PROJECT of Data Science

PROJECT REPORT
97

The Students of this team and their id:

Student name	id	group
Taha fawzy anwar elshrif (Project manager)	20221466557	g3
Mohamed Abdallah Mohamed abdelrehim	20201444880	g2
Mohamed mahmoud ali elgazzar	20221450635	g3
Ahmed Ibrahim hassan Mohamed hessien	20221373950	g3

Generally description about the project

Data:
Data about transactions from some people providing their age , place , name , total of spending in that transaction ,number of things bought ,way of payment

This project:
!-plot and deduce things from plots(taha elshrif , Mohamed Abdallah)
2-Dashboard plots(Mohamed el gazzar)
3-apply the algorithm of kmeans(Mohamed el gazzar)
4-apply the AprIori algorithm (Ahmed ibrahim)

Input : path of file , number of clusters to kmeans , number for minimum support and confidence.

Output: plots, rules about data , ways to group data

The role of each student in the project

1-User input path and data visualization (The role of ”Taha elshrif” &”Mohamed Abdallah abdelrehim”):

```
path<-readline("Enter path: ")

library(dplyr)

#raw data
dta<-read.csv(path,stringsAsFactors=FALSE)

#number of credit/cash transactions
NmbrTyp<-table(dta$paymentType)
NmbrTyp
```

First of all the user should enter the path where the data hold (in type .csv) then press ENTER
The data then warehoused in a variable (dta) and has the attribute (function read.csv) stringsAsFactors=false to provide it from being only strings
Note : read.csv is in the library (“dplyr”)

The role of each student in the project

1-User input path and data visualization (The role of ”Taha elshrif” &”Mohamed Abdallah abdelrehim”):

Visualizations:

i. Compare cash and credit totals (pie plot)

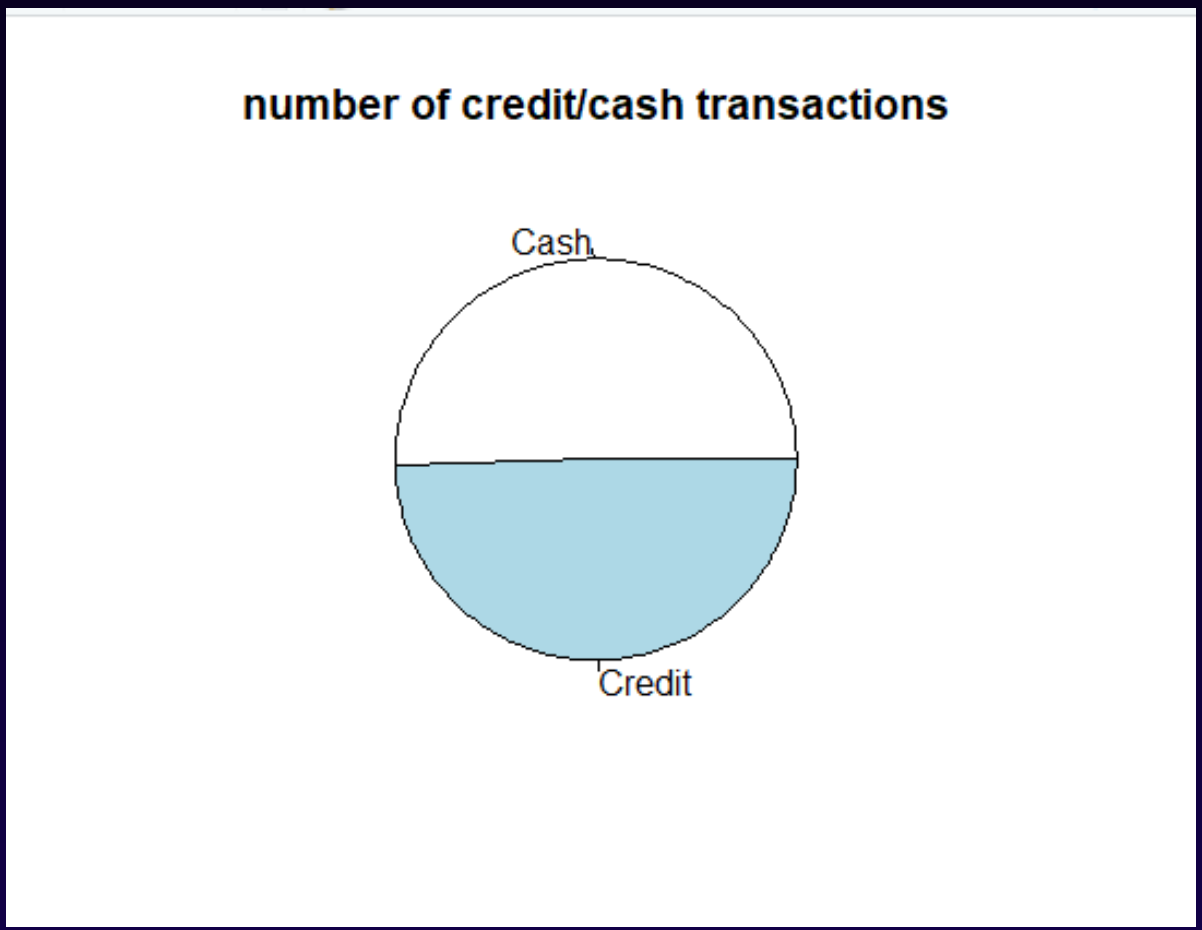
```
#number of credit/cash transactions
NmbrTyp<-table(dta$paymentType)
NmbrTyp

#pie visulaization
pie(
  NmbrTyp
  , main="number of credit/cash transactions"
)
#from pie visualization we can deduce that number of credit transactions =cash transaction.
```

