TP3_R_Tainakerriou.R

taina

2022-05-04

```
## TP3

##TAINA KERRIOU



library(readxl)
library(ggplot2)

#Loi Binomial B(n,p) (r=,p<=,q (X<a)=?)binom(m,size=n,prob=p)
#P(X = 3 )
dbinom(3,10,0.5)

## [1] 0.1171875

#P(X ≤ 3 )
pbinom(3,10,0.5)

## [1] 0.171875

#P(X < 3)
pbinom(2,10,0.5)

## [1] 0.0546875

#P(X ≥ 3)
1-pbinom(3,10,0.5)

## [1] 0.828125

#P(X > 3)
1-pbinom(2,10,0.5)

## [1] 0.9453125

#Affiche toutes les proba de 0 à 10
dbinom(0:10,10,0.5)

##  [1] 0.0009765625 0.0097656250 0.0439453125 0.1171875000 0.2050781250
##  [6] 0.2460937500 0.2050781250 0.1171875000 0.0439453125 0.0097656250
## [11] 0.0009765625

#Les formate sous forme de colonne
as.data.frame(dbinom(0:10,10,0.5))
```

```r
##    dbinom(0:10, 10, 0.5)
## 1           0.0009765625
## 2           0.0097656250
## 3           0.0439453125
## 4           0.1171875000
## 5           0.2050781250
## 6           0.2460937500
## 7           0.2050781250
## 8           0.1171875000
## 9           0.0439453125
## 10          0.0097656250
## 11          0.0009765625
```

```r
##Loi Poisson lambda=3 (p=<,d=)pois(x,lambda)
#P(X ≤ 4)
ppois(4,3)
```

```
## [1] 0.8152632
```

```r
#P(X = 2)
dpois(2,3)
```

```
## [1] 0.2240418
```

```r
##Loi Normale (x,expectation,standard deviation)
#(X ≤ 170)
pnorm(170,200,10)
```

```
## [1] 0.001349898
```

```r
#P(X < 170)
pnorm(170,200,10)
```

```
## [1] 0.001349898
```

```r
#P(X = 220)
dnorm(220,200,10)
```

```
## [1] 0.005399097
```

```r
#P(X > 220)
1-pnorm(220,200,10)
```

```
## [1] 0.02275013
```

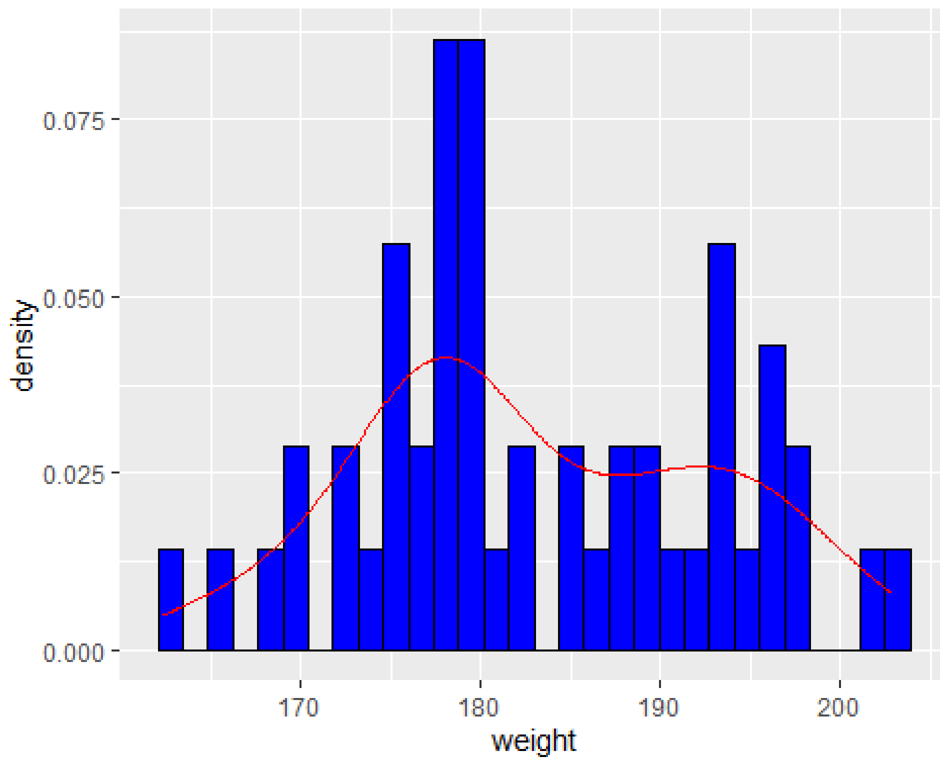
```r
#P(190 < X < 210)
pnorm(210,200,10)-pnorm(190,200,10)
```

```
## [1] 0.6826895
```

