TP1\_R\_tainaKERRIOU.R

taina

2022-04-01

#TP-RStudio   
  
#1  
year=365  
#2  
year

## [1] 365

#3  
is.numeric(year)

## [1] TRUE

#4  
year+35

## [1] 400

#5  
y=seq(1,20,by=0.8)  
#6  
history=c(14,18,19)  
physical=c(9,10,11)  
notes=data.frame(history,physical)  
rownames(notes)<-c("Peter", "Magali", "John")  
notes

## history physical  
## Peter 14 9  
## Magali 18 10  
## John 19 11

#Exercice 1  
#importation du fishier excel  
library(readxl)  
Heures\_travail\_UE2020 <- read\_excel("R/Heures\_travail\_UE2020.xlsx")  
View(Heures\_travail\_UE2020)  
median

## function (x, na.rm = FALSE, ...)   
## UseMethod("median")  
## <bytecode: 0x0000000012d4ddd8>  
## <environment: namespace:stats>

#1)  
#median,1Q and 3Q, max and min  
summary(Heures\_travail\_UE2020$HEURES\_PAR\_AN)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1332 1454 1577 1584 1702 1834

#3)  
dispersion = max(Heures\_travail\_UE2020$HEURES\_PAR\_AN) - min(Heures\_travail\_UE2020$HEURES\_PAR\_AN)  
dispersion

## [1] 502.3

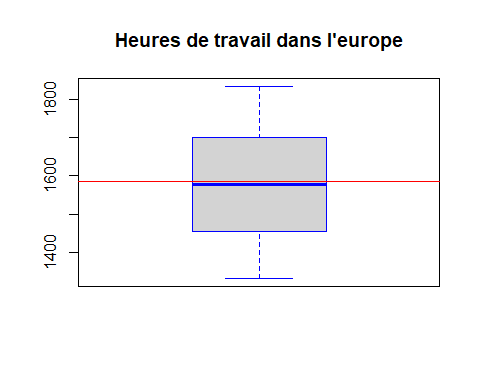
sd(Heures\_travail\_UE2020$HEURES\_PAR\_AN)

## [1] 148.9454

IQR(Heures\_travail\_UE2020$HEURES\_PAR\_AN)

## [1] 247.5

#4)  
moy<- mean(Heures\_travail\_UE2020$HEURES\_PAR\_AN)#moyenne   
boxplot(Heures\_travail\_UE2020$HEURES\_PAR\_AN,main="Heures de travail dans l'europe",border=c("blue"),names=c("UE"))  
abline(h=moy,col = "red")



#The mean is higher than the median 0.3806434 % of the max values which is 1834  
abs(1577 - 1583.981)

## [1] 6.981

pourcentage1 = 6.981/ max(Heures\_travail\_UE2020$HEURES\_PAR\_AN)  
pourcentage1 \*100

## [1] 0.3806434

#The distribution is well distributed between Q1,Q3 and the median, the extent is high in relation to the values   
#There is no extreme point in the distribution.  
  
#Manipulation of vectors :   
  
vec<-c(50,1,25,80,10)  
vec[1] #we have print the first value of the vector

## [1] 50

length(vec)

## [1] 5

mx = max(vec)  
mx

## [1] 80

which.max(vec) #the maximum value is located at the 4th place

## [1] 4

vec = vec[! vec %in% 50]#we have delete the number 1 of the vector  
vec

## [1] 1 25 80 10

#Exercice 2  
#import the data from decathlon  
library(readxl)  
DECATHLON <- read\_excel("C:/Users/taina/Downloads/DECATHLON.xlsx")  
View(DECATHLON)  
  
summary(DECATHLON)