The details of the code (at first):
We have made a variable NmbrTyp contains a table of cash /credit
And number of its transactions by function table()
And then we used the function of pie to visualize it as a pie and gave it the last table and main “number of credit/cash transactions”

The role of each student in the project

1-User input path and data visualization (The role of ”Taha elshrif” &”Mohamed Abdallah abdelrehim”):

Visualizations:



from pie visualization we can deduce that approximately number of credit transactions =cash transaction.

The role of each student in the project

1-User input path and data visualization (The role of ”Taha elshrif” &”Mohamed Abdallah abdelrehim”):

Visualizations:

ii. Compare each age and sum of total spending. (scatter plot)

```
#ages :group data ,column of age ,by ages.
ages<-group_by(dta,age)


comb<-summarise(ages,smtra=sum(total))#smtra is the sum of transactions

#comb makes a table of age and the sum of transactions(has been done by all with this age)


plot(
  y=comb$smtra ,
  x=comb$age,
  main = "Comparing ages with total spending",
  xlab = "total of spending",
  ylab = "age",
  las=1
)
```

The details of the code (at first):

We have used 2 other functions in “dplyr” library

----------→`ages<-group_by(dta,age)`

To group data by age (organize it by ages)

`comb<-summarise(ages,smtra=sum(total))#smtra is the sum of transactions`

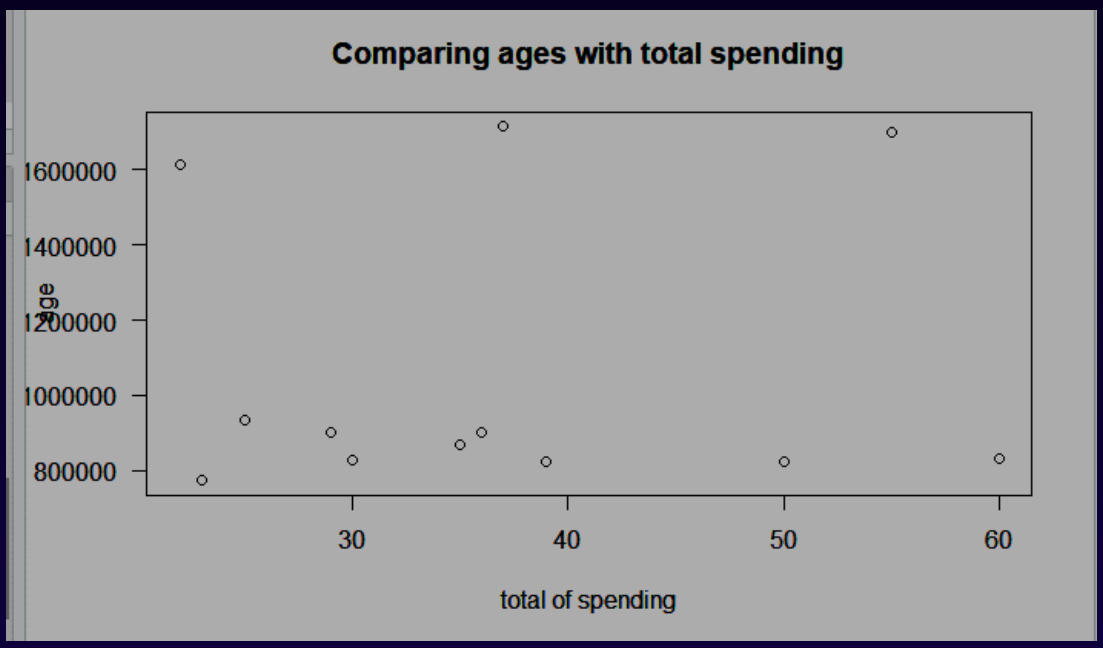
To make a table that shows each age and the sum of its transactions

```
plot(
  y=comb$smtra ,
  x=comb$age,
  main = "Comparing ages with total spending",
  xlab = "total of spending",
  ylab = "age",
  las=1
)
```

