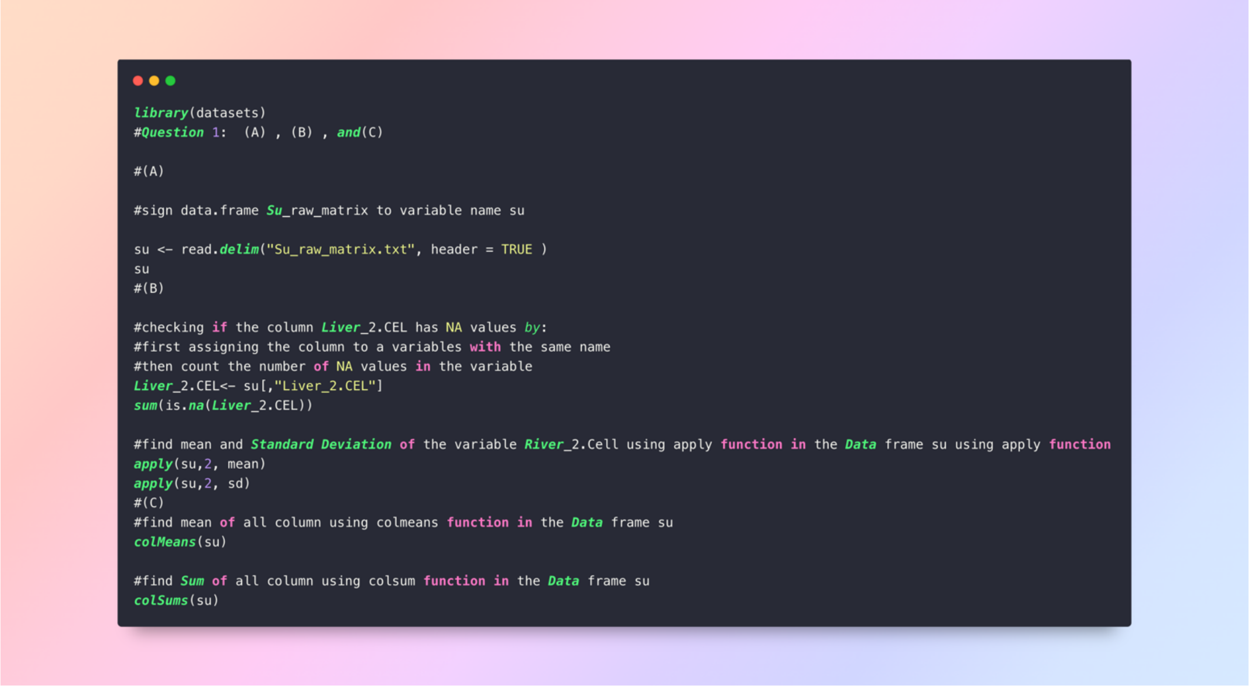
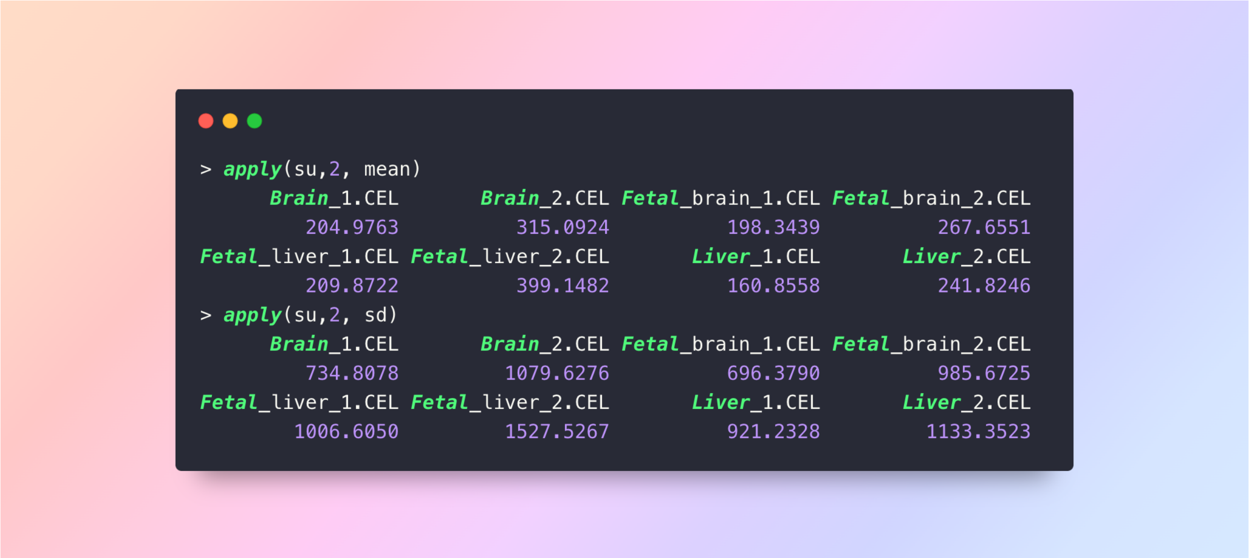
Question1:

explanation above each line of code assigned with# and comments also included



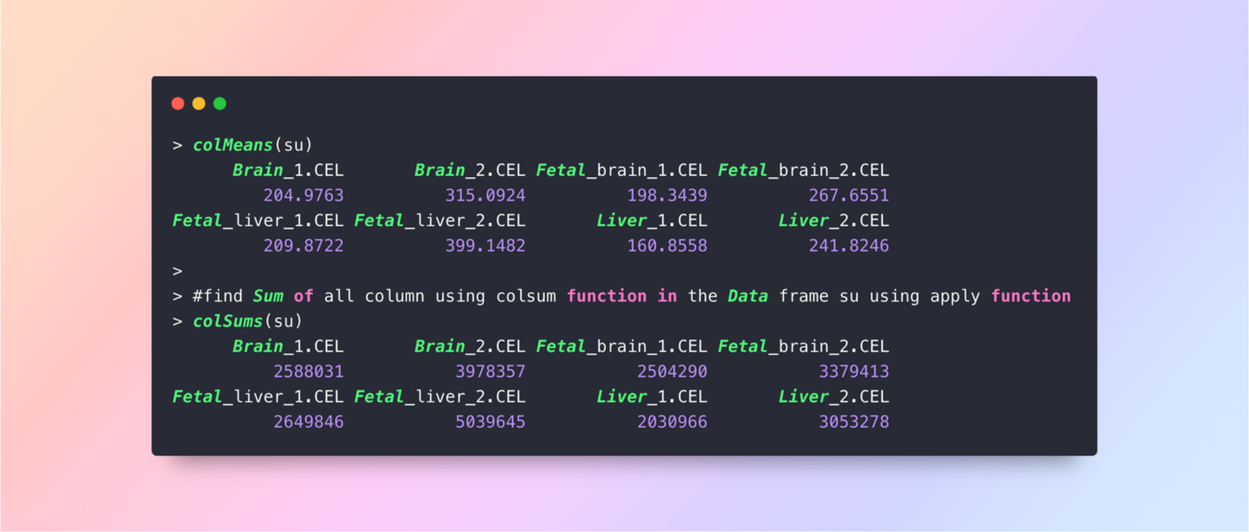
```
library(datasets)
#Question 1:  (A) , (B) , and(C)

#(A)

#sign data.frame Su_raw_matrix to variable name su

su <- read.delim("Su_raw_matrix.txt", header = TRUE )
su
#(B)

#checking if the column Liver_2.CEL has NA values by:
#first assigning the column to a variables with the same name
#then count the number of NA values in the variable
Liver_2.CEL<- su[,"Liver_2.CEL"]
sum(is.na(Liver_2.CEL))

#find mean and Standard Deviation of the variable River_2.Cell using apply function in the Data frame su using apply function
apply(su,2, mean)
apply(su,2, sd)
#(C)
#find mean of all column using colmeans function in the Data frame su
colMeans(su)

#find Sum of all column using colsum function in the Data frame su
colSums(su)
```

