**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Part 1**

**Code part:a**

forbes <- read.csv("https://people.bu.edu/kalathur/datasets/forbes.csv")

#a)

#using table function to calculaate the frequency by country

freq\_country <- table(forbes$country)

max\_val <- max(freq\_country)

par(mar=c(7,4,2,2))

barplot(freq\_country,col = "lightpink",las =2,ylim=c(0,max\_val))

Console section:

forbes <- read.csv("https://people.bu.edu/kalathur/datasets/forbes.csv")

> #a)

> #using table function to calculaate the frequency by country.

> freq\_country <- table(forbes$country)

> m <- max(freq\_country)

> par(mar=c(7,4,2,2))

> barplot(freq\_country,col = "lightpink",las =2, ylim=c(0,700))

Plot:a

A picture containing line, diagram

Description automatically generated

-------------------------------------------------------------------------------------------------------------

**Code section:b**

freq\_gender <- table(forbes$gender)

slices <- c(freq\_gender[1]/sum(freq\_gender),freq\_gender[2]/sum(freq\_gender))

lbls <- c("Female","Male")

pct <- round(slices\*100,2)

lbls <- paste(lbls,pct,"%")

lbsl <- paste(lbls,"%", sep = "")

pie(slices,labels = lbls,col = rainbow(2))

**console section:b**

> freq\_gender <- table(forbes$gender)

> slices <- c(freq\_gender[1]/sum(freq\_gender),freq\_gender[2]/sum(freq\_gender))

> lbls <- c("Female","Male")

> pct <- round(slices\*100,2)

> lbls <- paste(lbls,pct,"%")

> lbsl <- paste(lbls,"%", sep = "")

> pie(slices,labels = lbls,col = rainbow(2))

**Plot:a**

A blue circle with a red triangle

Description automatically generated with medium confidence

-------------------------------------------------------------------------------------------------------------------------------

**Code section:c**

#c)

top\_categories <-sort(table(forbes$category), decreasing = TRUE)[1:5]

# Subset of the dataset for the top 5 categories

subset\_forbes <- forbes[forbes$category %in% names(top\_categories), ]

top\_five <- table(subset\_forbes$category,subset\_forbes$gender)

bar\_names <- rownames(top\_five)

par(mar=c(3,3,2,0))

barplot(t(top\_five),beside = TRUE,col = c("lightpink","lightblue"),names.arg = bar\_names,ylim = c(0,350),legend.text = c("Female","Male"))

**Console section:c**

> top\_categories <-sort(table(forbes$category), decreasing = TRUE)[1:5]

> # Subset of the dataset for the top 5 categories

> subset\_forbes <- forbes[forbes$category %in% names(top\_categories), ]

> top\_five <- table(subset\_forbes$category,subset\_forbes$gender)

> bar\_names <- rownames(top\_five)

> par(mar=c(3,3,2,0))

>

> barplot(t(top\_five), beside = TRUE,col = c("lightpink","lightblue"),names.arg = bar\_names,ylim = c(0,350),legend.text = c("Female","Male"))

**Plot:c**

A picture containing text, screenshot, diagram, plot

Description automatically generated

-----------------------------------------------------------------------------------------------------------------------------

Section:D

d) following are inferences can be drawn from above plots

1.Threre are around 88% of male and around 12% of female among the billionaire datasets

2. Most billionaires are from United State followed by China.

3. most of the billionaires are from finance & Investments categories followed by Manufacturing.

4. Highest number of billionaires are from Finance and investments

5. highest numbers of male billionaires are from Finance and investments and Highest number of female billionaires are from Food and Beverage categories.

------------------------------------------------------------------------------------------------------------------------------

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Part 2

**--------------------------------------------------------------------------------------------------------------------------**

**Code section: a**

us\_quarters <- read.csv("https://people.bu.edu/kalathur/datasets/us\_quarters.csv")

attach(us\_quarters)

