Assim, reproduzindo a Exhibit 1.1 (que é realmente uma série temporal), tem-se utilizando o código do livro:

```
data(larain)
plot(larain,ylab='Inches',xlab='Year',type='o')
```



Agora utilizando o ggplot:

```
ggplot(data = data.frame(larain), aes(x = time(larain), y = as.vector(larain))) +  
  geom_line() +  
  labs(x = "Year", y = "Inches") +  
  theme_minimal()
```



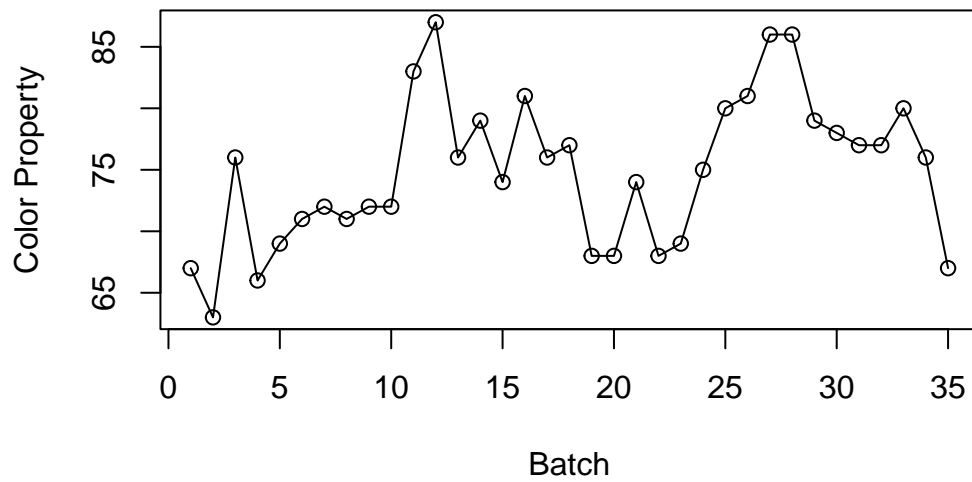
### Exercício 1.2

Produce the time series plot displayed in Exhibit 1.3, on page 3. The data file is named color.

- Resposta:

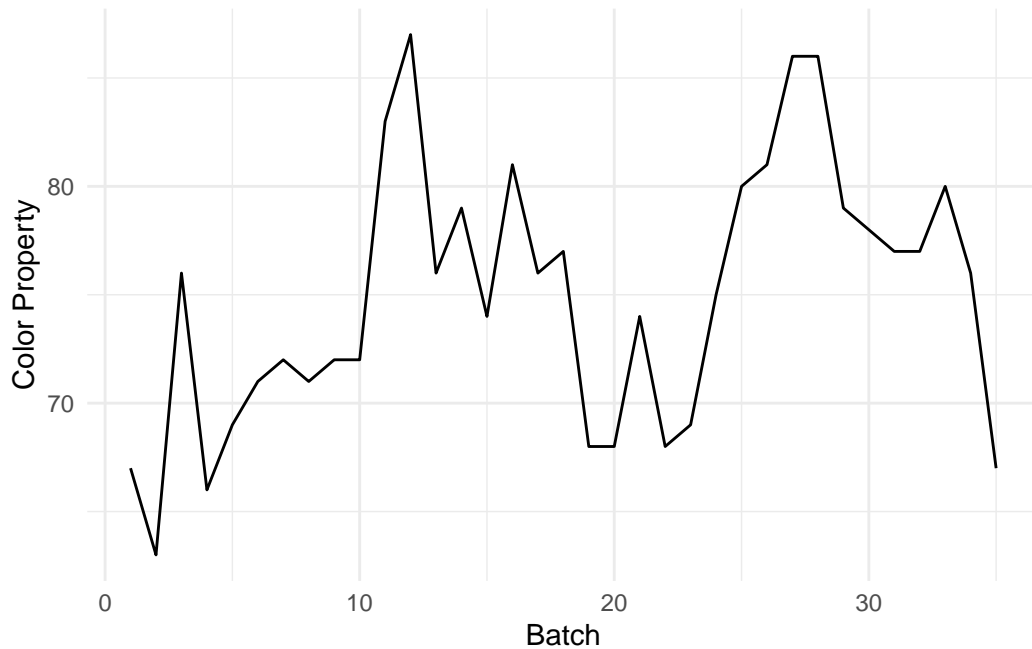
Utilizando o código do livro:

```
data(color)
plot(color,ylab='Color Property',xlab='Batch',type='o')
```



Agora utilizando o ggplot:

```
ggplot(data = data.frame(color), aes(x = time(color), y = as.vector(color))) +  
  geom_line() +  
  labs(x = "Batch", y = "Color Property") +  
  theme_minimal()
```