For Q1 B: Answer with



```
> apply(su,2, mean)
      Brain_1.CEL       Brain_2.CEL Fetal_brain_1.CEL Fetal_brain_2.CEL 
         204.9763          315.0924          198.3439          267.6551 
Fetal_liver_1.CEL Fetal_liver_2.CEL       Liver_1.CEL       Liver_2.CEL 
         209.8722          399.1482          160.8558          241.8246 
> apply(su,2, sd)
      Brain_1.CEL       Brain_2.CEL Fetal_brain_1.CEL Fetal_brain_2.CEL 
         734.8078         1079.6276          696.3790          985.6725 
Fetal_liver_1.CEL Fetal_liver_2.CEL       Liver_1.CEL       Liver_2.CEL 
        1006.6050         1527.5267          921.2328         1133.3523 
```

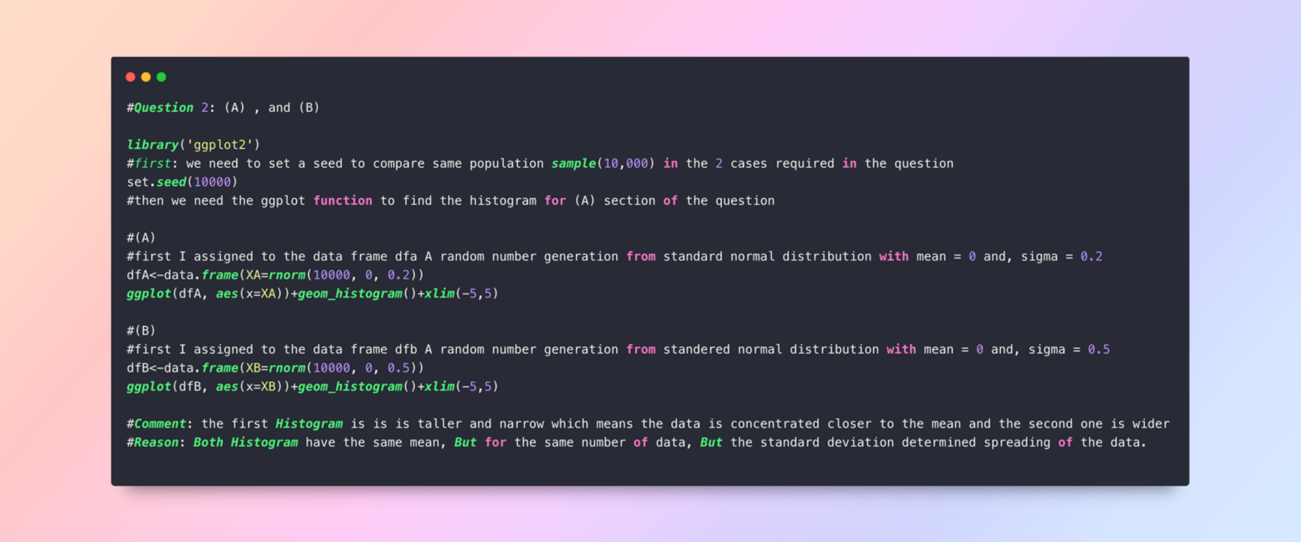
For Q1 C: Answer with explanation above each result



```
> colMeans(su)
      Brain_1.CEL       Brain_2.CEL Fetal_brain_1.CEL Fetal_brain_2.CEL
         204.9763          315.0924          198.3439          267.6551
Fetal_liver_1.CEL Fetal_liver_2.CEL       Liver_1.CEL       Liver_2.CEL
         209.8722          399.1482          160.8558          241.8246
>
> #find Sum of all column using colsum function in the Data frame su using apply function
> colSums(su)
      Brain_1.CEL       Brain_2.CEL Fetal_brain_1.CEL Fetal_brain_2.CEL
          2588031           3978357           2504290           3379413
Fetal_liver_1.CEL Fetal_liver_2.CEL       Liver_1.CEL       Liver_2.CEL
          2649846           5039645           2030966           3053278
```

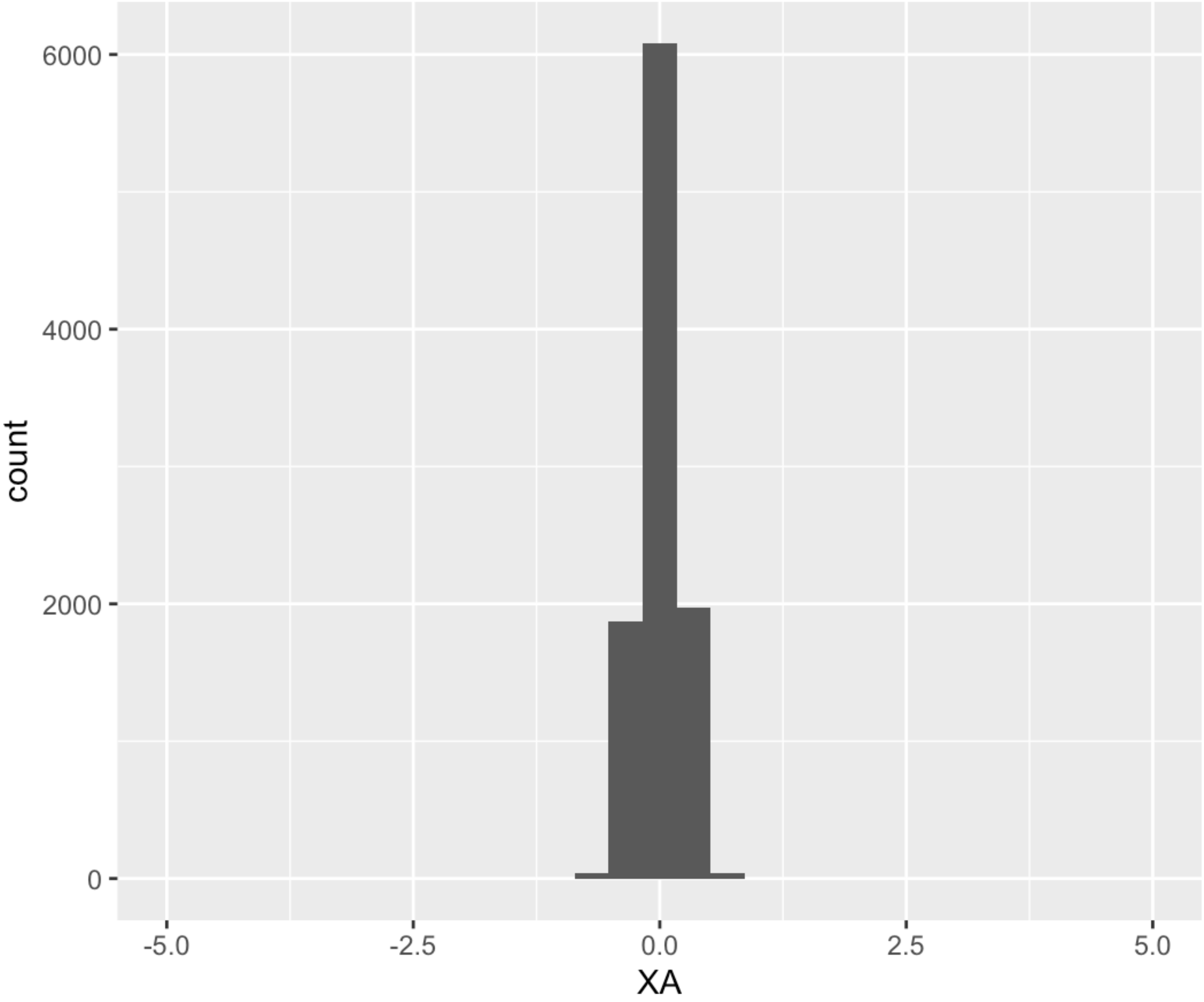
Question2:

explanation above each line of code assigned with# + and comments and reasons also included in the last 2 lines.