#a)

highest\_denvermint\_state <- subset(us\_quarters,DenverMint==max(DenverMint))

highest\_phyllymint\_state <- subset(us\_quarters,PhillyMint==max(PhillyMint))

lowest\_denvermint\_state <- subset(us\_quarters,DenverMint==min(DenverMint))

lowest\_phyllymint\_state <- subset(us\_quarters,PhillyMint==min(PhillyMint))

paste("The highest number of quaters produced for DenverMint is by ",highest\_denvermint\_state$State,"State which is",

highest\_denvermint\_state$DenverMint)

paste("The highest number of quaters produced for PhyllyMint is by ",highest\_phyllymint\_state$State,"State which is",

highest\_phyllymint\_state$PhillyMint)

paste("The lowest number of quaters produced by DenverMint is by ",lowest\_denvermint\_state$State,"State which is",

lowest\_denvermint\_state$DenverMint)

paste("The lowest number of quaters produced for PhyllyMint is by ",lowest\_phyllymint\_state$State,"State which is",

lowest\_phyllymint\_state$PhillyMint)

**Console section :a**

> us\_quarters <- read.csv("https://people.bu.edu/kalathur/datasets/us\_quarters.csv")

> attach(us\_quarters)

> #a)

> highest\_denvermint\_state <- subset(us\_quarters,DenverMint==max(DenverMint))

> highest\_phyllymint\_state <- subset(us\_quarters,PhillyMint==max(PhillyMint))

> lowest\_denvermint\_state <- subset(us\_quarters,DenverMint==min(DenverMint))

> lowest\_phyllymint\_state <- subset(us\_quarters,PhillyMint==min(PhillyMint))

> paste("The highest number of quaters produced for DenverMint is by ",highest\_denvermint\_state$State,"State which is",

+ highest\_denvermint\_state$DenverMint)

[1] "The highest number of quaters produced for DenverMint is by Connecticut State which is 657880"

> paste("The highest number of quaters produced for PhyllyMint is by ",highest\_phyllymint\_state$State,"State which is",

+ highest\_phyllymint\_state$PhillyMint)

[1] "The highest number of quaters produced for PhyllyMint is by Virginia State which is 943000"

> paste("The lowest number of quaters produced by DenverMint is by ",lowest\_denvermint\_state$State,"State which is",

+ lowest\_denvermint\_state$DenverMint)

[1] "The lowest number of quaters produced by DenverMint is by Oklahoma State which is 194600"

--------------------------------------------------------------------------------------------------------------------------

> paste("The lowest number of quaters produced for PhyllyMint is by ",lowest\_phyllymint\_state$State,"State which is",

+ lowest\_phyllymint\_state$PhillyMint)

[1] "The lowest number of quaters produced for PhyllyMint is by Iowa State which is 213800"

-----------------------------------------------------------------------------------------------------------------------

**Code section: b**

data\_matrix\_quaters <-rbind(DenverMint,PhillyMint)

data\_matrix\_quaters

state\_names <-State

bar\_colors <- c("blue", "grey")

bar\_legend <- c("Denvermint","Phillymint")

options(scipen = 5)

par(mar=c(8,5,3,0))

barplot(data\_matrix\_quaters,beside = TRUE, col = bar\_colors,legend.text = bar\_legend,

ylim = c(0,1000000), names.arg = state\_names,las = 2)

**Console section: b**

> data\_matrix\_quaters <-rbind(DenverMint,PhillyMint)

> data\_matrix\_quaters

[,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13] [,14] [,15] [,16] [,17] [,18] [,19] [,20]

DenverMint 401424 358332 299028 488744 657880 535184 556532 566208 495976 651616 619640 427876 447100 459404 370564 286468 414832 402204 327200 289600

PhillyMint 373400 349000 363200 451188 688744 628600 678200 742576 673040 943000 655400 627600 423000 423400 353000 361600 217200 362000 362600 290000

[,21] [,22] [,23] [,24] [,25] [,26] [,27] [,28] [,29] [,30] [,31] [,32] [,33] [,34] [,35] [,36] [,37] [,38] [,39] [,40]

DenverMint 237400 232400 231400 228200 229800 225800 241600 263000 251400 226800 263200 248400 404000 300000 356200 312800 276400 294200 359000 265800

PhillyMint 225800 225000 217400 225000 228000 233800 240200 278800 213800 226400 257200 239600 316200 263400 365400 277000 318000 274800 305800 245000

[,41] [,42] [,43] [,44] [,45] [,46] [,47] [,48] [,49] [,50]