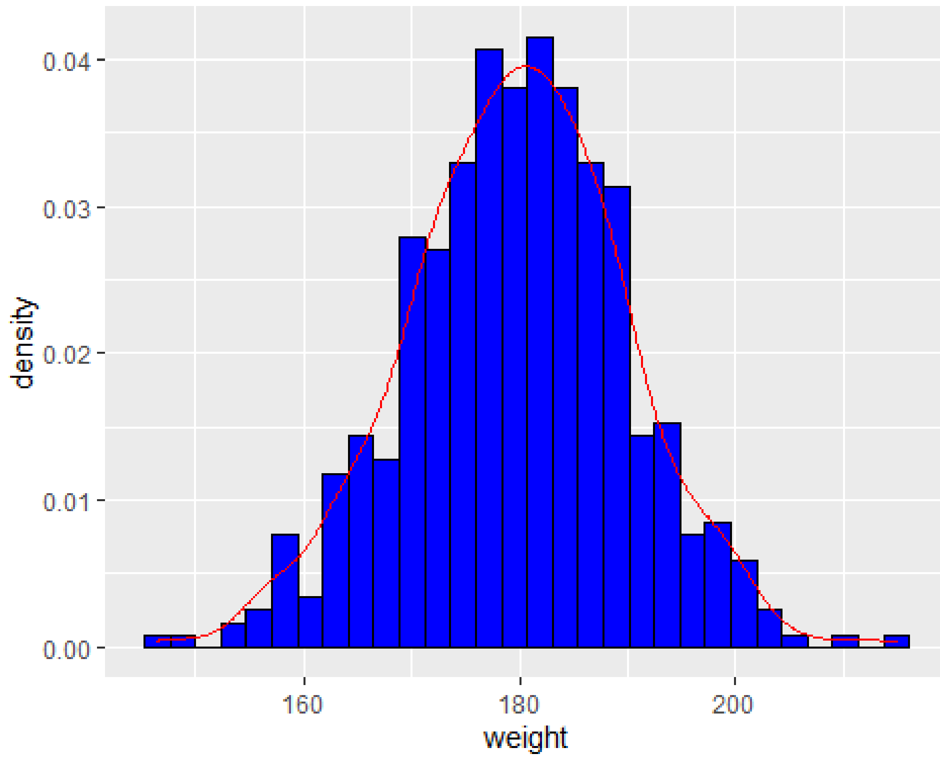
```r
##EXERCICE 1
library(readxl)
library(readxl)
```

```
database1 <- read_excel("C:/Users/taina/Downloads/database1.xlsx")
View(database1)
#Calculate the average of each column
apply(X = database1, MARGIN = 2, FUN = mean)

##  série 1  série 2  série 3
## 17.66667 21.00000 15.00000

database1_MEANS=as.data.frame(apply(X = database1, MARGIN = 2, FUN = mean))
#replicate(n, expr) to execute an expression n-times
##EXERCICE2
#génère n échantillon aléatoire rlaw(n,paramètre)
replicate(5,rnorm(10,0,1))

##              [,1]       [,2]       [,3]        [,4]          [,5]
##  [1,]  0.85807823 -1.9784181 -0.3582069  0.05287047  0.0004212495
##  [2,] -0.09011056 -0.1251514 -0.9329009  0.51944360 -2.9895850434
##  [3,] -0.48531560 -2.6595329  0.3721963 -0.66570848 -0.0310437024
##  [4,] -1.08668504  1.1294700 -0.1762655  2.49096058  1.0445572324
##  [5,] -0.36989082 -0.9659679  0.3962826  0.57816533  1.0741477315
##  [6,]  0.58273869 -1.1112653  1.1362763 -1.84835036 -0.2346714759
##  [7,] -1.16252456  1.4193457 -0.2877474 -1.00483932 -0.5652523666
##  [8,]  0.48144230  0.2048949 -0.1695070  0.34721626 -1.3076846462
##  [9,] -1.00412446 -0.6942811 -0.7141130  1.04187743 -0.3950297460
## [10,] -1.12769408  0.5115117 -0.6502513  0.95988302 -1.5278468216

##EXERCICE 3
#Add a column to a dataframe
#cbind(array, vector to be added in column)
Série4=c(12,14,18)
database2<-cbind(database1,Série4)
sample1=c(20,22,24,22)
database3<-rbind(database2,sample1)
##EXERCICE 4
#show 8 value of a dataframe
print.data.frame(database3, max=8)

##   série 1 série 2 série 3 Série4
## 1      15      18      10     12
## 2      18      22      15     14
##  [ reached 'max' / getOption("max.print") -- omitted 2 rows ]

#générer des valeurs
data=runif(50,-1,1)
mean(data)

## [1] 0.05982313

data2=replicate(20,data)
data3=as.data.frame(apply(X = data2, MARGIN = 2, FUN = mean))
data4=as.data.frame(apply(X = data2, MARGIN = 2, FUN = sd))
dataframe=cbind(data3,data4)
```