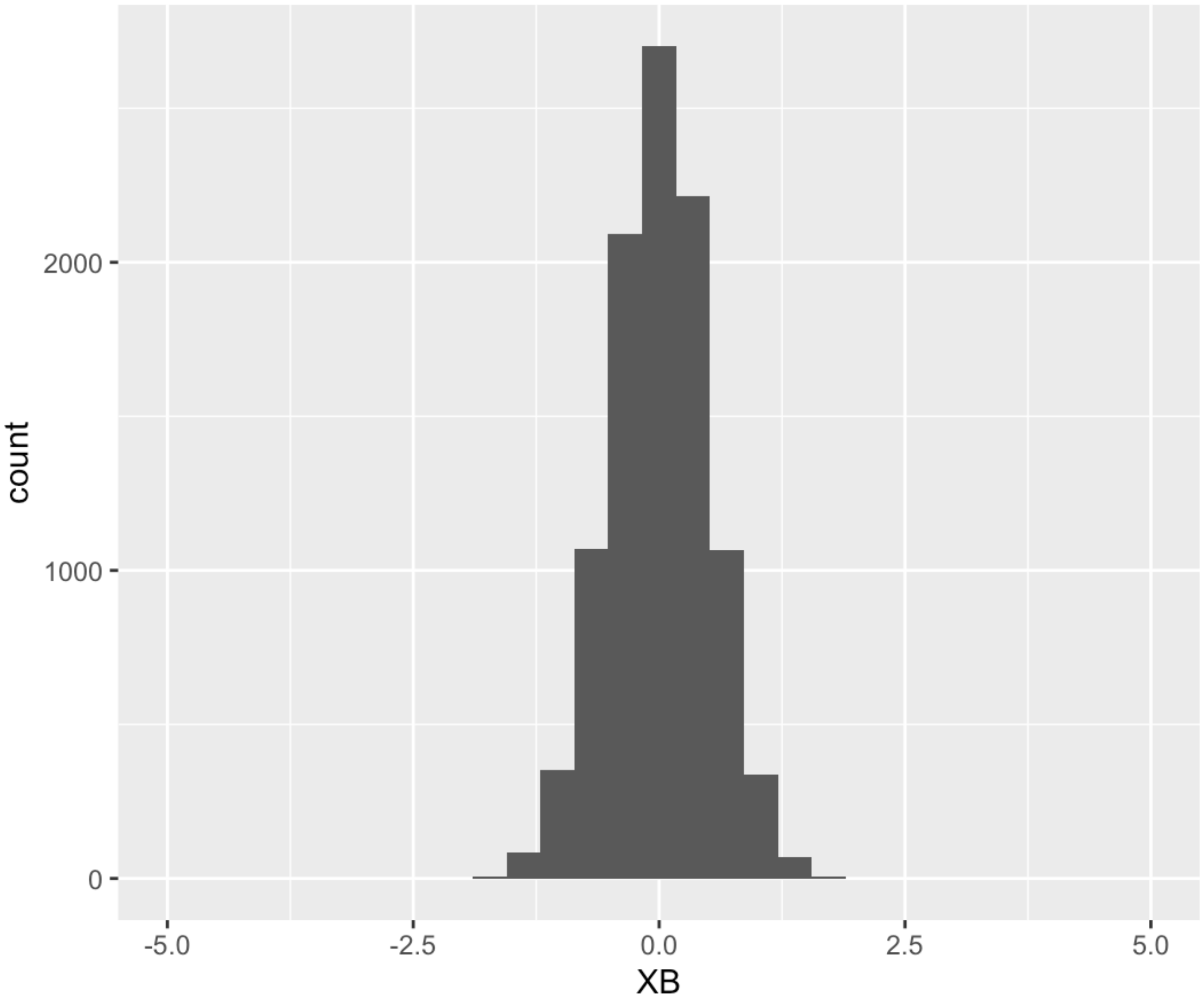


```r
#Question 2: (A) , and (B)

library('ggplot2')
#first: we need to set a seed to compare same population sample(10,000) in the 2 cases required in the question
set.seed(10000)
#then we need the ggplot function to find the histogram for (A) section of the question

#(A)
#first I assigned to the data frame dfa A random number generation from standard normal distribution with mean = 0 and, sigma = 0.2
dfA<-data.frame(XA=rnorm(10000, 0, 0.2))
ggplot(dfA, aes(x=XA))+geom_histogram()+xlim(-5,5)

#(B)
#first I assigned to the data frame dfb A random number generation from standered normal distribution with mean = 0 and, sigma = 0.5
dfB<-data.frame(XB=rnorm(10000, 0, 0.5))
ggplot(dfB, aes(x=XB))+geom_histogram()+xlim(-5,5)

#Comment: the first Histogram is is is taller and narrow which means the data is concentrated closer to the mean and the second one is wider
#Reason: Both Histogram have the same mean, But for the same number of data, But the standard deviation determined spreading of the data.
```

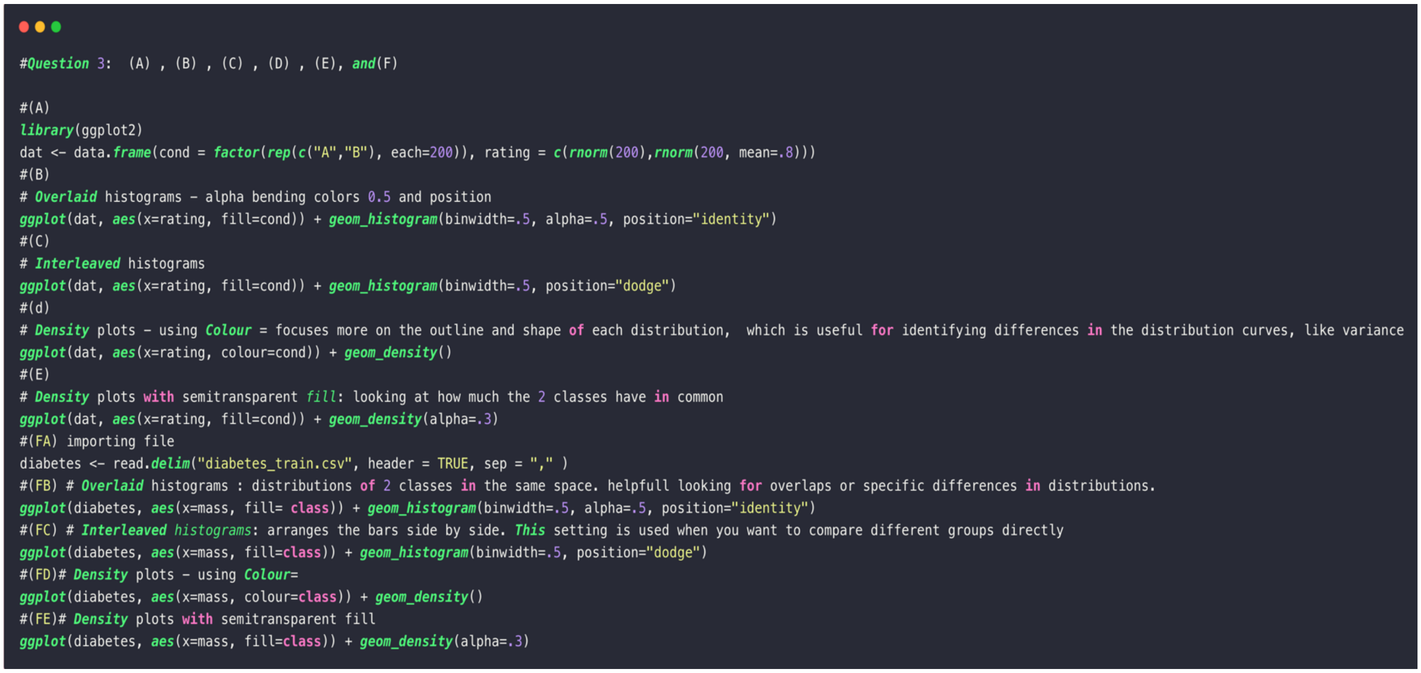
Histogram(A)



Histogram(B)