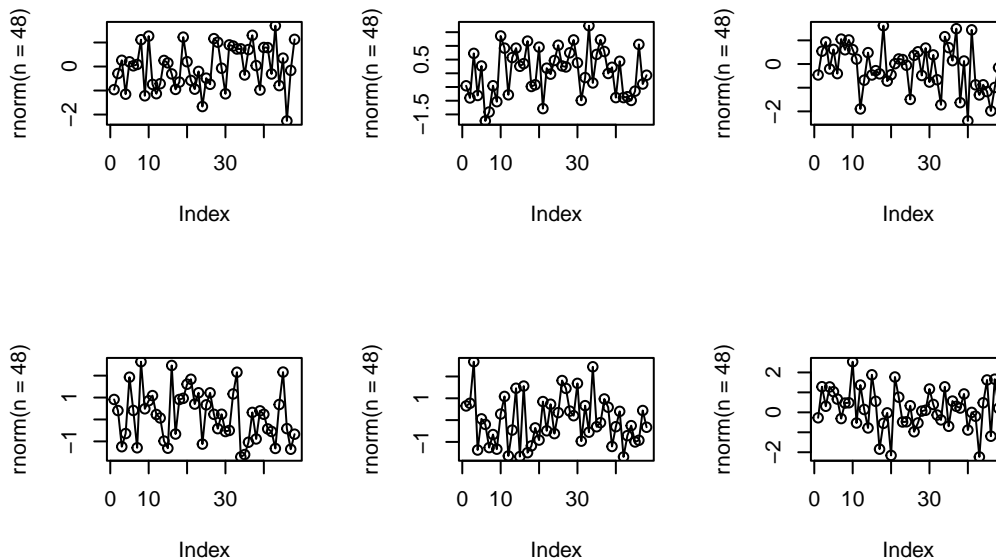
### Exercício 1.3

Simulate a completely random process of length 48 with independent, normal values. Plot the time series plot. Does it look “random”? Repeat this exercise several times with a new simulation each time.

- Resposta:

Para ter uma visualização adequada, 6 simulações são realizadas:

```
set.seed(3)
par(mfrow = c(2, 3))
plot(rnorm(n=48),type='o')
plot(rnorm(n=48),type='o')
plot(rnorm(n=48),type='o')
plot(rnorm(n=48),type='o')
plot(rnorm(n=48),type='o')
plot(rnorm(n=48),type='o')
```



É possível observar que os dados gerados por uma distribuição normal se comportam de forma aleatória e não têm uma estrutura temporal, mostrando que os dados não dependem do tempo, ou em outras palavras, da sequência em que são geradas.

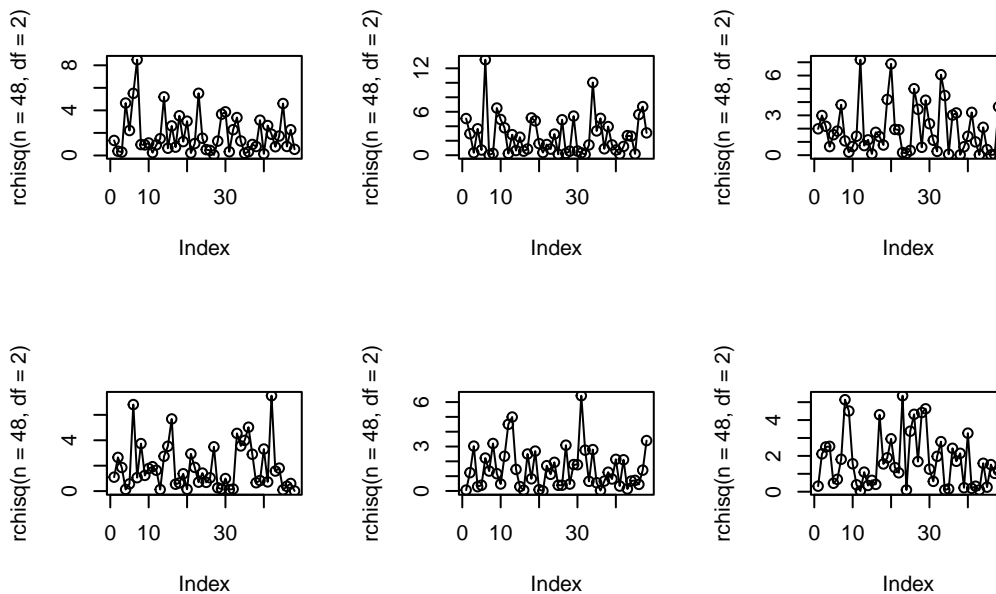
## Exercício 1.4

Simulate a completely random process of length 48 with independent, chi-square distributed values, each with 2 degrees of freedom. Display the time series plot. Does it look “random” and nonnormal? Repeat this exercise several times with a new simulation each time.

- Resposta:

Para ter uma visualização adequada, 6 simulações são realizadas:

```
set.seed(4)
par(mfrow = c(2, 3))
plot(rchisq(n=48,df=2),type='o')
plot(rchisq(n=48,df=2),type='o')
plot(rchisq(n=48,df=2),type='o')
plot(rchisq(n=48,df=2),type='o')
plot(rchisq(n=48,df=2),type='o')
plot(rchisq(n=48,df=2),type='o')
```



Assim como no exercício anterior, os dados gerados pela Qui Quadrado se mostram puramente aleatórios e não apresentam uma estrutura temporal.