## ID C100A C100B Slong   
## Length:34 Min. :10.62 Min. :10.32 Min. :5.830   
## Class :character 1st Qu.:11.02 1st Qu.:11.00 1st Qu.:6.985   
## Mode :character Median :11.18 Median :11.09 Median :7.085   
## Mean :11.22 Mean :11.08 Mean :7.095   
## 3rd Qu.:11.45 3rd Qu.:11.33 3rd Qu.:7.367   
## Max. :12.12 Max. :12.34 Max. :7.720   
## Lpoids Shaut C400 C110   
## Min. : 9.71 Min. :1.700 Min. :47.44 Min. :14.18   
## 1st Qu.:12.99 1st Qu.:1.940 1st Qu.:48.35 1st Qu.:14.73   
## Median :14.05 Median :1.970 Median :49.21 Median :15.00   
## Mean :13.85 Mean :1.974 Mean :49.37 Mean :15.11   
## 3rd Qu.:14.86 3rd Qu.:2.030 3rd Qu.:49.99 3rd Qu.:15.39   
## Max. :16.60 Max. :2.270 Max. :52.32 Max. :17.05   
## Ldisq Sperch Ljave C1500 Score   
## Min. :27.10 Min. :2.600 Min. :39.10 Min. :256.6 Min. :5339   
## 1st Qu.:38.88 1st Qu.:4.525 1st Qu.:55.06 1st Qu.:266.7 1st Qu.:7532   
## Median :42.09 Median :4.700 Median :58.66 Median :272.4 Median :7864   
## Mean :41.91 Mean :4.676 Mean :58.84 Mean :276.2 Mean :7783   
## 3rd Qu.:44.69 3rd Qu.:4.900 3rd Qu.:63.75 3rd Qu.:285.9 3rd Qu.:8177   
## Max. :50.66 Max. :5.700 Max. :72.60 Max. :303.2 Max. :8488   
## Age Club   
## Length:34 Length:34   
## Class :character Class :character   
## Mode :character Mode :character   
##   
##   
##

#Know how to create a vector from a data :  
Liste\_des\_score <- c(DECATHLON$Score)  
Liste\_des\_score

## [1] 7753 7860 6907 5339 8143 8083 7579 8093 8286 8036 8189 7859 8021 7505 8180  
## [16] 8216 8114 8328 8167 8488 7310 7623 7517 7743 7231 7869 7016 7745 7237 7422  
## [31] 8306 8272 8399 7781

#Know how to find the number of rows, in the source table, of values according to a criterion:  
which(DECATHLON$Lpoids >14, arr.ind=TRUE) #Find all the line where the poids is bigger than 14

## [1] 2 5 10 11 13 15 16 17 18 19 20 24 26 30 31 33 34

#Find the line number of the highest score achieved  
which.max(DECATHLON$Score)

## [1] 20

#Know how to calculate statistical indicators using the data of a data frame according to a criterion  
  
#Calculate the average and standard deviation of the Scores only of the athletes of the club decaRennes then only of the athletes of the club ParisII  
  
mean(DECATHLON$Score[DECATHLON$Club=='ParisII'])

## [1] 7784.824

mean(DECATHLON$Score[DECATHLON$Club=='decaRennes'])

## [1] 7780.882

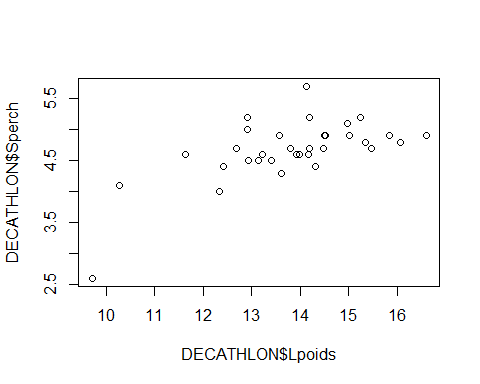
#Count (with the length() function) how many ParisII CLUB ATHLETES AND (use the sign & ) who have scores above 8000 points`  
  
length(DECATHLON$Score[DECATHLON$Score >8000 & DECATHLON$Club=='ParisII'])

## [1] 7

#Create a chart from the values of a column in a data frame  
#Make a scatterplot (fonction plot()) of the Lpoids against Sperch (x against y)  
  
  
plot(x = DECATHLON$Lpoids, y= DECATHLON$Sperch)  
#Find the minimum with which  
  
which.min( DECATHLON$Lpoids)

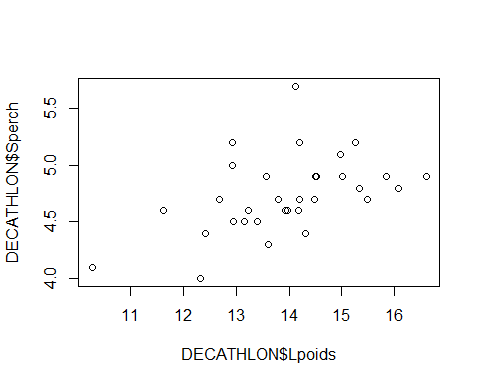
## [1] 4

#Remove the value   
DECATHLON$Lpoids = DECATHLON$Lpoids[! DECATHLON$Lpoids %in% 4]  
  
plot(x = DECATHLON$Lpoids, y= DECATHLON$Sperch)  
identify(DECATHLON$Lpoids,DECATHLON$Sperch,labels = DECATHLON$Club)

