

## ▼ Trabalho 1 - MO430

Patrick de Carvalho Tavares Rezende Ferreira - 175480

### ▼ Item 1

Abaixo, são importadas as duas primeiras listas fornecidas para execução dos testes, a1 e b1.

```
1 # Fixando a semente para termos números aleatórios capazes de serem reproduzidos
2 set.seed(1234)
3
4 a1=read.csv(file.path("/content/a1.csv"), header = FALSE)
5 a1=as.numeric(unlist(a1))
6
7 b1=read.csv(file.path("/content/b1.csv"), header = FALSE)
8 b1=as.numeric(unlist(b1))
```

O comprimento de cada uma das listas é exibido abaixo, onde fica claro que elas são de tamanho

```
1 print(paste0("Comprimento de a1: ", length(a1)))
2 print(paste0("Comprimento de b1: ", length(b1)))
```

```
↳ [1] "Comprimento de a1: 15"
   [1] "Comprimento de b1: 20"
```

Executamos agora os testes "t" e "Wilcoxon", os quais nos fornecem p-valores de, respectivamente

```
1 t.test(a1, b1)
2
3 wilcox.test(a1, b1)
```

```
↳
```

## ▼ 2

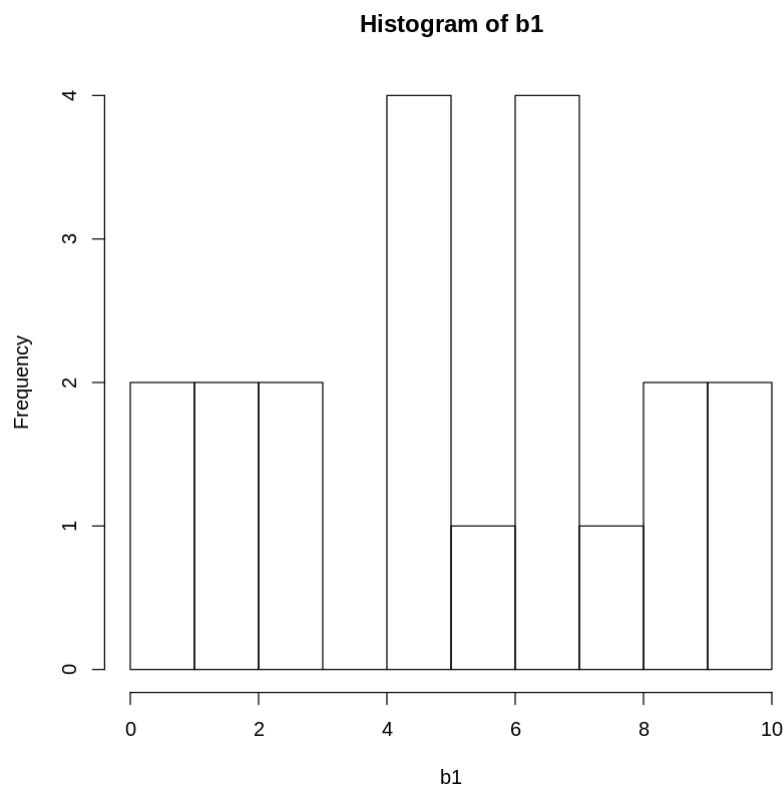
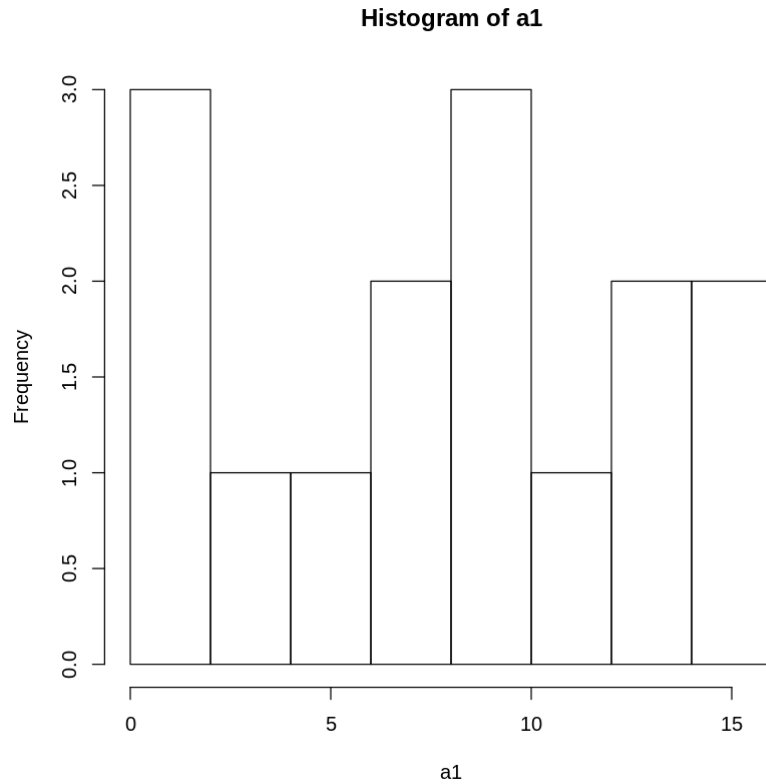
Geramos abaixo dois histogramas, um para cada lista sendo analisada, os quais nos evidenciam a origem gaussiana.

