UNIVERSIDADE ESTADUAL DE CAMPINAS
INSTITUTE OF COMPUTING

SCIENTIFIC METHODOLOGY FOR COMPUTING
MO430

EXERCISE 3

STUDENT: Gian Franco Joel Condori Luna
RA: 234826

1.- Study of the factors that influence the p-value.

- **Generate 2 sets of 30 data sampled from a normal of mean 10 and 12, both with a standard deviation of 5.**

```
#Point 1
set.seed(1234)
data1 <- rnorm(30, 10, 5)
data2 <- rnorm(30, 12, 5)
data1
data2
```

```
>data1
[1]  3.964671 11.387146 15.422206 -1.728489 12.145623 12.530279
7.126300
 [8]  7.266841  7.177740  5.549811  7.614037  5.008068  6.118731
10.322294
[15] 14.797470  9.448573  7.444952  5.444023  5.814142 22.079176
10.670441
[22]  7.546571  7.797261 12.297947  6.531399  2.758975 12.873779
4.881721
[29]  9.924308  5.320257
>data2
[1] 17.511488  9.622035  8.452800  9.493710  3.854533  6.161904
1.099802
 [8]  5.295034 10.528531  9.670512 19.247481  6.656786  7.723177
10.596885
[15]  7.028300  7.157428  6.463409  5.740071  9.380859  9.515750
2.969844
[22]  9.089620  6.455552  6.925190 11.188452 14.815279 20.239087
8.133233
[29] 20.029548  6.210957
```

- **Calculate the average p-value using the t-test for 50 repetitions of the pairs described above. (The idea of averaging several p-values is just to make us**

more sure that what we're going to study doesn't depend so much on luck when you generate the 30 samples for each group).

```
#Point 2
sum <- 0
for (i in 1:50) {
  test=t.test(rnorm(30,10,5),rnorm(30,12,5))
  print(paste(i,":",round(test$p.value,3)))
  sum <- sum +test$p.value
}
media <- sum/50
print(media)
```

```
[1] "1 : 0.123"
[1] "2 : 0.284"
[1] "3 : 0.071"
[1] "4 : 0.171"
[1] "5 : 0.5"
[1] "6 : 0.219"
[1] "7 : 0.529"
[1] "8 : 0.161"
[1] "9 : 0.328"
[1] "10 : 0"
[1] "11 : 0.232"
[1] "12 : 0.003"
[1] "13 : 0"
[1] "14 : 0.064"
[1] "15 : 0.558"
[1] "16 : 0.107"
[1] "17 : 0.402"
[1] "18 : 0.783"
[1] "19 : 0.377"
[1] "20 : 0.412"
[1] "21 : 0.023"
[1] "22 : 0.196"
[1] "23 : 0.027"
[1] "24 : 0.145"
[1] "25 : 0.005"
[1] "26 : 0.706"
[1] "27 : 0.31"
[1] "28 : 0.274"
[1] "29 : 0.001"
[1] "30 : 0.64"
[1] "31 : 0.255"
[1] "32 : 0.042"
[1] "33 : 0.172"
```

```
[1] "34 : 0.023"
[1] "35 : 0.004"
[1] "36 : 0.004"
[1] "37 : 0.031"
[1] "38 : 0.784"
[1] "39 : 0.324"
[1] "40 : 0.957"
[1] "41 : 0.109"
[1] "42 : 0.009"
[1] "43 : 0.129"
[1] "44 : 0.203"
[1] "45 : 0.054"
[1] "46 : 0.043"
[1] "47 : 0.216"
[1] "48 : 0.28"
[1] "49 : 0.485"
[1] "50 : 0.19"
```

```
> print(media)
[1] 0.239256
```

- **Calcule a media do p-valor para o teste t para 50 repetições dos pares acima, mas com 60 dados cada.**

```
#Point 3
sum <- 0
for (i in 1:50) {
  test=t.test(rnorm(60,10,5),rnorm(60,12,5))
  print(paste(i,":",round(test$p.value,3)))
  sum <- sum +test$p.value
}
media <- sum/50
print(media)
```

```
> print(media)
[1] 0.071907
```

- **Calculate the mean p-value for the t-test for 50 repetitions of the above pairs, with 30 dice each but with 10 as standard deviation.**

```
#Point 4
sum <- 0
for (i in 1:50) {
  test=t.test(rnorm(30,10,10),rnorm(30,12,10))
  print(paste(i,":",round(test$p.value,3)))
  sum <- sum +test$p.value
}
media <- sum/50
print(media)
```

```
> print(media)
[1] 0.4881853
```

- **Calculate the mean p-value for the t-test for 50 replicates of the above pairs, with 30 dice, 5 standard deviations but with means 10 and 15.**

```
#Point 5
sum <- 0
for (i in 1:50) {
  test=t.test(rnorm(30,10,5),rnorm(30,15,5))
  print(paste(i,":",round(test$p.value,3)))
  sum <- sum +test$p.value
}
media <- sum/50
print(media)
```

```
> print(media)
[1] 0.008091162
```

- **Discuss the influence of the 3 factors on the p-value: number of data, data noise (the standard deviation of sources) and difference between source means**

  We realize that:
  - When the amount of data increases the mean p-value decreases (`0.239256 a 0.071907`).
  - When the standard deviation increases the mean p-value (`0.239256 a 0.4881853`).
  - When the mean increases the mean p-value decreases (`0.239256 a 0.008091162`).

