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.

In [0]:

```
# Fixando a semente para termos números aleatórios capazes de serem reproduzidos.
set.seed(1234)

a1=read.csv(file.path("a1.csv"), header = FALSE)
a1=as.numeric(unlist(a1))

b1=read.csv(file.path("b1.csv"), header = FALSE)
b1=as.numeric(unlist(b1))
```

O comprimento de cada uma das listas é exibido abaixo, onde fica claro que elas são de tamanhos diferentes e, portanto, não pareadas.

In [68]:

```
print(paste0("Comprimento de a1: ", length(a1)))
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, 0.05724 e 0.06887.

In [69]:

```
t.test(a1, b1)

wilcox.test(a1, b1)
```

```
        Welch Two Sample t-test

data:  a1 and b1
t = 2.0123, df = 20.915, p-value = 0.05724
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.09871863  5.96056902
sample estimates:
mean of x mean of y
 7.995025  5.064100

        Wilcoxon rank sum test

data:  a1 and b1
W = 205, p-value = 0.06887
alternative hypothesis: true location shift is not equal to 0
```

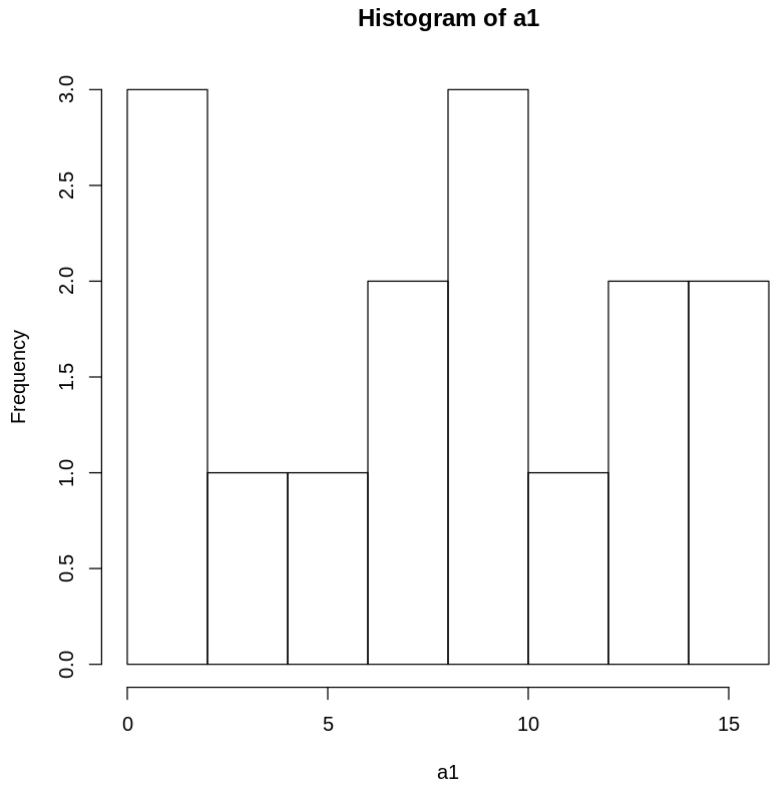
2

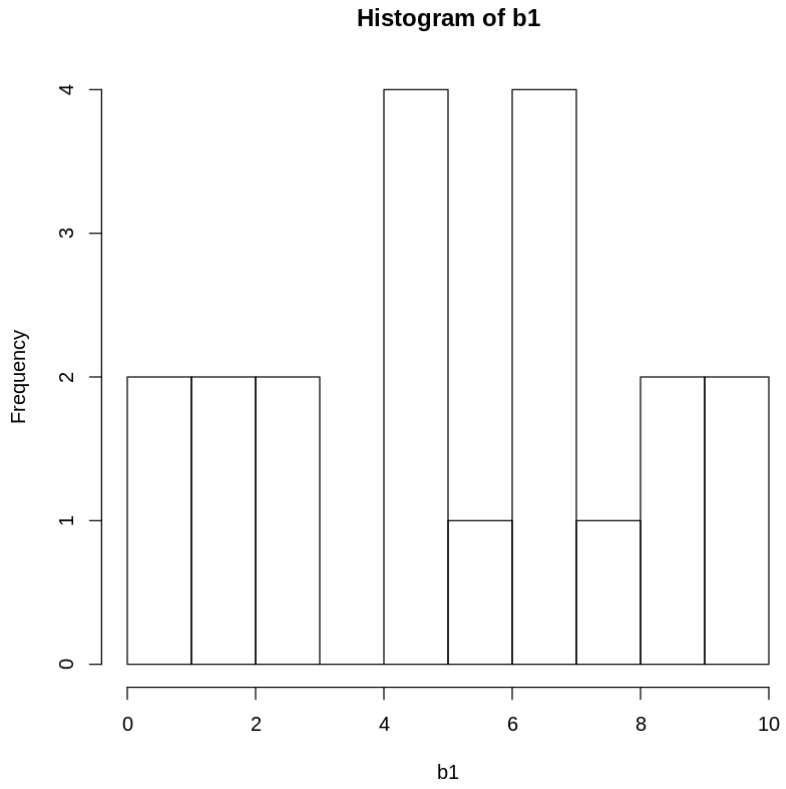
Geramos abaixo dois histogramas, um para cada lista sendo analisada, os quais nos evidenciam que estes dados não aparentam ser de origem gaussiana.

Não sendo dados amostrados de uma fonte gaussiana, não faz sentido aplicar o teste "t", que assume os dados tendo origem gaussiana, além de pressupor a disponibilidade de ao menos 30 amostras. Ou seja, estes dados não satisfazem as suposições do teste "t", que é um teste paramétrico. Já o teste de "Wilcoxon" é um teste não paramétrico que pode ser utilizado neste caso, tendo um p-valor provavelmente mais próximo da realidade e, portanto, sendo aquele em que devemos confiar.

In [70]:

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


Histogram of a1
Frequency
0.0
0.5
1.0
1.5
2.0
2.5
3.0
0
5
10
15
a1



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 (10 < 30), não deveríamos usar o teste "t", que assume que os dados têm pelo menos 30 amostras. Portanto, deve-se utilizar um teste que não faça essa suposição, como o teste não paramétrico de "Wilcoxon".

In [71]:

```
paired_data=read.csv(file.path("paired.csv"), header = FALSE)
column1=as.numeric(unlist(paired_data[1]))
column2=as.numeric(unlist(paired_data[2]))

print(paste0("Comprimento de coluna1: ", length(column1)))
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 versões pareadas são mais fortes que as não pareadas, pois fazem mais suposições acerca dos dados sendo tratadas e podem fazer cálculos mais adequados à entrada.

Mesmo as versões pareadas produzindo p-valores menores, como se pode verificar abaixo, não se deve utilizar o teste "t" para um número de amostras inferior a 30, como é o caso atual.

In [72]:

```
t.test(column1, column2, paired = TRUE)

wilcox.test(column1, column2, paired = TRUE)

t.test(column1, column2, paired = FALSE)

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 = FA
LSE):
“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 de 5.

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

In [73]:

```
# Fixando a semente para ter números aleatórios capazes de serem reproduzidos.
set.seed(1234)

# 15 valores, media 10 e desvpad 5
a1 = rnorm(15, 10, 5)

# Fixando a semente para ter números aleatórios capazes de serem reproduzidos.
set.seed(4321)

# 15 valores, media 13 e desvpad 5
b1 = rnorm(15, 13, 5)

# Variavel usada para somar o p-valores antes de fazer a media
soma_p_value = 0
# Range de tamamho desejado para determinar o numero de iteracoes a realizar
seq_range = seq(from=1, to=50, by=1)
for (i in seq_range){
    soma_p_value = soma_p_value + t.test(a1, b1)$p.value
}

# A media dos p_valores
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 de 5.

Média do p-valor usando o teste t para 50 repetições dos pares acima, mas com 25 dados cada

In [74]:

```
# Fixando a semente para ter números aleatórios capazes de serem reproduzidos.
set.seed(1234)

# 15 valores, media 10 e desvpad 5
a1 = rnorm(15, 10, 5)

# Fixando a semente para ter números aleatórios capazes de serem reproduzidos.
set.seed(4321)

# 15 valores, media 13 e desvpad 5
b1 = rnorm(25, 13, 5)

# Variavel usada para somar o p-valores antes de fazer a media
soma_p_value = 0
# Range de tamamho desejado para determinar o numero de iteracoes a realizar
seq_range = seq(from=1, to=50, by=1)
for (i in seq_range){
    soma_p_value = soma_p_value + t.test(a1, b1)$p.value
}

# A media dos p_valores
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 de 5.

Média do p-valor usando o teste t para 50 repetições dos pares acima, com 15 dados cada mas com 10 como desvio padrão

In [75]:

```
# Fixando a semente para ter números aleatórios capazes de serem reproduzidos.
set.seed(1234)

# 15 valores, media 10 e desvpad 5
a1 = rnorm(15, 10, 5)

# Fixando a semente para ter números aleatórios capazes de serem reproduzidos.
set.seed(4321)

# 15 valores, media 13 e desvpad 5
b1 = rnorm(15, 13, 10)

# Variavel usada para somar o p-valores antes de fazer a media
soma_p_value = 0
# Range de tamamho desejado para determinar o numero de iteracoes a realizar
seq_range = seq(from=1, to=50, by=1)
for (i in seq_range){
    soma_p_value = soma_p_value + t.test(a1, b1)$p.value
}

# A media dos p_valores
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 de 5.

Média do p-valor usando o teste t para 50 repetições dos pares acima, com 15 dados, 5 de desvio padrão mas com medias 10 e 17

In [76]:

```
# Fixando a semente para ter números aleatórios capazes de serem reproduzidos.
set.seed(1234)

# 15 valores, media 10 e desvpad 5
a1 = rnorm(15, 10, 5)

# Fixando a semente para ter números aleatórios capazes de serem reproduzidos.
set.seed(4321)

# 15 valores, media 17 e desvpad 5
b1 = rnorm(15, 17, 5)

# Variavel usada para somar o p-valores antes de fazer a media
soma_p_value = 0
# Range de tamamho desejado para determinar o numero de iteracoes a realizar
seq_range = seq(from=1, to=50, by=1)
for (i in seq_range){
    soma_p_value = soma_p_value + t.test(a1, b1)$p.value
}

# A media dos p_valores
print(paste0("Média dos p-valores: ", soma_p_value / length(seq_range)))
```

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

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 tiem 8, o p-valor encontrado é o menor de todos, sendo de 3.82684184231002e-06, pois as médias são muito mais distântes e fica ais evidente para o teste que estas não são iguais.