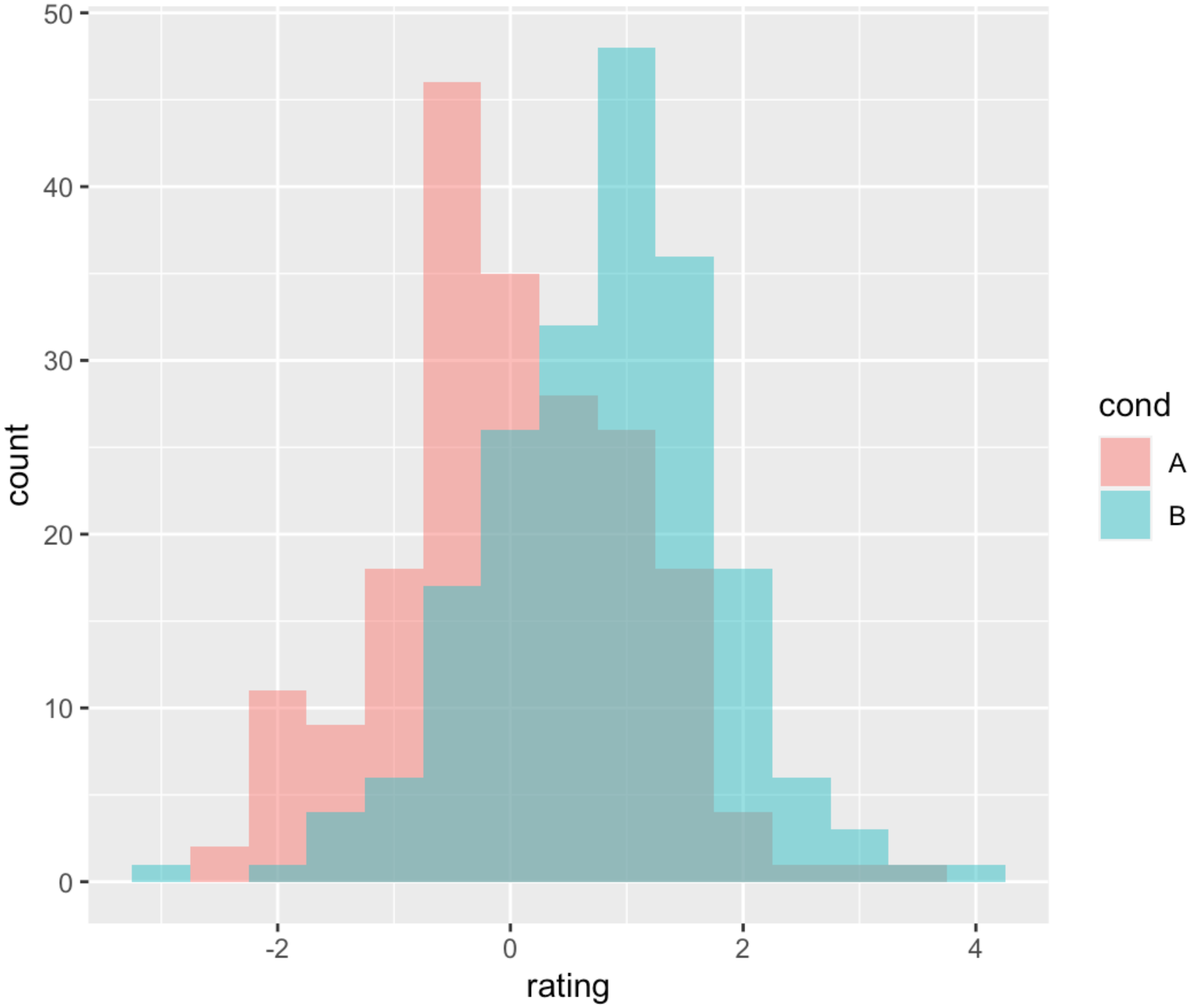


Question 3: explanation above each line of code assigned with# + and comments and reasons also included

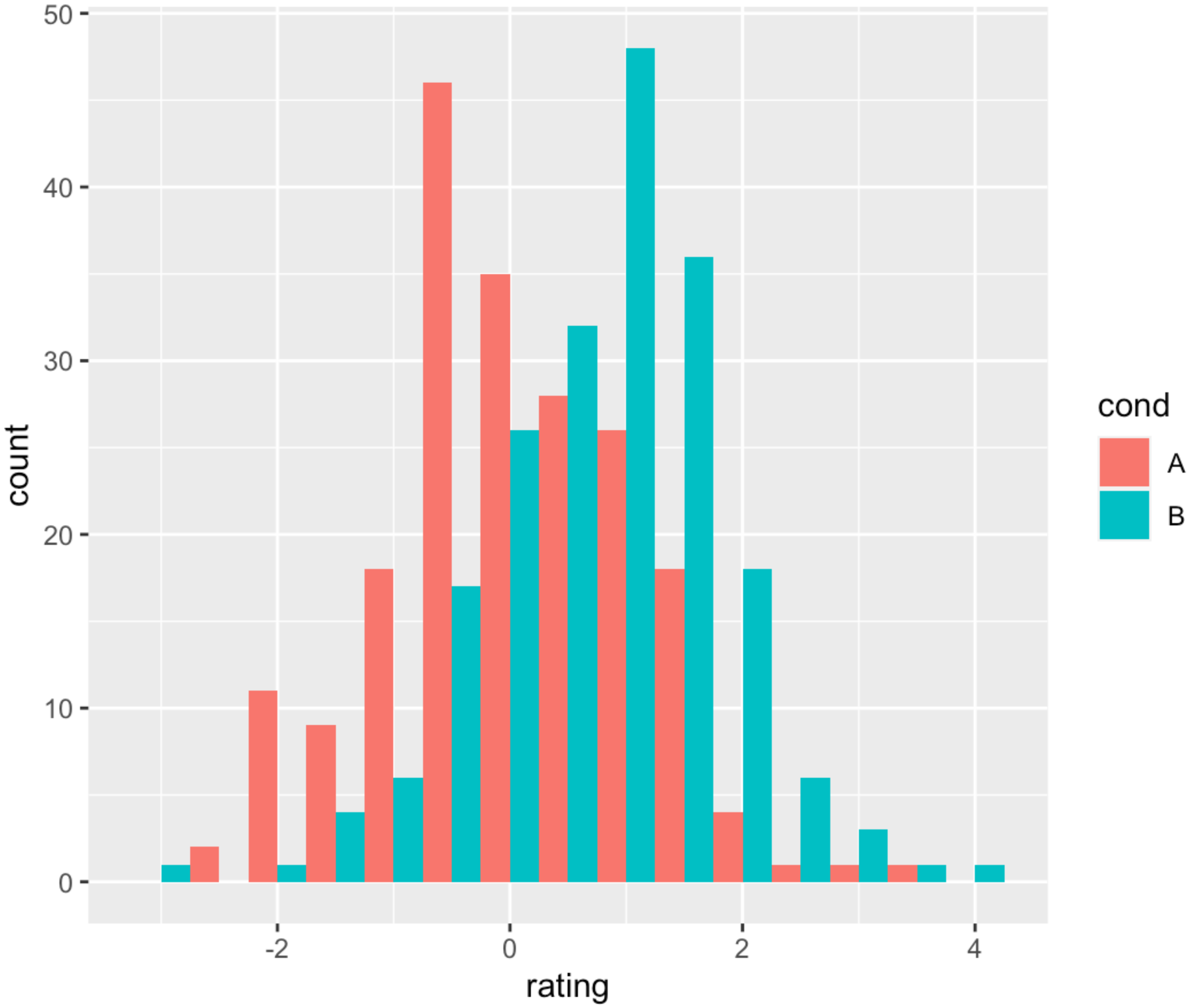


```
#Question 3:  (A) , (B) , (C) , (D) , (E), and(F)

#(A)
library(ggplot2)
dat <- data.frame(cond = factor(rep(c("A","B"), each=200)), rating = c(rnorm(200),rnorm(200, mean=.8)))
#(B)
# Overlaid histograms - alpha bending colors 0.5 and position
ggplot(dat, aes(x=rating, fill=cond)) + geom_histogram(binwidth=.5, alpha=.5, position="identity")
#(C)
# Interleaved histograms
ggplot(dat, aes(x=rating, fill=cond)) + geom_histogram(binwidth=.5, position="dodge")
#(d)
# Density plots - using Colour = focuses more on the outline and shape of each distribution,  which is useful for identifying differences in the distribution curves, like variance
ggplot(dat, aes(x=rating, colour=cond)) + geom_density()
#(E)
# Density plots with semitransparent fill: looking at how much the 2 classes have in common
ggplot(dat, aes(x=rating, fill=cond)) + geom_density(alpha=.3)
#(FA) importing file
diabetes <- read.delim("diabetes_train.csv", header = TRUE, sep = "," )
#(FB) # Overlaid histograms : distributions of 2 classes in the same space. helpfull looking for overlaps or specific differences in distributions.
ggplot(diabetes, aes(x=mass, fill= class)) + geom_histogram(binwidth=.5, alpha=.5, position="identity")
#(FC) # Interleaved histograms: arranges the bars side by side. This setting is used when you want to compare different groups directly
ggplot(diabetes, aes(x=mass, fill=class)) + geom_histogram(binwidth=.5, position="dodge")
#(FD)# Density plots - using Colour=
ggplot(diabetes, aes(x=mass, colour=class)) + geom_density()
#(FE)# Density plots with semitransparent fill
ggplot(diabetes, aes(x=mass, fill=class)) + geom_density(alpha=.3)
```