- **Run the examples above using the Wilcoxon rank-sum and show that (probably) the effects you found on the T-test are the same for Wilcoxon. This is to show that these p-value effects do not depend on the test itself but are properties of the p-value concept.**

```r
#Point 2
sum <- 0
for (i in 1:50) {
  wtest=wilcox.test(rnorm(30,10,5),rnorm(30,12,5))
  print(paste(i,":",round(wtest$p.value,3)))
  sum <- sum + wtest$p.value
}
media <- sum/50
print(media)
```

```
>print(media)
[1] 0.1904719
```

```r
#Point 3
sum <- 0
for (i in 1:50) {
  wtest=wilcox.test(rnorm(60,10,5),rnorm(60,12,5))
  print(paste(i,":",round(wtest$p.value,3)))
  sum <- sum + wtest$p.value
}
media <- sum/50
print(media)
```

```
>print(media)
[1] 0.1010471
```

```r
#Point 4
sum <- 0
for (i in 1:50) {
  wtest=wilcox.test(rnorm(30,10,10),rnorm(30,12,10))
  print(paste(i,":",round(wtest$p.value,3)))
  sum <- sum + wtest$p.value
}
media <- sum/50
print(media)
```

```
>print(media)
[1]  0.4034574
```

```r
#Point 5
sum <- 0
for (i in 1:50) {
  wtest=wilcox.test(rnorm(30,10,5),rnorm(30,15,5))
  print(paste(i,":",round(wtest$p.value,3)))
  sum <- sum + wtest$p.value
}
media <- sum/50
print(media)
```

```
>print(media)
[1] 0.007113554
```

Conclusion: There are slight variations in the results compared to the function "t.test ()".

t.test	wilcox.test
0.239256	0.1904719
0.071907	0.1010471
0.4881853	0.4034574
0.008091162	0.007113554

2.- Confidence Interval

Generate the confidence interval for the blood pressure of patients with diabetes and without diabetes (95% confidence interval)

```
#Read file
data <-
read.csv("~/Documentos/github/maestriaCS/metodoligia_investigacion_computacion/exercise_3//ex2.csv", stringsAsFactors = FALSE)
#Seed
set.seed(1234)

#Separate data by people who have diabetes and people who do not have diabetes
si_diabetes <- data$bp[data$type=="Yes"]
no_diabetes <- data$bp[data$type=="No"]
```

TEST T:

```
t.test(si_diabetes)
#     One Sample t-test
```

```
#data:  si_diabetes
#t = 53.097, df = 67, p-value < 2.2e-16
#alternative hypothesis: true mean is not equal to 0
#95 percent confidence interval:
# 71.78434 77.39213
#sample estimates:
#mean of x
# 74.58824
```

```
t.test(no_diabetes)
#     One Sample t-test
#
#data:  no_diabetes
#t = 72.09, df = 131, p-value < 2.2e-16
#alternative hypothesis: true mean is not equal to 0
#95 percent confidence interval:
# 67.63705 71.45386
#sample estimates:
#mean of x
# 69.54545
```

WILCOXON RANK SUM:

```
wilcox.test(si_diabetes,conf.int=T)
#     Wilcoxon signed rank test with continuity correction
#
#data:  si_diabetes
#V = 2346, p-value = 7.552e-13
#alternative hypothesis: true location is not equal to 0
#95 percent confidence interval:
# 71.99996 77.00002
#sample estimates:
#(pseudo)median
#     74.99998
```

```
wilcox.test(no_diabetes,conf.int=T)
#     Wilcoxon signed rank test with continuity correction
#
#data:  no_diabetes
#V = 8778, p-value < 2.2e-16
#alternative hypothesis: true location is not equal to 0
#95 percent confidence interval:
# 67.50002 71.00003
```

```
#sample estimates:
#(pseudo)median
 #    69.00005
```

BOOTSTRAP:

```r
library(boot)
auxf <- function(dado,indice){
  return(mean(dado[indice]))
}
bb = boot(si_diabetes,R=5000, statistic=auxf)
boot.ci(bb,type="bca")

#BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
#Based on 5000 bootstrap replicates
#
#CALL :
#boot.ci(boot.out = bb, type = "bca")
#
#Intervals :
#Level      BCa
#95%   (71.96, 77.28 )
#Calculations and Intervals on Original Scale
```

```r
library(boot)
auxf <- function(dado,indice){
  return(mean(dado[indice]))
}
bb = boot(no_diabetes,R=5000, statistic=auxf)
boot.ci(bb,type="bca")

#BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
#Based on 5000 bootstrap replicates
#
#CALL :
#boot.ci(boot.out = bb, type = "bca")
#
#Intervals :
#Level      BCa
#95%   (67.61, 71.40 )
#Calculations and Intervals on Original Scale
```

3.- Interception of confidence intervals

Using the same technique to calculate the confidence interval (one of the 3 above), is there an intersection between the confidence intervals of the 2 datasets? Does this agree with the test of significant difference between them?

```
set.seed(1234)
t.test(si_diabetes, no_diabetes)
#     Welch Two Sample t-test
#
#data:  si_diabetes and no_diabetes
#t = 2.9592, df = 130.28, p-value = 0.003665
#alternative hypothesis: true difference in means is not equal to 0
#95 percent confidence interval:
# 1.671482 8.414080
#sample estimates:
#mean of x mean of y
# 74.58824  69.54545
```

Answer: There is no intersection between the confidence intervals of the data sets (yes_diabetes and no_diabetes).

4.- Effect Size

```
install.packages("effsize")
library(effsize)
cohen.d(si_diabetes, no_diabetes, na.rm = T, pooled = T)
#Cohen's d
#
#d estimate: 0.4480343 (small)
#95 percent confidence interval:
#     lower       upper
#0.1503741 0.7456945
```

Answer: There is no significant difference since the value of the estimated distance is short: 0.4480343.