Não sendo dados amostrados de uma fonte gaussiana, não faz sentido aplicar o teste "t", que assume de pressupor a disponibilidade de ao menos 30 amostras. Ou seja, estes dados não satisfazem as condições para um teste paramétrico. Já o teste de "Wilcoxon" é um teste não paramétrico que pode ser utilizado neste caso, sendo aquele em que devemos confiar.

WILCOXON RANK SUM TEST

```
1 hist(a1, breaks=8)
2 hist(b1, breaks=8)
```





### ▼ 3

Abaixo, adquirimos as duas listas de valores pareados e verificamos seus comprimentos.

Executamos novamente os testes "t" e de "Wilconxon" para obter os p-valores. Dado o comprimento das amostras, para usar o teste "t", que assume que os dados têm pelo menos 30 amostras. Portanto, deve-se utilizar

como o teste não paramétrico de "Wilcoxon".

```
1 paired_data=read.csv(file.path("/content/paired.csv"), header = FALSE)
2 column1=as.numeric(unlist(paired_data[1]))
3 column2=as.numeric(unlist(paired_data[2]))
4
5 print(paste0("Comprimento de coluna1: ", length(column1)))
6 print(paste0("Comprimento de coluna2: ", length(column2)))
```

```
↳ [1] "Comprimento de coluna1: 10"
   [1] "Comprimento de coluna2: 10"
```

## ▼ 4

Abaixo executamos os testes "t" e de "Wilcoxon" nas suas versões pareadas e não pareadas. Sabemos que as não pareadas, pois fazem mais suposições acerca dos dados sendo tratadas e podem ser aplicadas em amostras maiores. Mesmo as versões pareadas produzindo p-valores menores, como se pode verificar abaixo, não são aplicadas em amostras inferior a 30, como é o caso atual.

```
1 t.test(column1, column2, paired = TRUE)
2
3 wilcox.test(column1, column2, paired = TRUE)
4
5 t.test(column1, column2, paired = FALSE)
6
7 wilcox.test(column1, column2, paired = FALSE)
```

```
↳
```

## Paired t-test

```
data: column1 and column2
t = 3.7366, df = 9, p-value = 0.00465
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.5445317 2.2154683
sample estimates:
mean of the differences
      1.38
```

## Wilcoxon signed rank test

```
data: column1 and column2
V = 52, p-value = 0.009766
alternative hypothesis: true location shift is not equal to 0
```

## Welch Two Sample t-test

```
data: column1 and column2
t = 1.5582, df = 14.856, p-value = 0.1402
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.5093412  3.2693412
sample estimates:
mean of x mean of y
    14.48    13.10
Warning message in wilcox.test.default(column1, column2, paired = FALSE):
"cannot compute exact p-value with ties"
```

## Wilcoxon rank sum test with continuity correction

```
data: column1 and column2
W = 67.5, p-value = 0.1984
alternative hypothesis: true location shift is not equal to 0
```

## ▼ 5

2 conjuntos de 15 dados amostrados de uma normal de media 10 e 13, ambos com desvio padrão

Média do p-valor usando o teste t para 50 repetições dos pares descritos acima

```
1 # Fixando a semente para ter números aleatórios capazes de serem reproduzidos.
2 set.seed(1234)
3
4 # 15 valores, media 10 e desvpad 5
5 a1 = rnorm(15, 10, 5)
6
7 # Fixando a semente para ter números aleatórios capazes de serem reproduzidos.
8 set.seed(4321)
9
10 # 15 valores, media 13 e desvpad 5
11 b1 = rnorm(15, 13, 5)
```

```

12
13 # Variavel usada para somar o p-valores antes de fazer a media
14 soma_p_value = 0
15 # Range de tamanho desejado para determinar o numero de iteracoes a realizar
16 seq_range = seq(from=1, to=50, by=1)
17 for (i in seq_range){
18     soma_p_value = soma_p_value + t.test(a1, b1)$p.value
19 }
20
21 # A media dos p_valores
22 print(paste0("Média dos p-valores: ", soma_p_value / length(seq_range)))

[1] "Média dos p-valores: 0.00295510269531462"