DenverMint 256240 280000 286800 320800 253200 194600 244400 265000 254000 263600

PhillyMint 257000 265200 294600 243600 255000 222000 244200 244600 251800 254000

> state\_names <-State

> bar\_colors <- c("blue", "grey")

> bar\_legend <- c("Denvermint","Phillymint")

> options(scipen = 5)

> par(mar=c(8,5,3,0))

> barplot(data\_matrix\_quaters,beside = TRUE, col = bar\_colors,legend.text = bar\_legend,

+ ylim = c(0,1000000), names.arg = state\_names,las = 2)

Plot :a

A picture containing screenshot, plot, diagram, line

Description automatically generated

From above chart,

1. The PhillyMint produces high number of Quarters in Virginia state

2. The DenverMint produces higher number of Quarters in Connecticut State

------------------------------------------------------------------------------------------------------------------------

Code part:c

boxplot(DenverMint,PhillyMint,col = c("blue","lightpink"),pch=16,names = c("Denvermint","Phillymint"))

Console part: c

boxplot(DenverMint,PhillyMint,col = c("blue","lightpink"),pch=16,names = c("Denvermint","Phillymint"))

Plot : c

A picture containing rectangle, screenshot, diagram, text

Description automatically generated

DenverMint:

1 There are two outliers at the upper end of the data

2 The upper whisker is longer than the lower whisker, suggesting that there is greater variability or spread in the upper part of the data distribution.

Phillymint:

1 There are 8 outliers at the upper end of the data.

#2 lower whisker is short which means there is less variability/spread in data in the lower part of the data.

--------------------------------------------------------------------------------------------------------------

**Code part :d**

**#d**

For DenverMint

#finding five number in Denvermint column

fivnum\_denv <- fivenum(DenverMint)

#finding outliers towards upper end

denv\_out\_high <- fivnum\_denv[4]+1.5\*(fivnum\_denv[4]-fivnum\_denv[2])

denver\_outlier\_high <- subset(us\_quarters,DenverMint > denv\_out\_high)

denver\_outlier\_high$State

#finding outliers towars lower end

denv\_out\_low <- fivnum\_denv[2]-1.5\*(fivnum\_denv[4]-fivnum\_denv[2])

denver\_outlier\_low <- subset(us\_quarters,DenverMint < denv\_out\_low)

denver\_outlier\_low

**so, there is no outliers in lower ends**

**For Phillymint**

fivnum\_philly <- fivenum(PhillyMint)

#finding outliers towards upper end

philly\_out <- fivnum\_philly[4]+1.5\* (fivnum\_philly[4]-fivnum\_philly[2])

philly\_outlier <- subset(us\_quarters,PhillyMint>philly\_out)

#outlier states are

philly\_outlier$State

#finding outliers towards lower end.

philly\_out\_low <- fivnum\_philly[2]-1.5\*(fivnum\_philly[4]-fivnum\_philly[2])

philly\_outlier\_low <- subset(us\_quarters,PhillyMint< philly\_out\_low)

philly\_outlier\_low

**so, there is no outliers in lower ends**

**Console section :d**

#For DenverMint

> #finding five number in Denvermint column

> fivnum\_denv <- fivenum(DenverMint)

> #finding outliers towards upper end

> denv\_out\_high <- fivnum\_denv[4]+1.5\*(fivnum\_denv[4]-fivnum\_denv[2])

> denver\_outlier\_high <- subset(us\_quarters,DenverMint > denv\_out\_high)

> denver\_outlier\_high$State

[1] "Connecticut" "Virginia"

> #finding outliers towars lower end

> denv\_out\_low <- fivnum\_denv[2]-1.5\*(fivnum\_denv[4]-fivnum\_denv[2])

> denver\_outlier\_low <- subset(us\_quarters,DenverMint < denv\_out\_low)

> denver\_outlier\_low

[1] State DenverMint PhillyMint

<0 rows> (or 0-length row.names)

> #so there is no outliers in lower ends data

>

>

> #for Phillymint

> fivnum\_philly <- fivenum(PhillyMint)

> #finding outliers towards upper end

> philly\_out <- fivnum\_philly[4]+1.5\* (fivnum\_philly[4]-fivnum\_philly[2])

> philly\_outlier <- subset(us\_quarters,PhillyMint>philly\_out)

> #outlier states are

> philly\_outlier$State

[1] "Connecticut" "Massachusetts" "Maryland" "South Carolina" "New Hampshire" "Virginia" "New York" "North Carolina"

>

> #finding outliers towards lower end.

