Business Analytics for Strategic Decision Making

Note: Install packages

```
library(lubridate)
```

```
library(RColorBrewer)
library(factoextra)
```

Steps to perform:

1. Import the input file. Check if the variable with date values has been imported appropriately

```
data <- read.table('marketing_campaign.csv', sep = '\t', header = TRUE)

head(data)
```

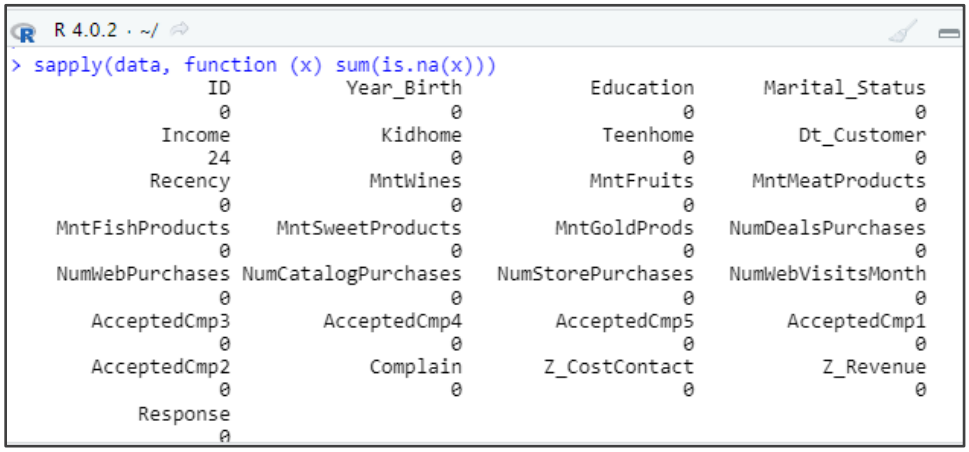
```
str(data)
```

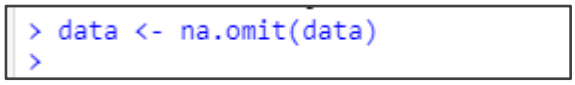
```
##     ID Year_Birth  Education Marital_Status Income Kidhome Teenhome Dt_Customer
## 1 5524       1957 Graduation         Single  58138       0        0  04-09-2012
## 2 2174       1954 Graduation         Single  46344       1        1  08-03-2014
## 3 4141       1965 Graduation       Together  71613       0        0  21-08-2013
## 4 6182       1984 Graduation       Together  26646       1        0  10-02-2014
## 5 5324       1981        PhD        Married  58293       1        0  19-01-2014
## 6 7446       1967     Master       Together  62513       0        1  09-09-2013
##   Recency MntWines MntFruits MntMeatProducts MntFishProducts MntSweetProducts
## 1      58      635        88             546             172               88
## 2      38       11         1               6               2                1
## 3      26      426        49             127             111               21
## 4      26       11         4              20              10                3
## 5      94      173        43             118              46               27
## 6      16      520        42              98               0               42
##   MntGoldProds NumDealsPurchases NumWebPurchases NumCatalogPurchases
## 1           88                 3               8                  10
## 2            6                 2               1                   1
## 3           42                 1               8                   2
## 4            5                 2               2                   0
## 5           15                 5               5                   3
## 6           14                 2               6                   4
##   NumStorePurchases NumWebVisitsMonth AcceptedCmp3 AcceptedCmp4 AcceptedCmp5
## 1                 4                 7            0            0            0
## 2                 2                 5            0            0            0
## 3                10                 4            0            0            0
## 4                 4                 6            0            0            0
## 5                 6                 5            0            0            0
```

```
## 'data.frame':    2240 obs. of  29 variables:
##  $ ID                 : int  5524 2174 4141 6182 5324 7446 965 6177 4855 5899 ...
##  $ Year_Birth         : int  1957 1954 1965 1984 1981 1967 1971 1985 1974 1950 ...
##  $ Education          : chr  "Graduation" "Graduation" "Graduation" "Graduation" ...
##  $ Marital_Status     : chr  "Single" "Single" "Together" "Together" ...
##  $ Income             : int  58138 46344 71613 26646 58293 62513 55635 33454 30351 5648 ...
##  $ Kidhome            : int  0 1 0 1 1 0 0 1 1 1 ...
##  $ Teenhome           : int  0 1 0 0 0 1 1 0 0 1 ...
##  $ Dt_Customer        : chr  "04-09-2012" "08-03-2014" "21-08-2013" "10-02-2014" ...
##  $ Recency            : int  58 38 26 26 94 16 34 32 19 68 ...
##  $ MntWines           : int  635 11 426 11 173 520 235 76 14 28 ...
##  $ MntFruits          : int  88 1 49 4 43 42 65 10 0 0 ...
##  $ MntMeatProducts    : int  546 6 127 20 118 98 164 56 24 6 ...
##  $ MntFishProducts    : int  172 2 111 10 46 0 50 3 3 1 ...
##  $ MntSweetProducts   : int  88 1 21 3 27 42 49 1 3 1 ...
##  $ MntGoldProds       : int  88 6 42 5 15 14 27 23 2 13 ...
##  $ NumDealsPurchases  : int  3 2 1 2 5 2 4 2 1 1 ...
##  $ NumWebPurchases    : int  8 1 8 2 5 6 7 4 3 1 ...
##  $ NumCatalogPurchases: int  10 1 2 0 3 4 3 0 0 0 ...
##  $ NumStorePurchases  : int  4 2 10 4 6 10 7 4 2 0 ...
##  $ NumWebVisitsMonth  : int  7 5 4 6 5 6 6 8 9 20 ...
##  $ AcceptedCmp3       : int  0 0 0 0 0 0 0 0 0 1 ...
##  $ AcceptedCmp4       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ AcceptedCmp5       : int  0 0 0 0 0 0 0 0 0 0 ...
```