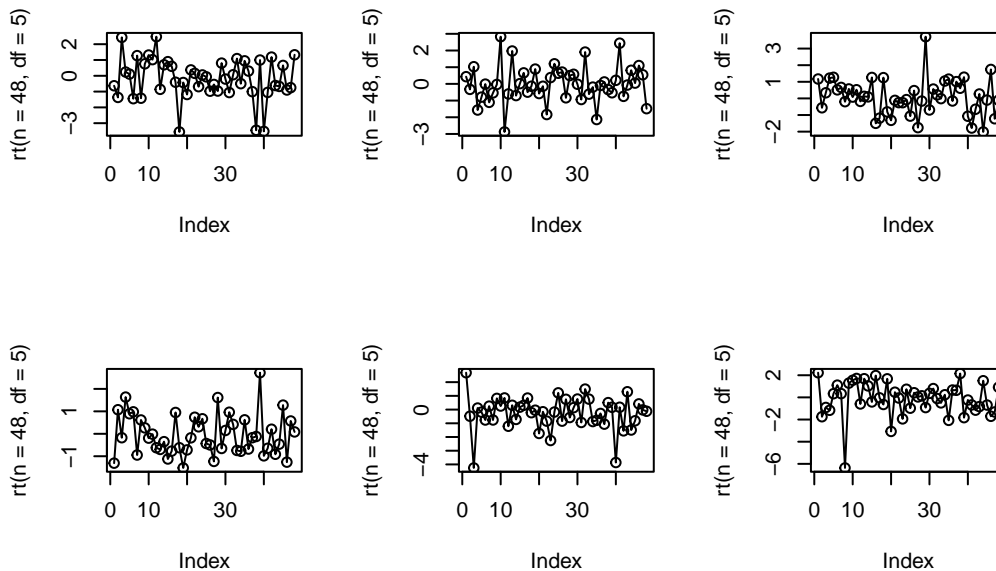
### Exercício 1.5

Simulate a completely random process of length 48 with independent, t-distributed values each with 5 degrees of freedom. Construct the time series plot. Does it look “random” and nonnormal? Repeat this exercise several times with a new simulation each time.

- Resposta:

Para ter uma visualização adequada, 6 simulações são realizadas:

```
set.seed(5)
par(mfrow = c(2, 3))
plot(rt(n=48,df=5),type='o')
plot(rt(n=48,df=5),type='o')
plot(rt(n=48,df=5),type='o')
plot(rt(n=48,df=5),type='o')
plot(rt(n=48,df=5),type='o')
plot(rt(n=48,df=5),type='o')
```



Complementando os dois exercícios anteriores, os dados gerados por uma T de Student se comportam aleatoriamente e não dependem de uma sequência de tempo.

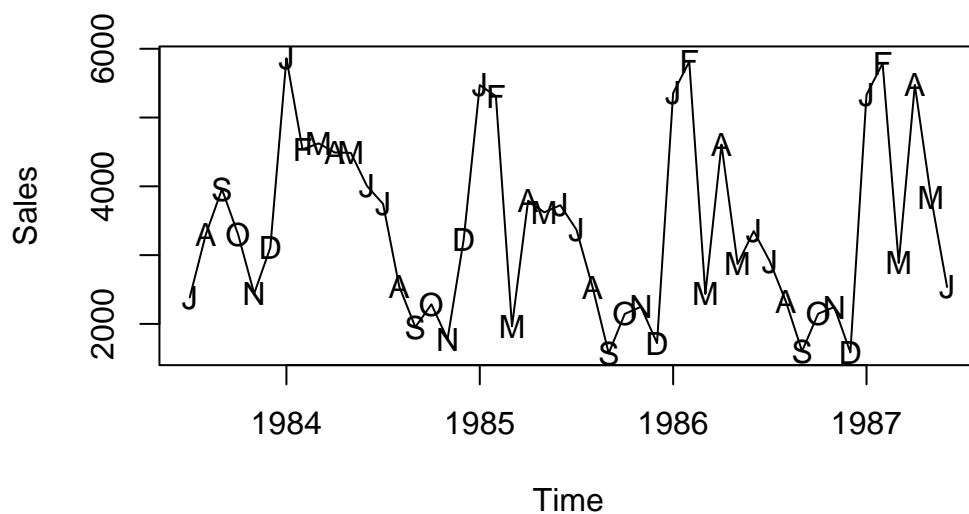
## Exercício 1.6

Construct a time series plot with monthly plotting symbols for the Dubuque temperature series as in Exhibit 1.9, on page 7. The data are in the file named tempdub.

- Resposta: Utilizando o código do livro:

```
data(oilfilters)
plot(oilfilters,type='l',ylab='Sales')
points(y=oilfilters,x=time(oilfilters), pch=as.vector(season(oilfilters)))
```





Agora utilizando o ggplot:

```
ggplot(data = data.frame(oilfilters),
       aes(x = time(oilfilters), y = as.vector(oilfilters))) +
  geom_line() +
  geom_text(aes(label = substr(as.character(season(oilfilters)), 1, 1)),
            check_overlap = FALSE) +
  labs(x = "Year", y = "Sales") +
  theme_minimal()
```