To scatter plot “comb”

The role of each student in the project

And has a main to give a name for its plot ,and Xlab and Ylab to give a name for what have been visualized in x and y
Las=1 to provide the values which put in y from being vertical (to make it easy to read){values not lab}



#from the visualizations between ages and total of spending we can deduce that the most number of transactions are from the customers whose age 22 ,55, and espacially 37
As some values are far in the plot

The role of each student in the project

iii. Show each city total spending and arrange it by total descending (horz. plot)

```
#cities :group data ,column of city ,by cities.
cities<-group_by(dta,city)

#creating a table of cities and total spending of each city
x<-summarise(cities,Sum=sum(total))#total_Sum is the total spending of each city

#arranging the table descendingly according to the total spending of each city
x<-arrange(x,desc(Sum),city)

barplot(
  height = x$Sum,
  name=x$city,
  main = "Comparing cities with total spending",
  xlab = "total of spending",
  las=1,
  horiz=TRUE
)
#from the bar plot we deduce that Alexandria,Cairo,Hurghada have the largest amount of total spendings
```

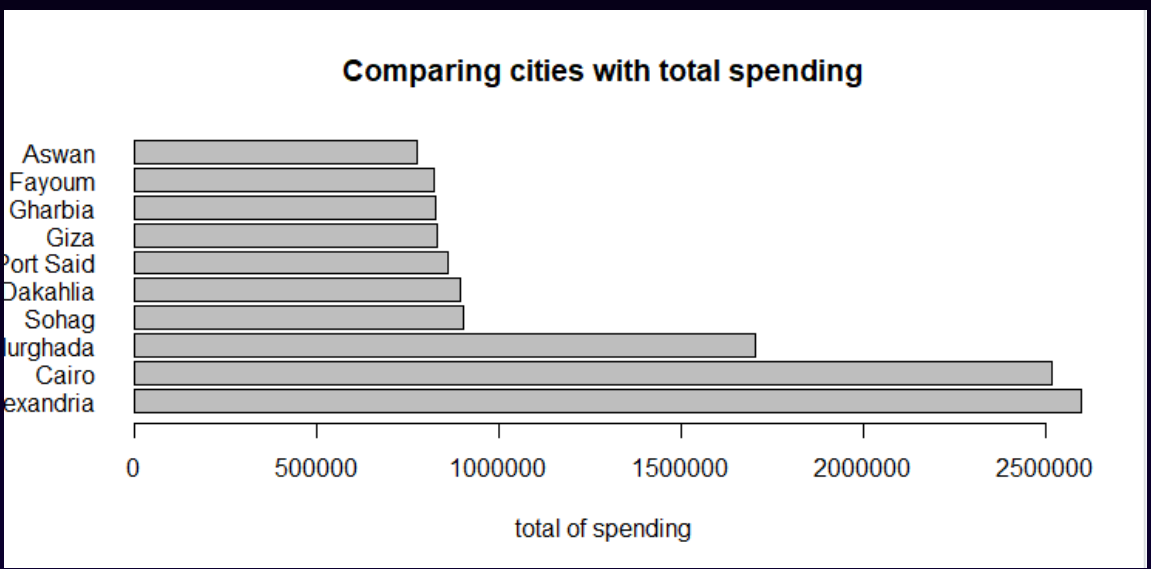
cities<-group_by(dta,city)
To group the data according tp cities