> philly\_out\_low <- fivnum\_philly[2]-1.5\*(fivnum\_philly[4]-fivnum\_philly[2])

> philly\_outlier\_low <- subset(us\_quarters,PhillyMint< philly\_out\_low)

> philly\_outlier\_low

[1] State DenverMint PhillyMint

<0 rows> (or 0-length row.names)

> #so there is no outliers in lower ends

-----------------------------------------------------------------------------------------------------------------------------

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Part 3**

-------------------------------------------------------------------------------------------------------------------------

**Code section :a**

stocks <- read.csv("https://people.bu.edu/kalathur/datasets/stocks.csv")

#a)

#removing date column from data set

stocks <- stocks[,colnames(stocks)!="Date"]

# drawing pair plot between different pairs among six stocks.

pairs(stocks,pch=16)

**Console Section :a**

> stocks <- read.csv("https://people.bu.edu/kalathur/datasets/stocks.csv")

> #a)

> #removing date column from data set

> stocks <- stocks[,colnames(stocks)!="Date"]

> # drawing pair plot between different pairs among six stocks.

> pairs(stocks,pch=16)

**Plot :a**

A picture containing text, diagram, pattern

Description automatically generated

-----------------------------------------------------------------------------------------------------------------------------

**Code section :b**

#b)

#Correlation matrix among the six stocks in the data set

round(cor(stocks),digits = 2)

**Console section: b**

> #b)

> #Correlation matrix among the six stocks in the data set

> round(cor(stocks),digits = 2)

MSFT AAPL GOOG FB AMZN TSLA

MSFT 1.00 0.90 0.95 0.68 0.64 0.71

AAPL 0.90 1.00 0.79 0.54 0.59 0.73

GOOG 0.95 0.79 1.00 0.85 0.67 0.47

FB 0.68 0.54 0.85 1.00 0.66 0.05

AMZN 0.64 0.59 0.67 0.66 1.00 0.34

TSLA 0.71 0.73 0.47 0.05 0.34 1.00

-----------------------------------------------------------------------------------------------------------------------------

#c)

1. Google and Microsoft stocks has the strongest positive correlations (0.95)

2.Tesla and Facebook has weakest positive correlations (0.05)

3 there is no negative correlation among all six stocks which means there is no

such stock which increases its value while other decrease.

4 There is also strongest relationships between Microsoft and apple (0.90) which

means increase or decrease in value of Microsoft stock also increase or decreases value of apple stock.(90% of time)

--------------------------------------------------------------------------------------------------------------------------

**Code section : d**

cm <- round(cor(stocks),digits = 2)

#names of stocks

stock\_names <- rownames(cm)

for (i in seq(1, ncol(cm))) { #looping through each row of correlation matrix

pick\_row <- cm[i,] # in every loop, it picks one row

#on each row it sorts the values, remove the first index value which is not important and then finds top 3 values

top3\_stocks <- sort (pick\_row,decreasing = TRUE)[-1][1:3]

top3\_stocks\_names <- names(top3\_stocks) #names for top 3 stocks in each row

cat("Top 3 for Stock",stock\_names[i],"\n",top3\_stocks\_names,"\n",top3\_stocks,"\n","\n")

}

**Console section :d**

> cm <- round(cor(stocks),digits = 2)

> #names of stocks

> stock\_names <- rownames(cm)

> for (i in seq(1,ncol(cm))) { #looping through each row of correlation matrix

+ pick\_row <- cm[i,] # in every loop, it pick one row

+ #on each row it sorts the values, remove the first index value which is not important and then finds top 3 values

+ top3\_stocks <- sort(pick\_row,decreasing = TRUE)[-1][1:3]

+ top3\_stocks\_names <- names(top3\_stocks) #names for top 3 stocks in each row

+ cat("Top 3 for Stock",stock\_names[i],"\n",top3\_stocks\_names,"\n",top3\_stocks,"\n","\n")