2. Find the variables with missing values. If the proportion of missing values is less than 5%, then delete the rows

```
sapply(data, function (x) sum(is.na(x)))

data <- na.omit(data)
```





3. Calculate the latest and oldest customer's enrolment date in the records

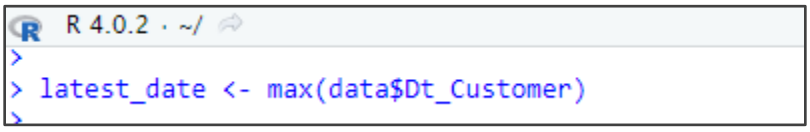
```
cat('Class of Dt_Customer is :', class(data$Dt_Customer), '\n')

print('...Converting Class ...')

data$Dt_Customer <- as.Date(data$Dt_Customer, "%d-%m-%Y")

recent <-max(data$Dt_Customer)

oldest <- min(data$Dt_Customer)

print(paste('Oldest enrolment : ', oldest))

print(paste('Newest enrolment : ', recent))
```

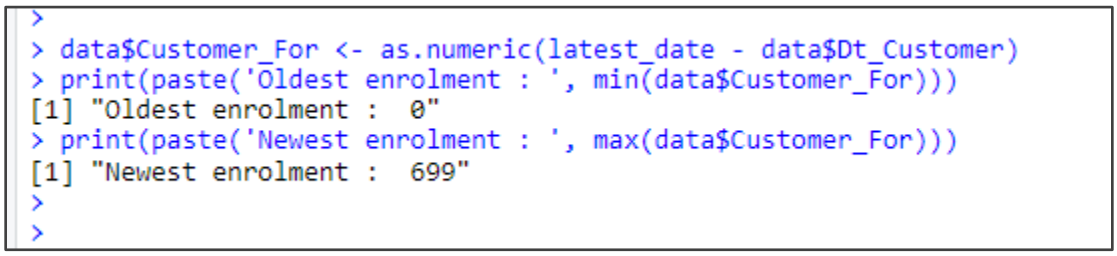
```
>
> cat('Class of Dt_Customer is :', class(data$Dt_Customer), '\n')
Class of Dt_Customer is : character
> print('...Converting Class ...')
[1] "...Converting Class ..."
> data$Dt_Customer <- as.Date(data$Dt_Customer, "%d-%m-%Y")
> recent <-max(data$Dt_Customer)
> oldest <- min(data$Dt_Customer)
> print(paste('Oldest enrolment : ', oldest))
[1] "Oldest enrolment :  2012-07-30"
> print(paste('Newest enrolment : ', recent))
[1] "Newest enrolment :  2014-06-29"
>
```