x<-summarise(cities,Sum=sum(total))#Sum is the total spending of each city
To add the column of total spending of each city individually

x<-arrange(x,desc(Sum),city)
To arrange the total spending of each city descendungly

barplot(
  height = x$Sum,
  name=x$city,
  main = "Comparing cities with total spending",
  xlab = "total of spending",
  las=1,
  horiz=TRUE
)
To plot the data using a horizontal barplot ,has the same attributes Of the last plot and (horiz=TRUE) to make it horizational

The role of each student in the project



From the barplot we can deduce that the cities Alexandria, Cairo, Hurghada are the cities with the highest spending value

The role of each student in the project

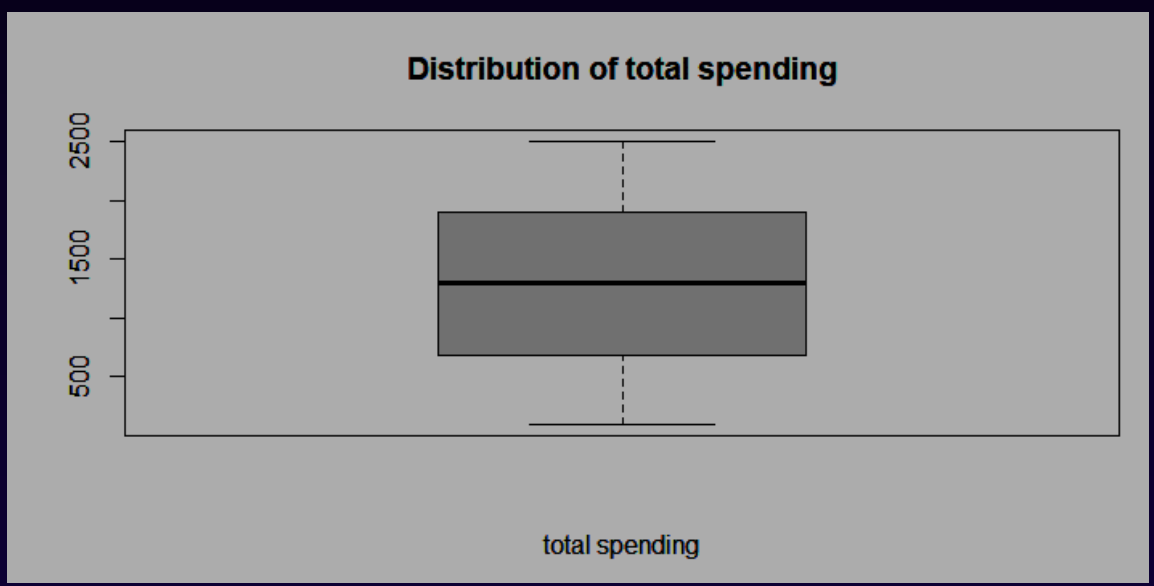
iv. Display the distribution of total spending.(BOXPLOT)

```
#Distribution of spendings
boxplot(
  x=dta$total,
  main="Distribution of total spending",
  xlab="total spending"
)
```

boxplot(
  x=dta$total,
  main="Distribution of total spending",
  xlab="total spending"
)