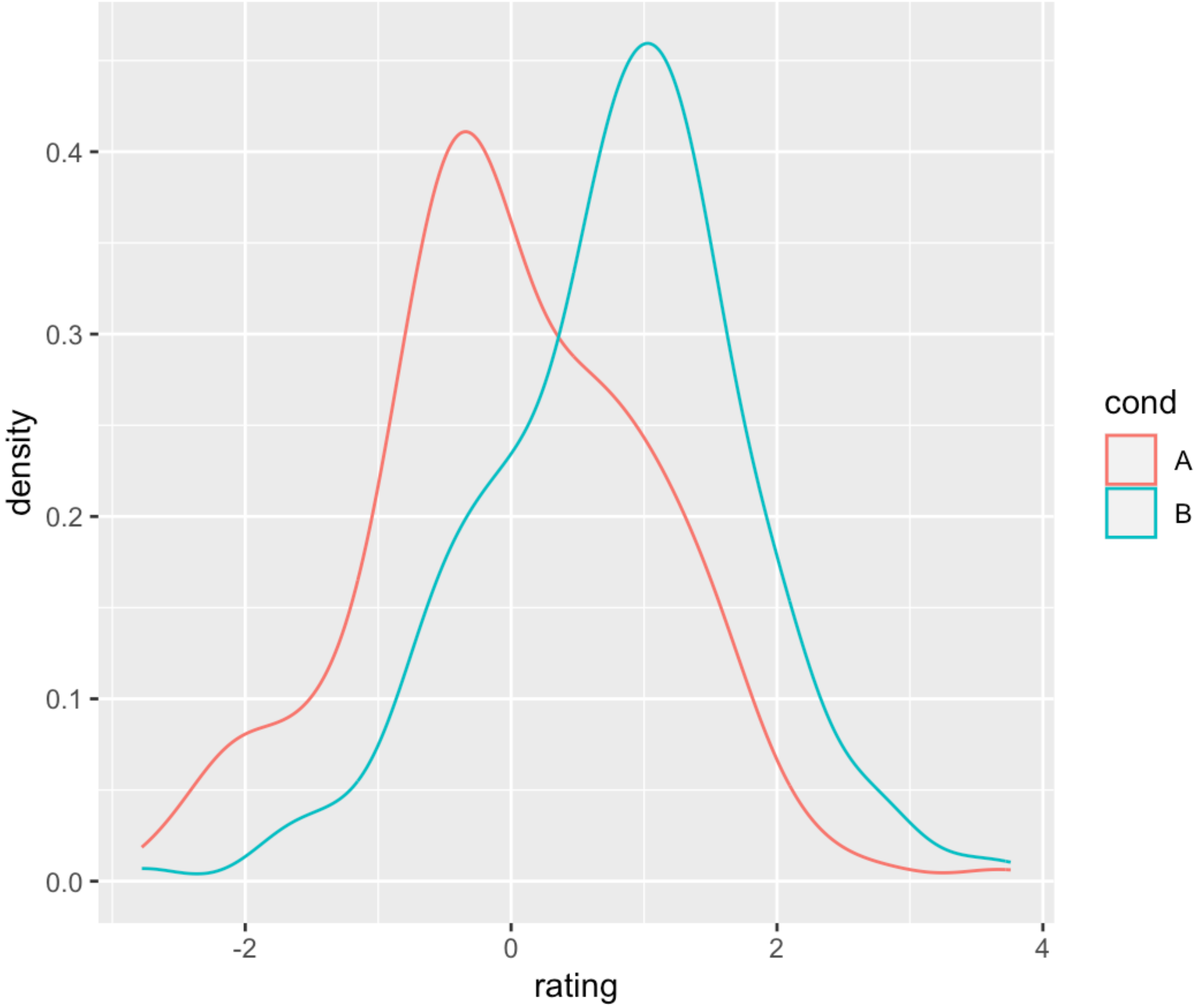
Histogram(B)



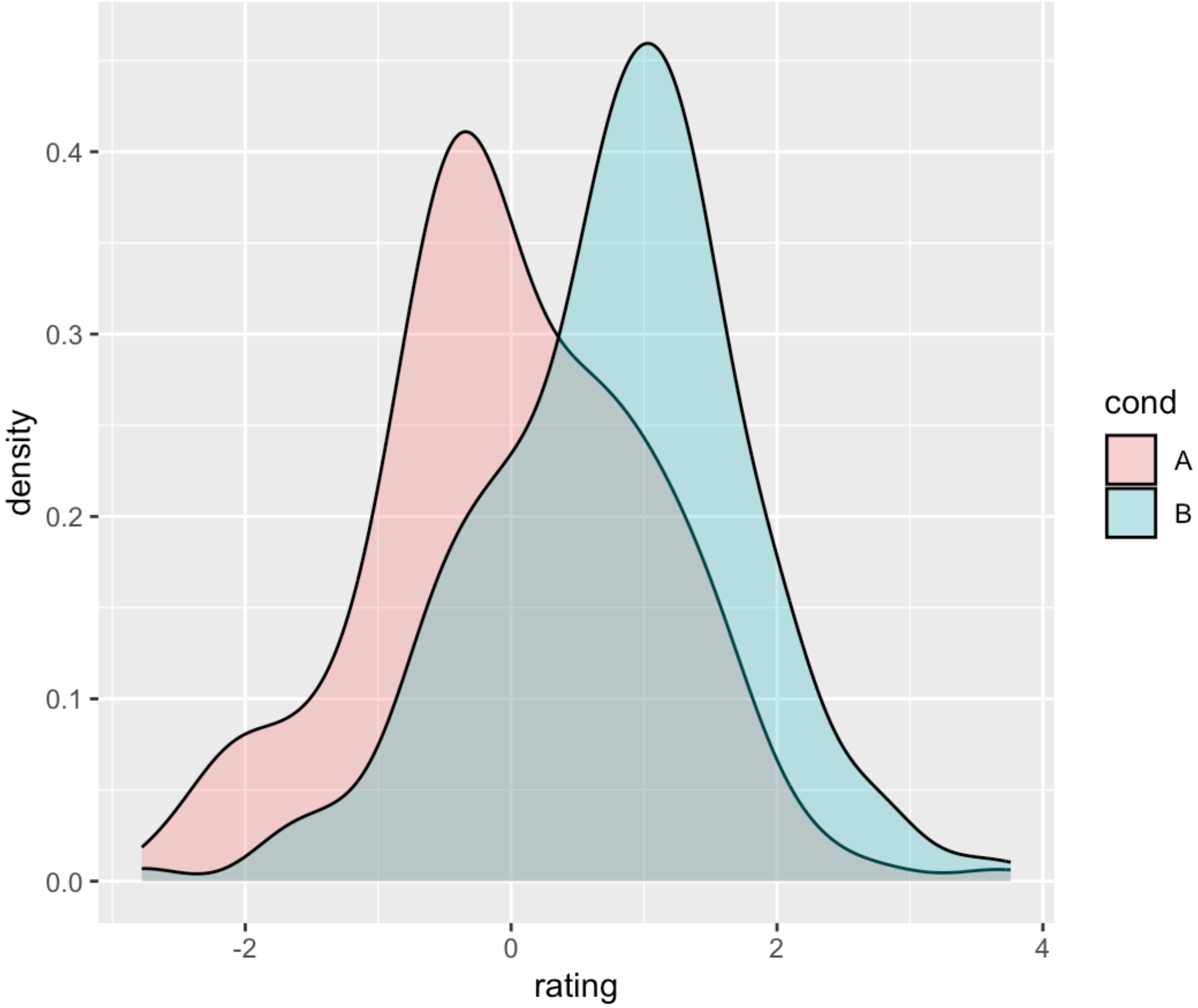
Histogram(C)