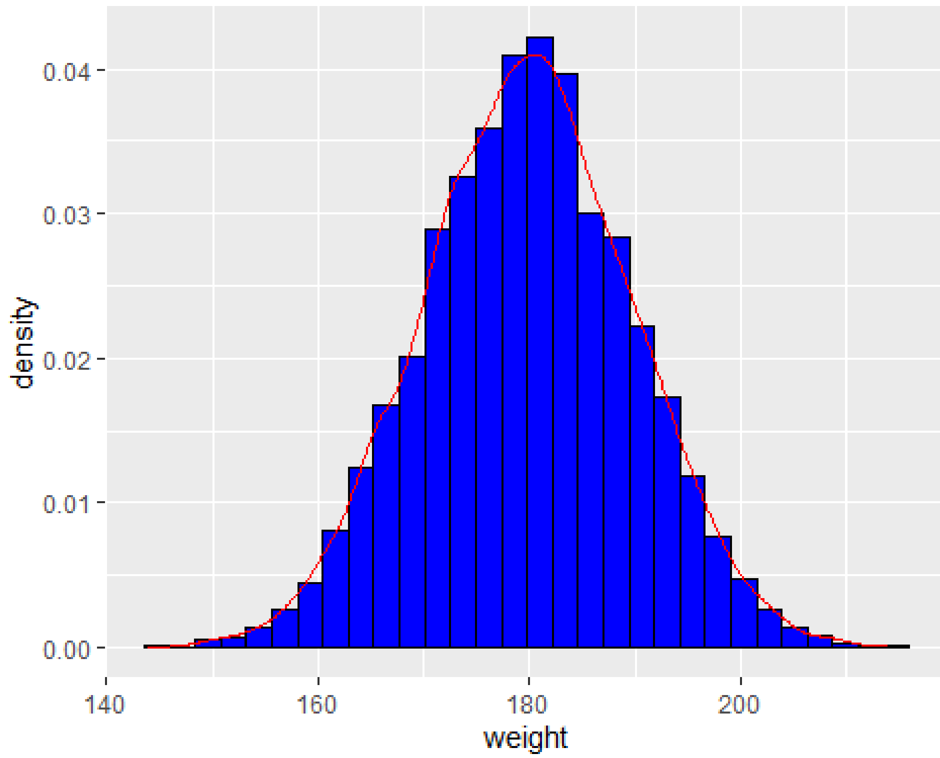
```r
##EXERCICE 5: Package reshape
MaBDD=data.frame(ID=c(1:3),SERIE1=c(5,5,5),SERIE2=c(10,10,10))
#reshape2 allows us to reformate easly our dataframe
library(reshape2)
MaBDDreshape=melt(MaBDD,id.vars='ID',measure.vars=c('SERIE1','SERIE2'))
##Normal law application
library(ggplot2)
#n=50
variety1=as.data.frame(rnorm(50,180,10))
colnames(variety1)="weight"
ggplot(variety1,aes(x=weight,y=..density..))+
  geom_histogram(fill='blue',color='black')+
    geom_density(color='red')

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



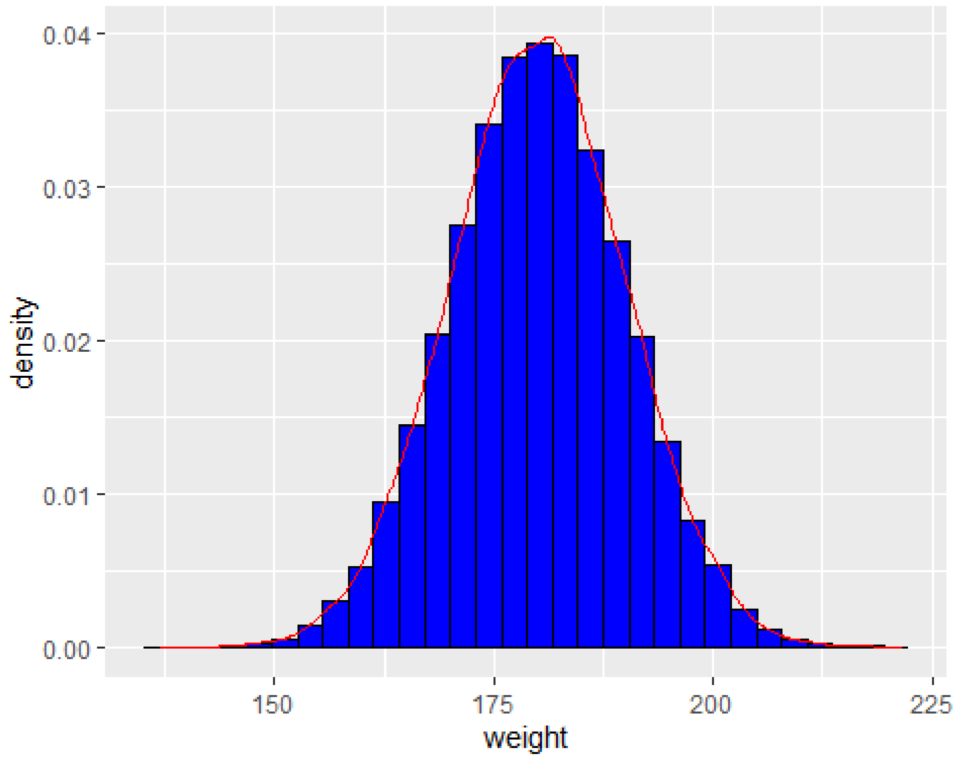