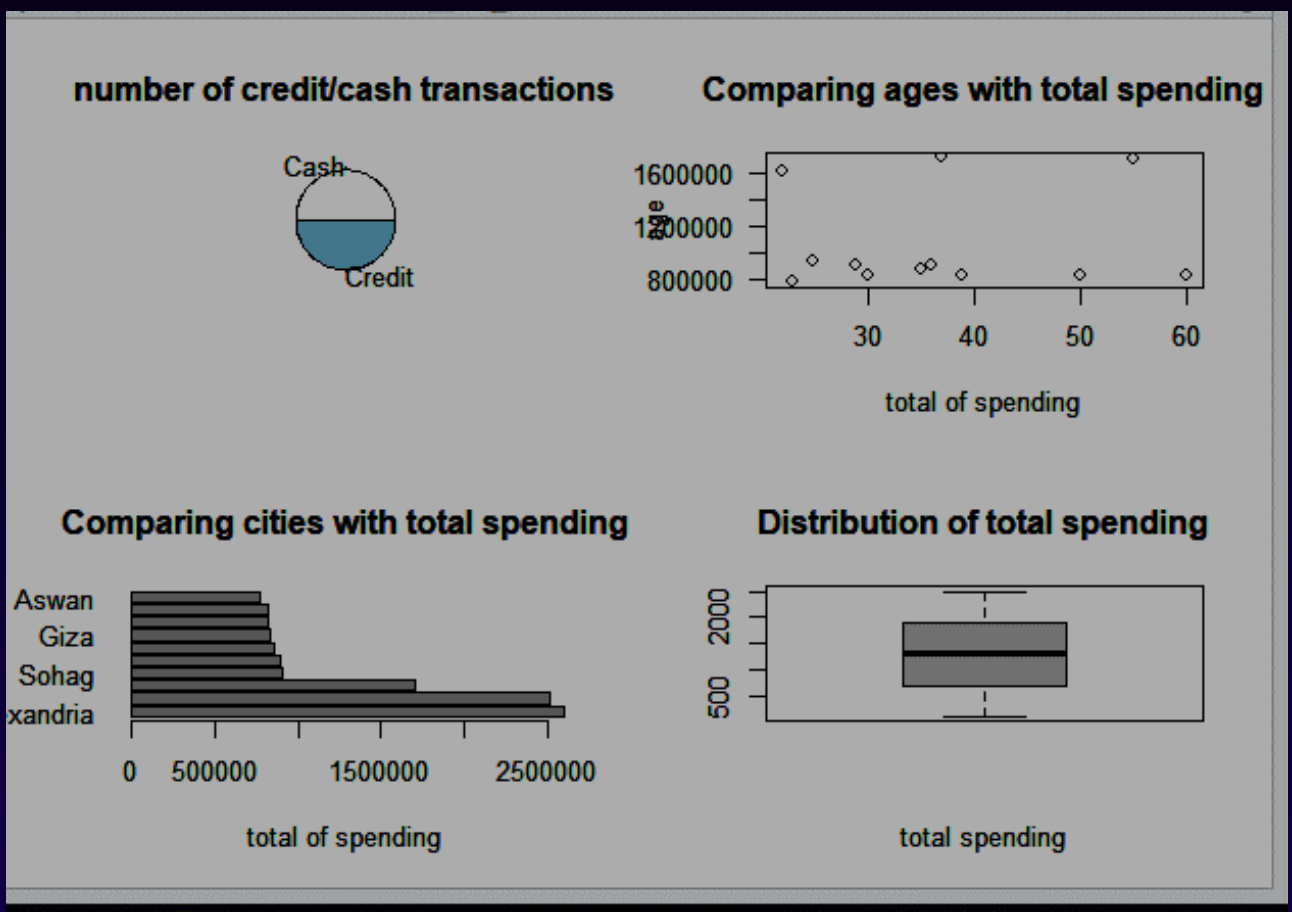
To plot a boxplot in order to show the distribution of spendings
Has the attributes:
Main to give a title of it , xlab to give a title for things which visualized in x

The role of each student in the project



From the box plot we can deduce that the minimum spending us approximately 100, the maximum is 2500 and the average is about 1400

Butting all plots into one dashboard



Code:

```
par(mfrow=c(2,2))
```

---→written at first(after first lines)

The role of each student in the project

2- Applying Algorithems on the data (The role of ”Mohamed elgazzar” &”ahmed ibrahim”):

1-KMEANS algorithm :

```
#from the box plot we deduce that the distribution of spendings has maximum=250
#kmeans
nmbrcnts<-readline("Enter number of groups [1:4] :")
#to classify data
clstr <- kmeans(comb, centers = nmbrcnts)
clstr
# to show data in new groups
grops<-cbind(clstr$cluster,comb$age,comb$smtra)
#giving names to columns
colnames(grops)<-c("Cluster","Age","total")
grops
```

At first :the user should input the number of clusters which data will be classified then press enter.