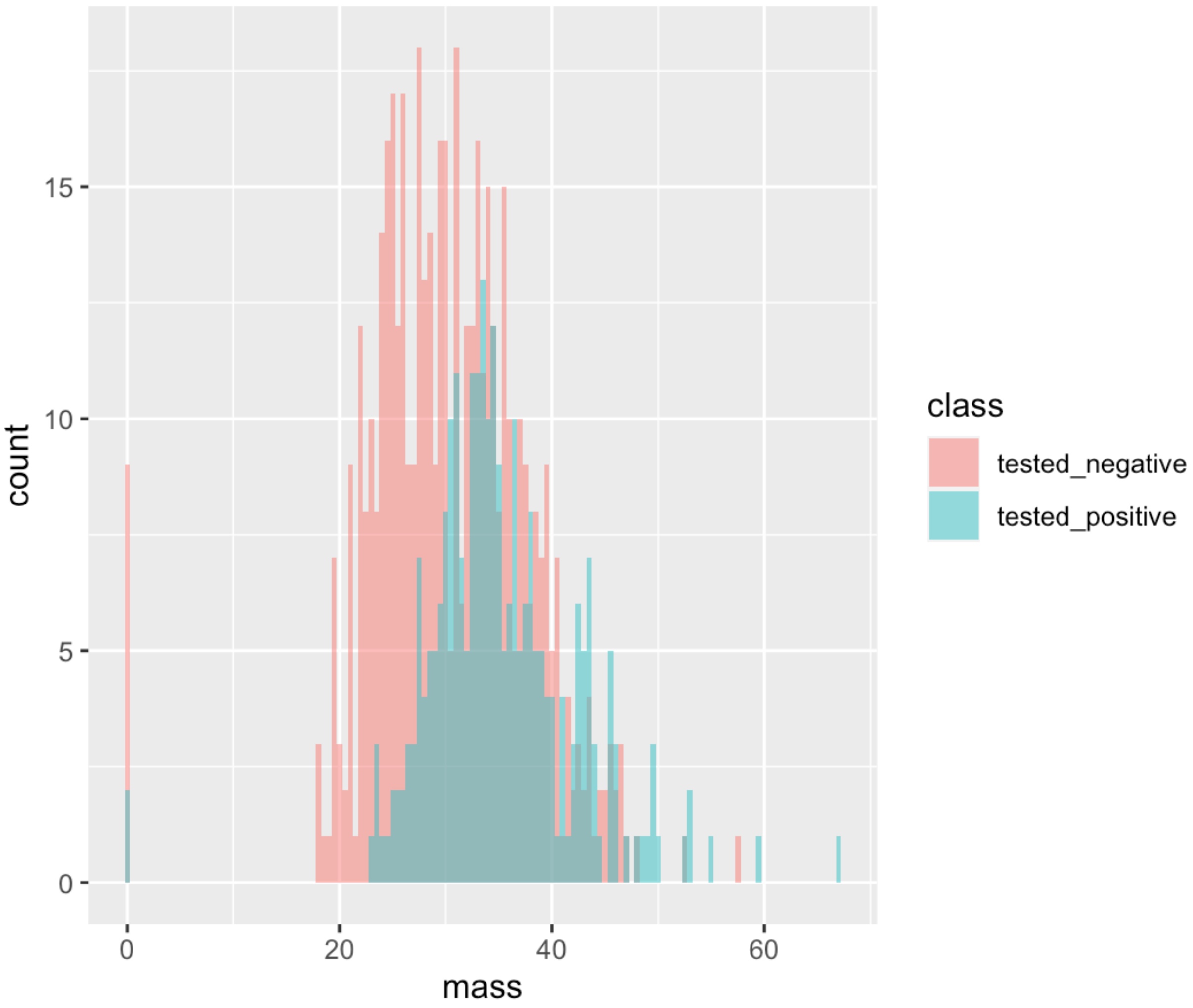
Histogram(D)



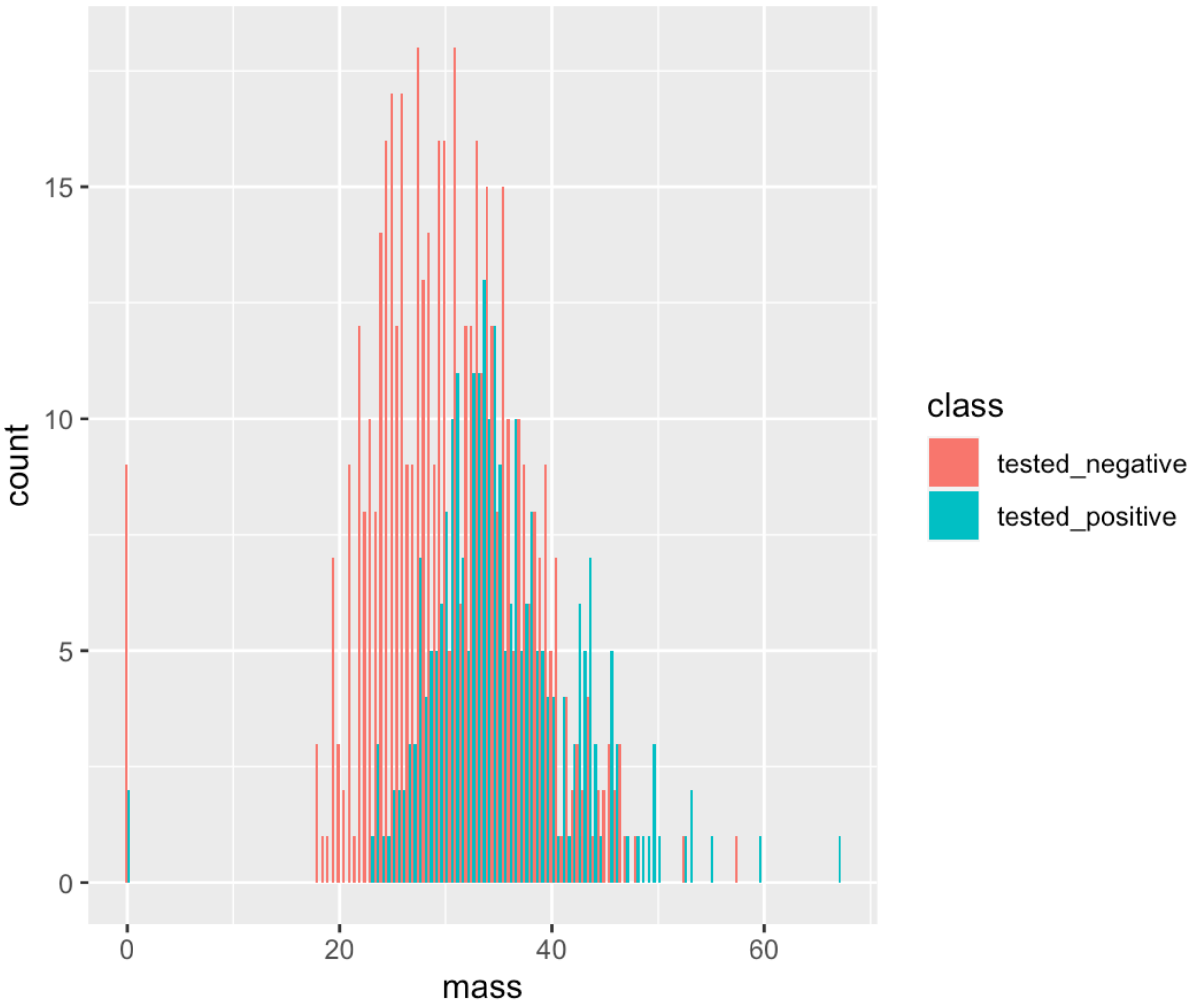
Histogram(E)