```r
#n=500
variety500=as.data.frame(rnorm(500,180,10))
colnames(variety500)="weight"
ggplot(variety500,aes(x=weight,y=..density..))+
  geom_histogram(fill='blue',color='black')+
  geom_density(color='red')

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
#n=5000
variety5000=as.data.frame(rnorm(5000,180,10))
colnames(variety5000)="weight"
ggplot(variety5000,aes(x=weight,y=..density..))+
  geom_histogram(fill='blue',color='black')+
  geom_density(color='red')

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

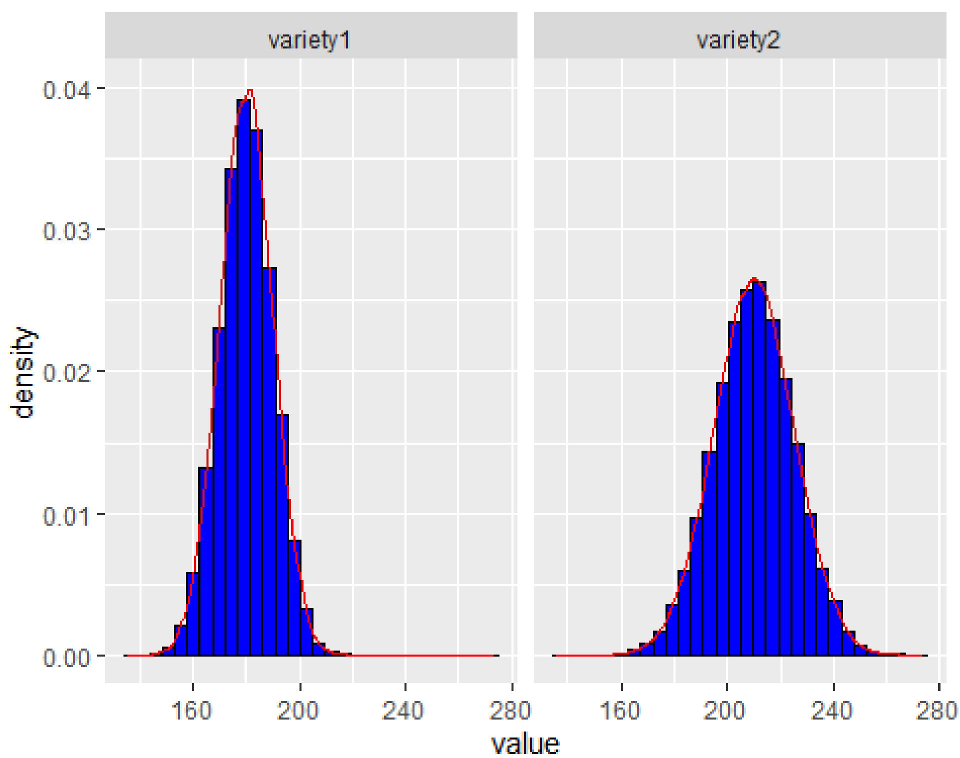



```
#n=50000
variety50000=as.data.frame(rnorm(50000,180,10))
colnames(variety50000)="weight"
ggplot(variety50000,aes(x=weight,y=..density..))+
  geom_histogram(fill='blue',color='black')+
  geom_density(color='red')

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
lebeau=cbind(variety1,variety500,variety5000,variety50000)
meanlebeau=as.data.frame(apply(X = lebeau, MARGIN = 2, FUN = mean))
colnames(meanlebeau)="meanN"
sDNlebeau=as.data.frame(apply(X = lebeau, MARGIN = 2, FUN = sd))
colnames(sDNlebeau)="sdN"
f<-function(x)(abs(100-(x*100/180)))
g<-function(x)(abs(100-(x*100/10)))
meanecartlebeau=as.data.frame(apply(X = meanlebeau, MARGIN = 2, FUN=f))
colnames(meanecartlebeau)="ecartpercent_mean"
sDNecartlebeau=as.data.frame(apply(X = sDNlebeau, MARGIN = 2, FUN=g))