4. Create a feature "Customer_For" for the number of days the customers started to shop in the store relative to the last recorded date

```
latest_date <- max(data$Dt_Customer)
```

```
data$Customer_For <- as.numeric(latest_date - data$Dt_Customer)

print(paste('Oldest enrolment : ', min(data$Customer_For)))

print(paste('Newest enrolment : ', max(data$Customer_For)))
```





5. Find the "Age" of customers by the "Year_Birth" indicating the birth year

```
present_year <- year(Sys.Date())

data$Age <- present_year - data$Year_Birth

summary(data$Age)
```

```
>
> present_year <- year(Sys.Date())
> data$Age <- present_year - data$Year_Birth
> summary(data$Age)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  26.00   45.00   52.00   53.18   63.00  129.00
>
>
```

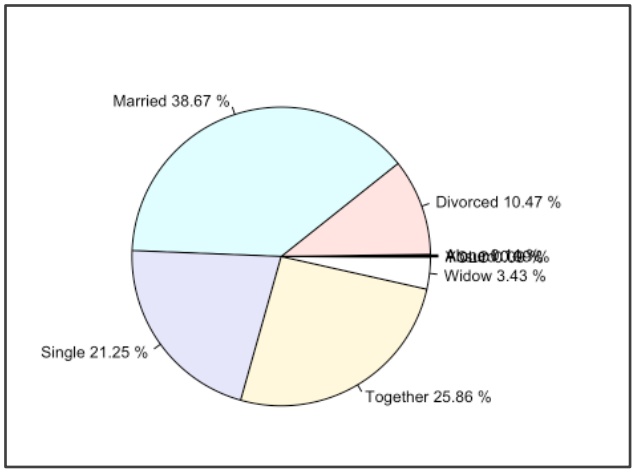
6. Create a feature "Spent" indicating the total amount spent by the customer in various categories over two years

```
subdata <- data[,grep('Mnt', names(data))]

data$Spent <- apply(subdata, FUN = sum, MARGIN = 1  )

summary(data$Spent)
```

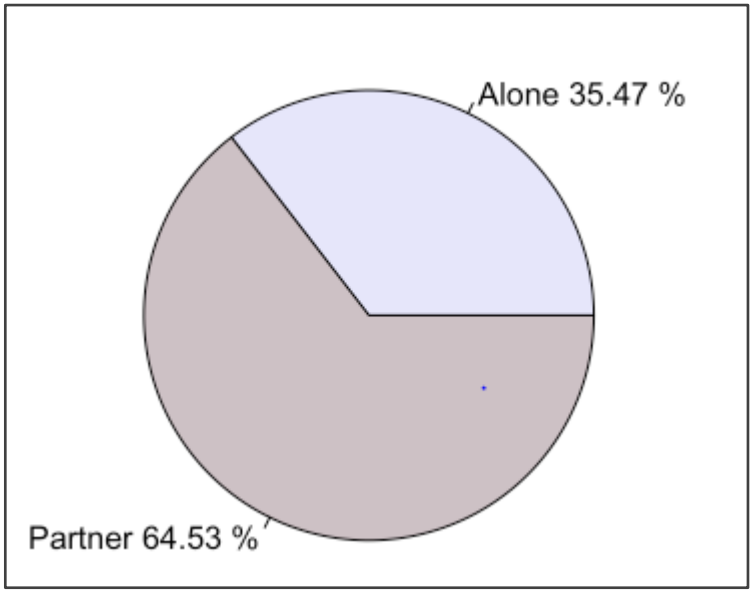
```
>
> subdata <- data[,grep('Mnt', names(data))]
> # use apply to do row wise addition
> data$Spent <- apply(subdata, FUN = sum, MARGIN = 1  )
> summary(data$Spent)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    5.0    69.0   396.5   607.1  1048.0  2525.0
>
```

7. Create another feature "Living_With" out of "Marital_Status" to extract the living situation of couples. Consider a value ‘Partner’ for the variable "Living_With" for the instances where "Marital_Status" is either “Married” or “Together”. The rest can be taken as ‘Alone’

```
count <- table(data$Marital_Status)

perc <- round(count/ sum(count) * 100, 2)

label <- paste(names(count), perc, '%')

pie(perc, labels =label , cex = 0.75 )
```



```
data$Living_With <- ifelse(data$Marital_Status %in% c("Married", "Together"),
"Partner", "Alone")

count <- table(data$Living_With)

perc <- round(count/sum(count)*100, 2)

label <- paste(names(count), perc, '%')

pie(perc, labels =label , col = c('lavender', 'lavenderblush3') ,

    main = 'Living With')
```