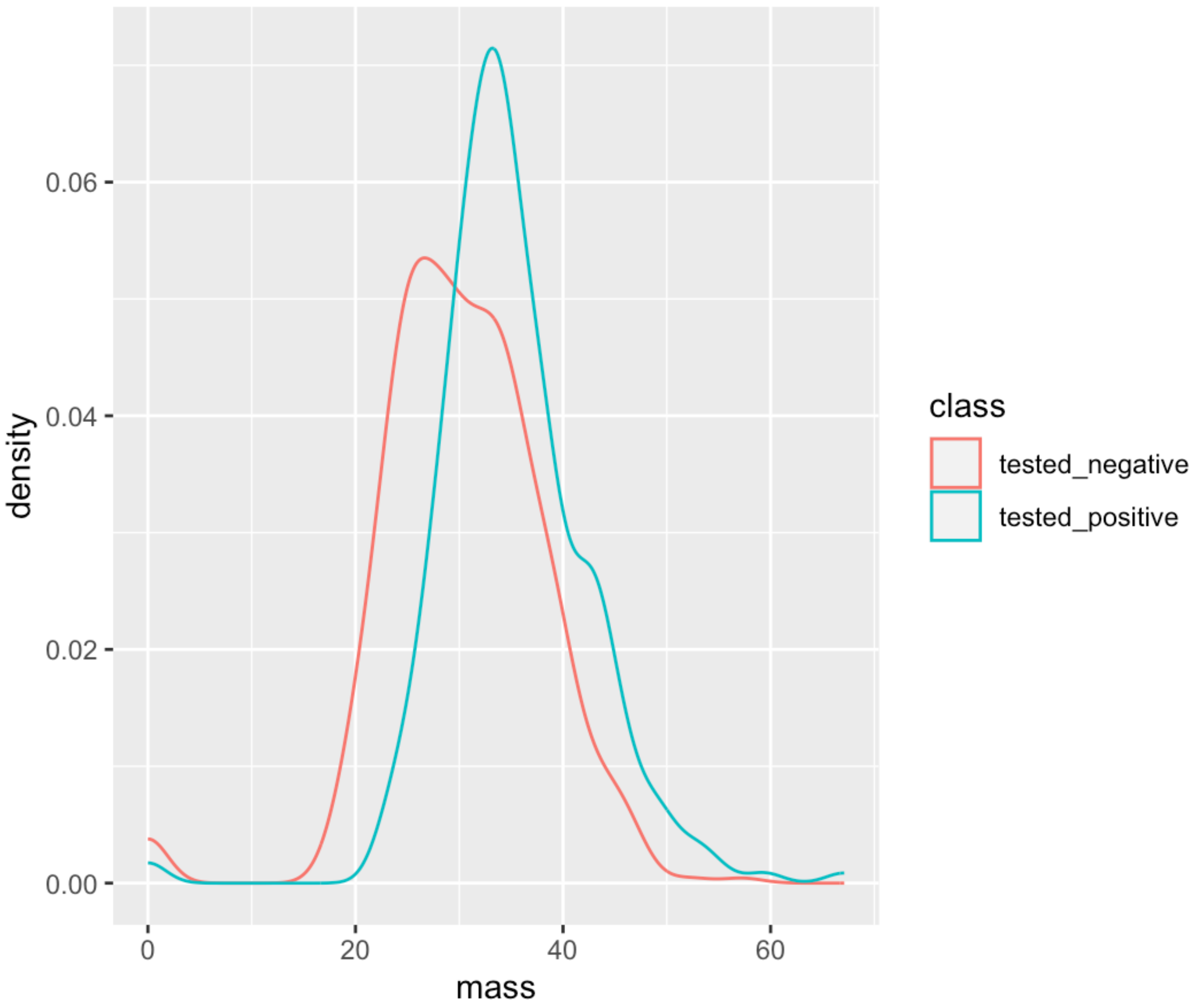
Histogram(FB)



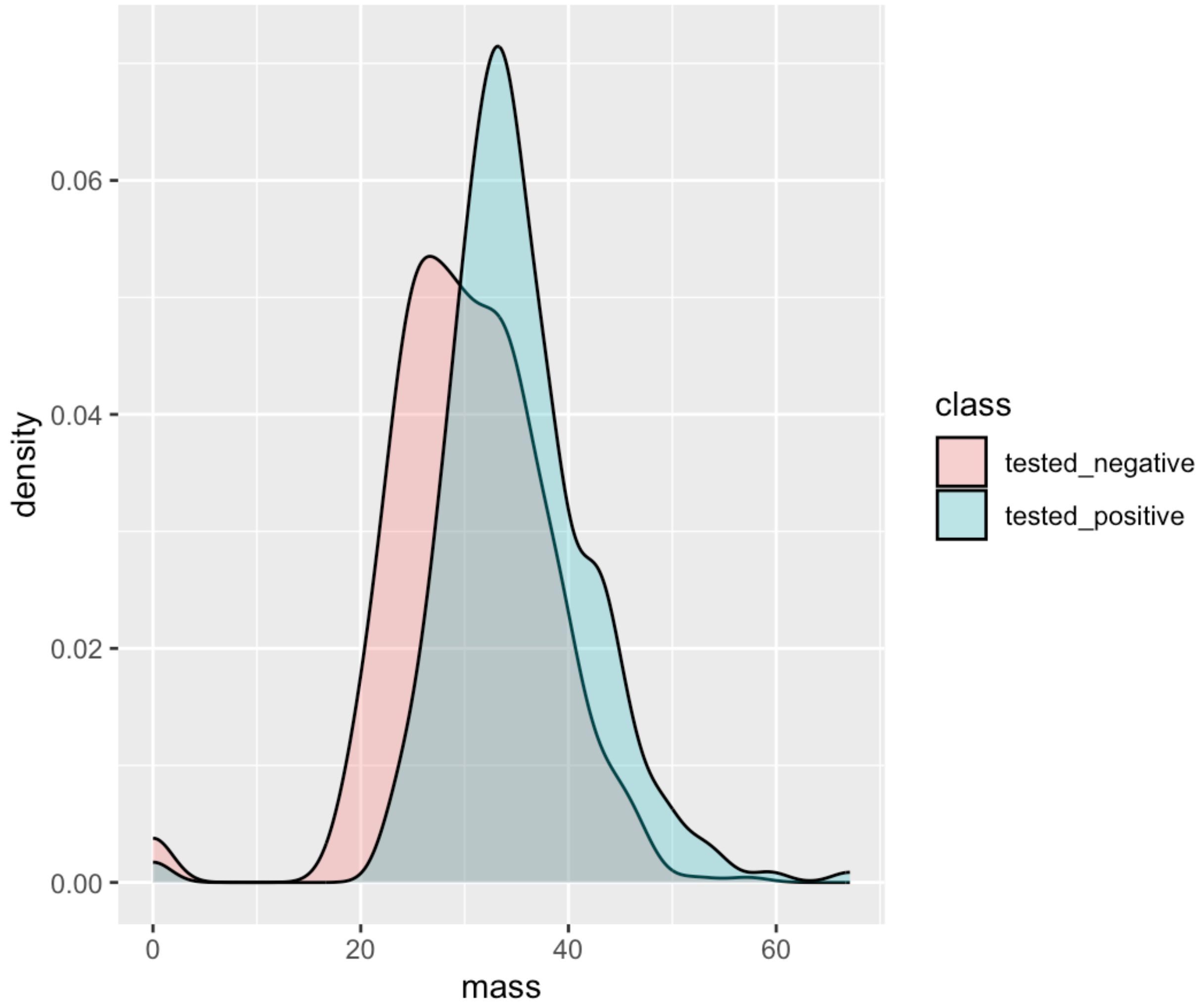
Histogram(FC)