Then the code:
We used the function kmeans
-the data we classified is”comb” which we have used before and consisting of each age and number of total spending (as said in the question)
Then we used (cbind) to make a table of ages,clusters ,total spending (to view data in new clusters) and gave a name to each column by (colname())
Variables:
Nmbrcnts: number of centers (input)
Clstr:variable used in kmeans algorithm
Grops :variable to show new clusters

The role of each student in the project

The following figure is when we classified data in 3 groups

R 4.1.1 · C:/Users/tahae/OneDrive/Desktop/The project/

```
> #kmeans
> nmbrcnts<-readline("Enter number of groups [1:4] :")
Enter number of groups [1:4] :3
> #to classify data
> clstr <- kmeans(comb, centers = nmbrcnts)
> clstr
K-means clustering with 3 clusters of sizes 4, 5, 3

Cluster means:
    age     smtra
1 31.25  900931.2
2 40.40  816588.2
3 38.00 1675852.7

Clustering vector:
 [1] 3 2 1 1 2 1 1 3 2 2 3 2

Within cluster sum of squares by cluster:
[1] 1958279208 2424917892 5897622371
 (between_SS / total_SS =  99.3 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss" "betweenss"    "size"
[8] "iter"         "ifault"
> # to show data in new groups
> grops<-cbind(clstr$cluster,comb$age,comb$smtra)
> #giving names to columns
> colnames(grops)<-c("Cluster","Age","total")
> grops
      Cluster Age   total
 [1,]       3  22 1613801
 [2,]       2  23  772871
 [3,]       1  25  932250
 [4,]       1  29  900797
 [5,]       2  30  829587
 [6,]       1  35  869668
 [7,]       1  36  901010
 [8,]       3  37 1714689
 [9,]       2  39  825147
[10,]       2  50  824064
[11,]       3  55 1699068
[12,]       2  60  831272
>
```

The role of each student in the project

2- Applying Algorithems on the data (The role of ”Mohamed elgazzar” &”ahmed ibrahim”):

2- APRORI algorithm :

```
mnSup<-readline("Enter min support [.01:1]:")
mnConf<-readline("Enter min Confidence [.01:1]:")

library("arules")
library("gtools")

tdata <- read.transactions(path, sep=",")
inspect(tdata)
apriori_ruls <- apriori(tdata, parameter = list(supp = mnSup, conf = mnConf ,minlen=2))
inspect(apriori_ruls)
```

At first :the user should input the minimum support and confidence (first two lines)
User should input them between ,01 to 1
Then we used read.transactions and put the attribute “,” to it to choose what was separated by ,
Then we used the function apriori and gave it min support(minsp) and min confidence(minConf)
Then inspect to view the result
Note those functions found in package arules

The role of each student in the project

Result by minsup:.01 and minConf =.01

```
 is(value, "numeric") is not TRUE
> inspect(apriori_rules)
     lhs          rhs      support   confidence coverage  lift      count
[1]  {Cairo}   => {Cash}   0.1010573 0.5100051  0.1981497 1.0119852  994
[2]  {Cash}    => {Cairo}  0.1010573 0.2005245  0.5039650 1.0119852  994
[3]  {2}       => {Credit} 0.1101057 0.4884980  0.2253965 0.9850074 1083
[4]  {Credit}  => {2}      0.1101057 0.2220172  0.4959333 0.9850074 1083
[5]  {2}       => {Cash}   0.1152908 0.5115020  0.2253965 1.0149554 1134
[6]  {Cash}    => {2}      0.1152908 0.2287674  0.5039650 1.0149554 1134
[7]  {1}       => {Credit} 0.1321675 0.4907512  0.2693168 0.9895509 1300
[8]  {Credit}  => {1}      0.1321675 0.2665027  0.4959333 0.9895509 1300
[9]  {1}       => {Cash}   0.1371492 0.5092488  0.2693168 1.0104844 1349
[10] {Cash}    => {1}      0.1371492 0.2721404  0.5039650 1.0104844 1349
> result <- kmeans(comb, centers = 2)
```