colnames(sDNecartlebeau)="ecartpercent_mean"

tableau=data.frame(meanlebeau,sDNlebeau,meanecartlebeau,sDNecartlebeau)
rownames(tableau)=c("variety1_50","variety1_500","variety1_5000","variety1_50
000")

#We can see that more the data set is big and more mean and sDN value are
closer than the theorical mean and sDN

#6
#a.for variety1, the weight of a pear is less than 170gr
pnorm(170,180,10)

## [1] 0.1586553

#b. for variety1, the weight of a pear is between 160 and 200gr.
pnorm(200,180,10)-pnorm(160,180,10)
```

```
## [1] 0.9544997

#With the law of two sigma 180-2*10=160 and 180+2*10=200 we already know that
there is 95% to get a pear between these two value
#c.
pnorm(200,210,15)-pnorm(160,210,15)

## [1] 0.2520635

#d
1-pnorm(230,210,15)

## [1] 0.09121122

#7.
variety2_50000=as.data.frame(rnorm(50000,210,15))
colnames(variety2_50000)="weight"

#8.
variety=data.frame(ID=c(1:100000),variety50000,variety2_50000)
colnames(variety)=c("ID","variety1","variety2")
variety_reshape=melt(variety,id.vars='ID',measure.vars=c('variety1','variety2
'))
ggplot(variety_reshape,aes(x=value,y=..density..))+
geom_histogram(fill='blue',color='black')+
  geom_density(color='red')+facet_wrap(~variable)

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```r
#9
ggplot(variety_reshape,aes(x=value))+
  stat_ecdf()+facet_wrap(~variable)
```