+

+ }

Top 3 for Stock MSFT

GOOG AAPL TSLA

0.95 0.9 0.71

Top 3 for Stock AAPL

MSFT GOOG TSLA

0.9 0.79 0.73

Top 3 for Stock GOOG

MSFT FB AAPL

0.95 0.85 0.79

Top 3 for Stock FB

GOOG MSFT AMZN

0.85 0.68 0.66

Top 3 for Stock AMZN

GOOG FB MSFT

0.67 0.66 0.64

Top 3 for Stock TSLA

AAPL MSFT GOOG

0.73 0.71 0.47

---------------------------------------------------------------------------------------------------------------------------

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Part 4

---------------------------------------------------------------------------------------------------------------------------

Code Section :a

#a)

hist\_scores <- hist(Score,col = "lightblue")

breaks\_score <- hist\_scores$breaks

l <- length(breaks\_score)

count <- hist\_scores$counts

i <- 1

while (i<l) {

start\_val <- breaks\_score[i] #starting value in each interval

end\_val <- breaks\_score[i+1] #ending value in each interval

interval\_count <- count[i] #counts in each interval

cat(interval\_count,"students in range (",start\_val,",",end\_val,"]","\n")

i <- i+1

}

Console section :b

#a)

> hist\_scores <- hist(Score,col = "lightblue")

> breaks\_score <- hist\_scores$breaks

> l <- length(breaks\_score)

> count <- hist\_scores$counts

> i <- 1

> while (i<l) {

+ start\_val <- breaks\_score[i] #starting value in each interval

+ end\_val <- breaks\_score[i+1] #ending value in each interval

+ interval\_count <- count[i] #counts in each interval

+ cat(interval\_count,"students in range (",start\_val,",",end\_val,"]","\n")

+ i <- i+1

+ }

3 students in range ( 35 , 40 ]

4 students in range ( 40 , 45 ]

10 students in range ( 45 , 50 ]

13 students in range ( 50 , 55 ]

17 students in range ( 55 , 60 ]

27 students in range ( 60 , 65 ]

13 students in range ( 65 , 70 ]

8 students in range ( 70 , 75 ]

3 students in range ( 75 , 80 ]

2 students in range ( 80 , 85 ]

Plot :a

A picture containing diagram, plot, line, plan

Description automatically generated

----------------------------------------------------------------------------------------------------------------------------

**Code section :a**

#b)

#costomizing the above scores

hist\_scores2 <- hist(Score,col = "lightpink",breaks = seq(30,90,length.out = 4))

len\_breaks <- length(hist\_scores2$breaks)

count\_interval <- hist\_scores2$counts

letter\_grade <-LETTERS[(len\_breaks-1):1]#intervals is one less than number of breaks

letter\_grade

j <- 1

while (j<len\_breaks) {

start\_score\_interval <- hist\_scores2$breaks[j] #starting value in each interval

end\_score\_interval<- hist\_scores2$breaks[j+1] #ending value in each interval

count\_in\_interval <- count\_interval[j] #counts in each interval

cat(count\_in\_interval,"students in",letter\_grade[j],"grade range (",

start\_score\_interval,",",end\_score\_interval,"]","\n")

j <- j+1

}

**Console section :b**

> #b)

> #costomizing the above scores

> hist\_scores2 <- hist(Score,col = "lightpink",breaks = seq(30,90,length.out = 4))

> len\_breaks <- length(hist\_scores2$breaks)

> count\_interval <- hist\_scores2$counts

> letter\_grade <-LETTERS[(len\_breaks-1):1]#intervals is one less than number of breaks

> letter\_grade

[1] "C" "B" "A"

>

>

> j <- 1

> while (j<len\_breaks) {

+ start\_score\_interval <- hist\_scores2$breaks[j] #starting value in each interval

+ end\_score\_interval<- hist\_scores2$breaks[j+1] #ending value in each interval

+ count\_in\_interval <- count\_interval[j] #counts in each interval

+ cat(count\_in\_interval,"students in",letter\_grade[j],"grade range (",

+ start\_score\_interval,",",end\_score\_interval,"]","\n")

+ j <- j+1

+

+ }

17 students in C grade range ( 30 , 50 ]

70 students in B grade range ( 50 , 70 ]

13 students in A grade range ( 70 , 90 ]

**Plot section:b**

A picture containing diagram, line, plot, rectangle

Description automatically generated

------------------------------------------------------------------------------------------------------------------------

The End!

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*