8. Create a feature "Children" to indicate the total number of kids and teenagers in a household

```
data$Children <- data$Kidhome + data$Teenhome

summary(data$Children)
```

```
>
> data$Children <- data$Kidhome + data$Teenhome
> summary(data$Children)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.0000  0.0000  1.0000  0.9472  1.0000  3.0000
>
>
```

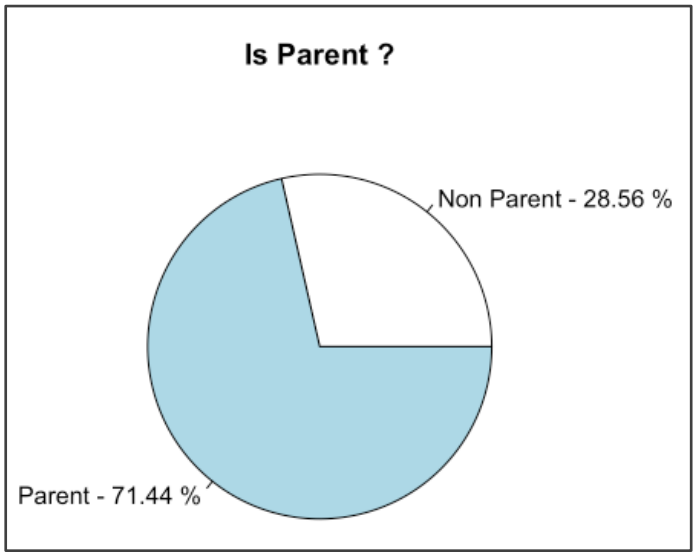
9. To get further clarity on a household, create a feature indicating "Family_Size"

```
data$Family_Size <- ifelse(data$Living_With == "Partner", 2, 1) +
data$Children

summary(data$Family_Size )
```

```
>
> data$Family_Size <- ifelse(data$Living_With == "Partner", 2, 1) + data$Children
> summary(data$Family_Size )
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   2.000   3.000   2.593   3.000   5.000
>
```

10. Create a feature "Is_Parent" to indicate the parenthood status

```
data$Is_Parent <- ifelse(data$Children > 0, 1, 0)
cnt <- table(data$Is_Parent)
names(cnt) <- ifelse(names(cnt)==0, 'Non Parent', 'Parent')
perc <- round(cnt/sum(cnt) * 100, 2 )
pie(cnt, label = paste(names(cnt), '-', perc, '%'), main = 'Is Parent ?')
```