Histogram(FD)



Histogram(FE)



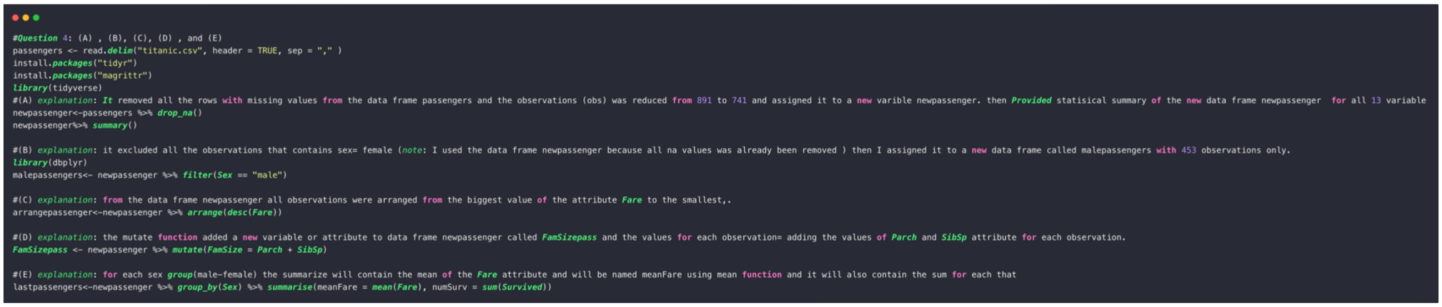
Question 4:
#(A) explanation: It removed all the rows with missing values from the data frame passengers and the observations (obs) was reduced from 891 to 741 and assigned it to a new variable newpassenger. then Provided statistical summary of the new data frame newpassenger for all 13 variable

#(B) explanation: it excluded all the observations that contains sex= female (note: I used the data frame newpassenger because all NA values was already been removed) then I assigned it to a new data frame called malepassengers with 453 observations only.

#(C) explanation: from the data frame newpassenger all observations were arranged from the biggest value of the attribute Fare to the smallest.

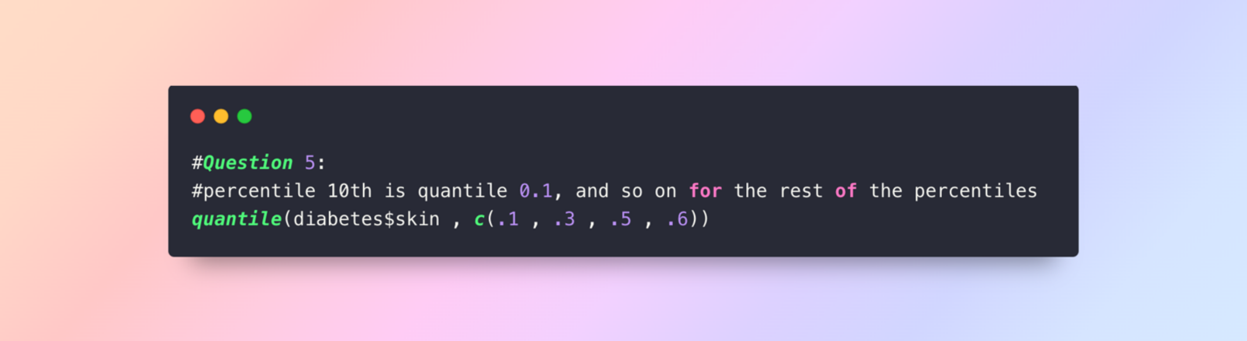
#(D) explanation: the mutate function added a new variable or attribute to data frame newpassenger called FamSizepass and the values for each observation= adding the values of Parch and SibSp attribute for each observation.

#(E) explanation: for each sex group(male-female) the summarize will contain the mean of the Fare attribute and will be named meanFare using mean function and it will also contain the sum for each.



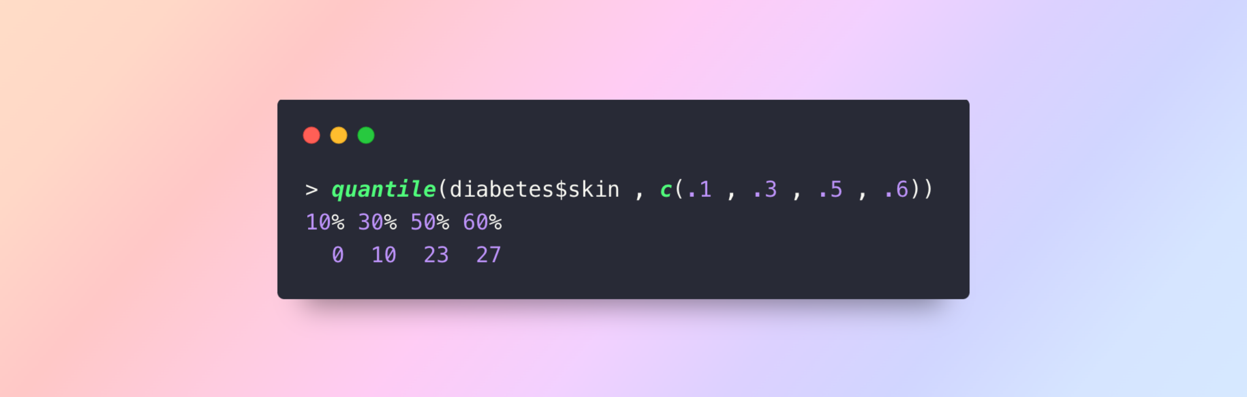
```
#Question 4: (A) , (B), (C), (D) , and (E)
passengers <- read.delim("titanic.csv", header = TRUE, sep = "," )
install.packages("tidyr")
install.packages("magrittr")
library(tidyverse)
#(A) explanation: It removed all the rows with missing values from the data frame passengers and the observations (obs) was reduced from 891 to 741 and assigned it to a new varible newpassenger. then Provided statisical summary of the new data frame newpassenger  for all 13 variable
newpassenger<-passengers %>% drop_na()
newpassenger%>% summary()

#(B) explanation: it excluded all the observations that contains sex= female (note: I used the data frame newpassenger because all na values was already been removed ) then I assigned it to a new data frame called malepassengers with 453 observations only.
library(dbplyr)
malepassengers<- newpassenger %>% filter(Sex == "male")

#(C) explanation: from the data frame newpassenger all observations were arranged from the biggest value of the attribute Fare to the smallest,.
arrangepassenger<-newpassenger %>% arrange(desc(Fare))

#(D) explanation: the mutate function added a new variable or attribute to data frame newpassenger called FamSizepass and the values for each observation= adding the values of Parch and SibSp attribute for each observation.
FamSizepass <- newpassenger %>% mutate(FamSize = Parch + SibSp)

#(E) explanation: for each sex group(male-female) the summarize will contain the mean of the Fare attribute and will be named meanFare using mean function and it will also contain the sum for each that
lastpassengers<-newpassenger %>% group_by(Sex) %>% summarise(meanFare = mean(Fare), numSurv = sum(Survived))
```

Question 5 :



```
#Question 5:
#percentile 10th is quantile 0.1, and so on for the rest of the percentiles
quantile(diabetes$skin , c(.1 , .3 , .5 , .6))
```

Answer:



```
> quantile(diabetes$skin , c(.1 , .3 , .5 , .6))
10% 30% 50% 60%
  0  10  23  27
```