```

## ▼ 6

2 conjuntos de 15 dados amostrados de uma normal de media 10 e 13, ambos com desvio padrão 5  
Média do p-valor usando o teste t para 50 repetições dos pares acima, mas com 25 dados cada

```

1 # Fixando a semente para ter números aleatórios capazes de serem reproduzidos.
2 set.seed(1234)
3
4 # 15 valores, media 10 e desvpad 5
5 a1 = rnorm(15, 10, 5)
6
7 # Fixando a semente para ter números aleatórios capazes de serem reproduzidos.
8 set.seed(4321)
9
10 # 15 valores, media 13 e desvpad 5
11 b1 = rnorm(25, 13, 5)
12
13 # Variavel usada para somar o p-valores antes de fazer a media
14 soma_p_value = 0
15 # Range de tamanho desejado para determinar o numero de iteracoes a realizar
16 seq_range = seq(from=1, to=50, by=1)
17 for (i in seq_range){
18     soma_p_value = soma_p_value + t.test(a1, b1)$p.value
19 }
20
21 # A media dos p_valores
22 print(paste0("Média dos p-valores: ", soma_p_value / length(seq_range)))

[1] "Média dos p-valores: 0.0023720036502327"

```

## ▼ 7

2 conjuntos de 15 dados amostrados de uma normal de media 10 e 13, ambos com desvio padrão 5  
Média do p-valor usando o teste t para 50 repetições dos pares acima, com 15 dados cada mas com

```

1 # Fixando a semente para ter números aleatórios capazes de serem reproduzidos.

```

```
2 set.seed(1234)
3
4 # 15 valores, media 10 e desvpad 5
5 a1 = rnorm(15, 10, 5)
6
7 # Fixando a semente para ter números aleatórios capazes de serem reproduzidos.
8 set.seed(4321)
9
10 # 15 valores, media 13 e desvpad 5
11 b1 = rnorm(15, 13, 10)
12
13 # Variável usada para somar o p-valores antes de fazer a media
14 soma_p_value = 0
15 # Range de tamanho desejado para determinar o numero de iteracoes a realizar
16 seq_range = seq(from=1, to=50, by=1)
17 for (i in seq_range){
18     soma_p_value = soma_p_value + t.test(a1, b1)$p.value
19 }
20
21 # A media dos p_valores
22 print(paste0("Média dos p-valores: ", soma_p_value / length(seq_range)))
```

➡ [1] "Média dos p-valores: 0.0307690025017924"

## ▼ 8

2 conjuntos de 15 dados amostrados de uma normal de media 10 e 13, ambos com desvio padrão  
Média do p-valor usando o teste t para 50 repetições dos pares acima, com 15 dados, 5 de desvio

```
1 # Fixando a semente para ter números aleatórios capazes de serem reproduzidos.
2 set.seed(1234)
3
4 # 15 valores, media 10 e desvpad 5
5 a1 = rnorm(15, 10, 5)
6
7 # Fixando a semente para ter números aleatórios capazes de serem reproduzidos.
8 set.seed(4321)
9
10 # 15 valores, media 17 e desvpad 5
11 b1 = rnorm(15, 17, 5)
12
13 # Variável usada para somar o p-valores antes de fazer a media
14 soma_p_value = 0
15 # Range de tamanho desejado para determinar o numero de iteracoes a realizar
16 seq_range = seq(from=1, to=50, by=1)
17 for (i in seq_range){
18     soma_p_value = soma_p_value + t.test(a1, b1)$p.value
19 }
20
21 # A media dos p_valores
22 print(paste0("Média dos p-valores: ", soma_p_value / length(seq_range)))
```

```
[11] "Média dos p-valores: 3.82684184231002e-06"
```

Rich text editor toolbar with icons for bold, italic, link, unlink, list, and other formatting options.

## 9

Tomando como base o p-valor encontrado para o item 5, 0.00295510269531462, iremos comparar com os demais conjuntos modificados.

No item 6, para 25 dados em cada, o p-valor encontrado é maior em relação ao caso com 15 amostras de cada fonte, sendo de 0.0023720036502327. Isto indica que com mais amostras o teste pôde ter mais indícios de que os dados realmente não eram da mesma fonte, diminuindo ainda mais o p-valor.

No item 7, o p-valor encontrado foi maior que o do item 5, sendo de 0.0307690025017924. Isto é compreensível, já que, com um desvio padrão maior, o teste tem menos confiança para dizer que os dados são de fontes diferentes, já que a alternância de valores é maior e a divergência pode se dar devido a ruídos de amostragem.

No item 8, o p-valor encontrado é o menor de todos, sendo de 3.82684184231002e-06, pois as médias são muito mais distântes e fica mais evidente para o teste que estas não são iguais.

9

omando com  
amos compa

o item 6, par  
aso com 15  
dica que cor  
almente nãc

o item 7, o p  
0307690025  
maior, o teste  
ferentes, já c  
evido a ruído

o item 8, o p  
3.8268418423  
evidente para