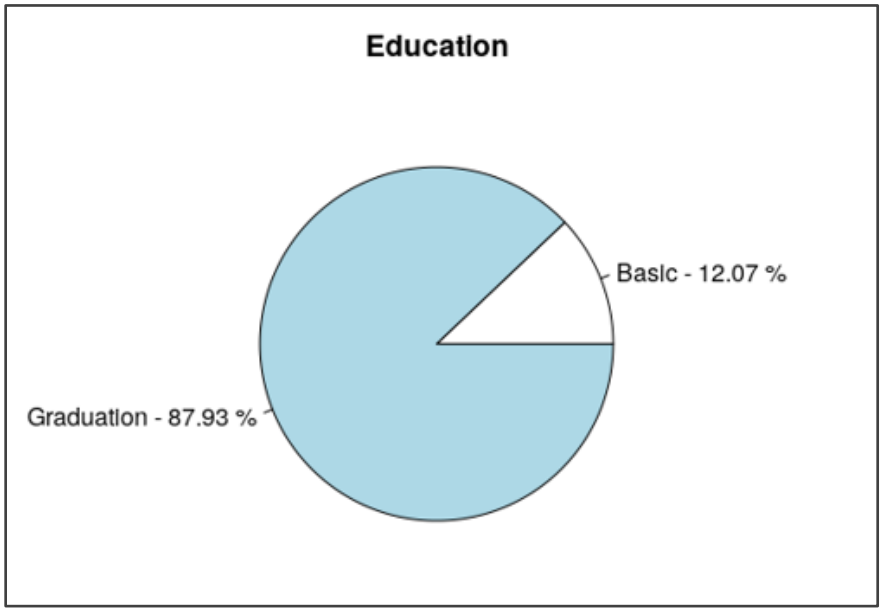


11. Keep only two categories in the field – ‘Education’ – Undergraduate, Graduate

```
table(data$Education)
```

```
>
> table(data$Education)

  2n Cycle      Basic Graduation     Master        PhD
       200          54       1116        365        481
>
```

```
data$Education <- ifelse(data$Education %in% c("Basic", "2n Cycle"),
                         "Basic",
                         ifelse(data$Education %in% c("Master", "PhD"),
                                'Higher Degree',
                                'Graduation'))
cnt <- table(data$Education)
perc <- round(cnt/sum(cnt) * 100, 2 )
pie(cnt, label = paste(names(cnt), '-', perc, '%'), main = 'Education')
```



12. For the sake of clarity, change the name of the variables as:

MntWines	Wines
MntFruits	Fruits
MntMeatProducts	MeatProducts
MntFishProducts	FishProducts
MntSweetProducts	SweetsProducts
MntGoldProds	GoldProds

```
names(data) <- sub('Mnt', '', names(data))

names(data)
```

```
> names(data)
 [1] "ID"                 "Year_Birth"          "Education"
 [4] "Marital_Status"     "Income"              "Kidhome"
 [7] "Teenhome"           "Dt_Customer"         "Recency"
[10] "Wines"              "Fruits"              "MeatProducts"
[13] "FishProducts"       "SweetProducts"       "GoldProds"
[16] "NumDealsPurchases"  "NumWebPurchases"     "NumCatalogPurchases"
[19] "NumStorePurchases"  "NumWebVisitsMonth"   "AcceptedCmp3"
[22] "AcceptedCmp4"       "AcceptedCmp5"        "AcceptedCmp1"
[25] "AcceptedCmp2"       "Complain"            "Z_CostContact"
[28] "Z_Revenue"          "Response"            "Customer_For"
[31] "Age"                "Spent"               "Living_With"
[34] "Children"           "Family_Size"         "Is_Parent"
>
```

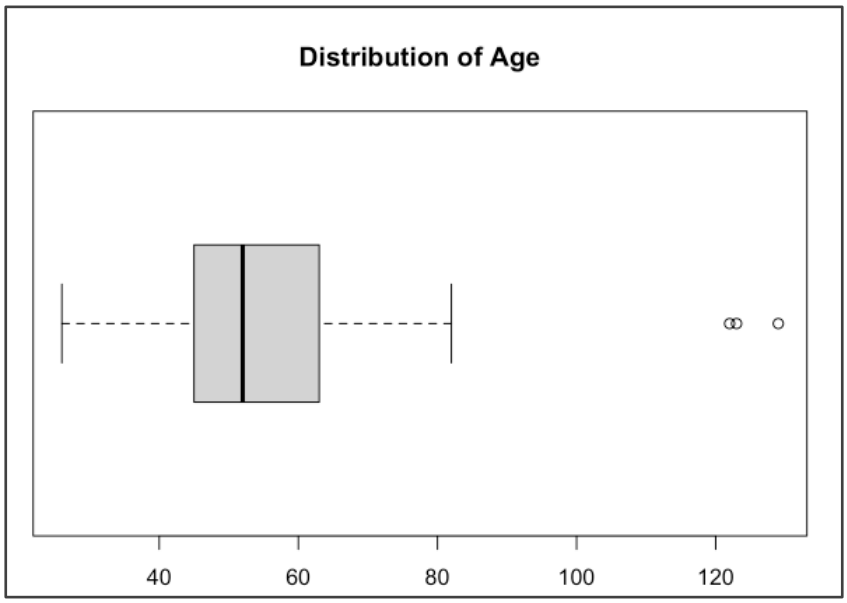
13. Drop the redundant columns: "Marital_Status", "Dt_Customer", "Z_CostContact", "Z_Revenue", "Year_Birth", and "ID"