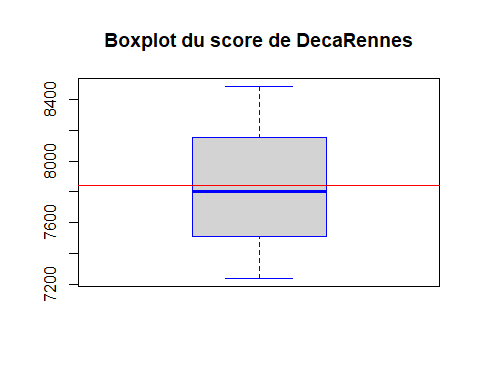


## integer(0)

## integer(0)  
DECATHLON = DECATHLON[-4,]  
  
plot(x = DECATHLON$Lpoids, y= DECATHLON$Sperch)



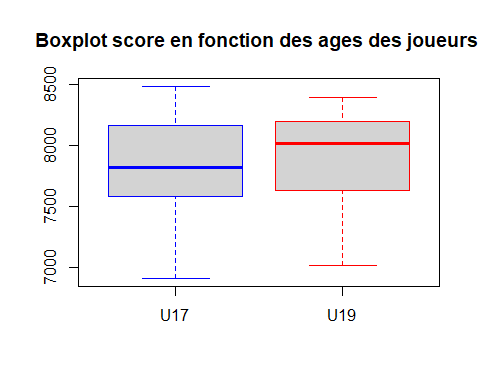
#Sort a data frame in ascending order by the values in a column:  
  
DECATHLON$Score<- sort(DECATHLON$Score) # the function sort, sort the value of the Score column .  
  
#in the DECATLHON data frame, make a boxplot of the scores of the decaRennes team.  
  
boxplot(DECATHLON$Score[DECATHLON$Club=='decaRennes'],main="Boxplot du score de DecaRennes",border=c("blue"),names=c("UE"))  
moy2 = mean(DECATHLON$Score[DECATHLON$Club=='decaRennes'])  
abline(h = moy2, col = "red")



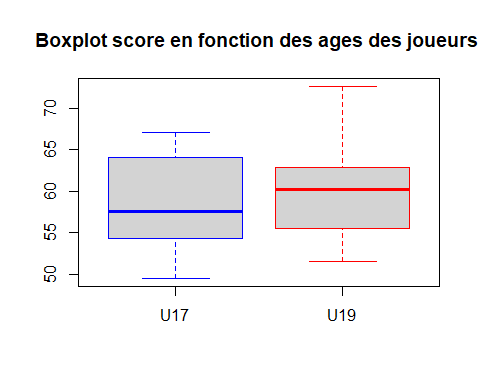
#Know how to count how many cases there are in each modality of a factor:  
table(DECATHLON$Club=='decaRennes') #16 True donc 16 iteration

##   
## FALSE TRUE   
## 17 16

#Exercise : Analysis by age category  
  
boxplot(DECATHLON$Score[DECATHLON$Age=='U17'],DECATHLON$Score[DECATHLON$Age=='U19'],main="Boxplot score en fonction des ages des joueurs ",border=c("blue","red"),names=c("U17","U19"))



#It scores more in the U17 because the range is larger, the distributions are not homogeneous on both sides   
#For the U19, the boxplot is desequilibred on the top,the gap between the max and the Q3 is small. The is close because 25% of the distribution is in this gap.  
#For the U17, the gap between the Min and the Q1 is as big as the gap between the Q3 and the Q1 representing 50% of the data   
#The U19 average is much higher than the U17 average. So the best athelthes have 19 years old so the score is probably related to age   
  
#Study of age complementary javelot   
  
boxplot(DECATHLON$Ljave[DECATHLON$Age=='U17'],DECATHLON$Ljave[DECATHLON$Age=='U19'],main="Boxplot score en fonction des ages des joueurs ",border=c("blue","red"),names=c("U17","U19"))



#We can see that the 19 years old player is globally better than the 17 years old players  
#But there is more guy at 17 years old that is better than the mean. (The Q3 for U17 is higher than Q3 for U19)  
#There is one player in the 19 years old that is an extreme point. his score his strangly better than all the others players