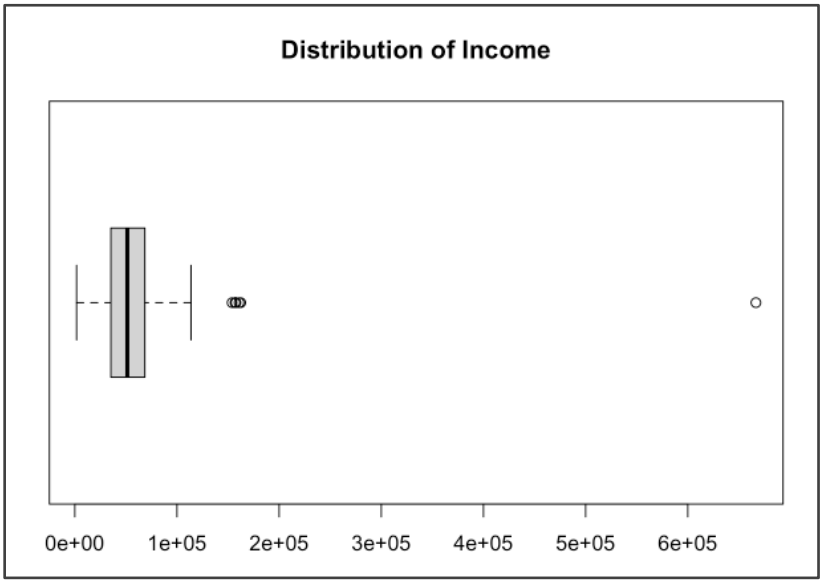
```
data <- data[,- which(names(data) %in% c("Marital_Status","Dt_Customer",
"Z_CostContact","Z_Revenue","Year_Birth","ID"))]

head(data)
```

```
##       Education Income Kidhome Teenhome Recency Wines Fruits MeatProducts
## 1      Graduate  58138       0        0      58   635     88          546
## 2      Graduate  46344       1        1      38    11      1            6
## 3      Graduate  71613       0        0      26   426     49          127
## 4      Graduate  26646       1        0      26    11      4           20
## 5 Higher Degree  58293       1        0      94   173     43          118
## 6 Higher Degree  62513       0        1      16   520     42           98
##   FishProducts SweetProducts GoldProds NumDealsPurchases NumWebPurchases
## 1          172            88        88                 3                8
## 2            2             1         6                 2                1
## 3          111            21        42                 1                8
## 4           10             3         5                 2                2
## 5           46            27        15                 5                5
## 6            0            42        14                 2                6
##   NumCatalogPurchases NumStorePurchases NumWebVisitsMonth AcceptedCmp3
## 1                  10                 4                 7            0
## 2                   1                 2                 5            0
## 3                   2                10                 4            0
## 4                   0                 4                 6            0
## 5                   3                 6                 5            0
## 6                   4                10                 6            0
##   AcceptedCmp4 AcceptedCmp5 AcceptedCmp1 AcceptedCmp2 Complain Response
## 1            0            0            0            0        0        1
## 2            0            0            0            0        0        0
## 3            0            0            0            0        0        0
## 4            0            0            0            0        0        0
```

14. Create box plots for age and income. Identify the outliers and delete rows with the outliers

```
boxplot(data$Age, horizontal = TRUE, main = "Distribution of Age")

boxplot(data$Income, horizontal = TRUE, main = "Distribution of Income")
```


Distribution of Age
40
60
80
100
120


Distribution of Income
0e+00
1e+05
2e+05
3e+05
4e+05
5e+05
6e+05

```
finding_limits <- function(x){
  q1 <- quantile(x, 0.25)
  q3 <- quantile(x, 0.25)
  iqr <- IQR(x)  lb <- q1 - (1.5 * iqr)
  ub <- q3 + (1.5 * iqr)
  return(c(lb, ub))  }
```

```
limits_age <- finding_limits(data$Age)
limits_inc <- finding_limits(data$Income)
print(paste('Age limits :' , paste(limits_age, collapse = ' - ')))
print(paste('Income limits :' , paste(limits_inc, collapse = ' - ')))
```

```
> limits_age <- finding_limits(data$Age)
> limits_inc <- finding_limits(data$Income)
> print(paste('Age limits :' , paste(limits_age, collapse = ' - ')))
[1] "Age limits : 18 - 72"
> print(paste('Income limits :' , paste(limits_inc, collapse = ' - ')))
[1] "Income limits : -14525.5 - 85131.5"
>
```

```
data <- data[(data$Age >limits_age[1]) & (data$Age < limits_age[2]),]

data <- data[(data$Income >limits_inc[1]) &

             (data$Income < limits_inc[2]),]
```

```
str(data)

summary(data)
```

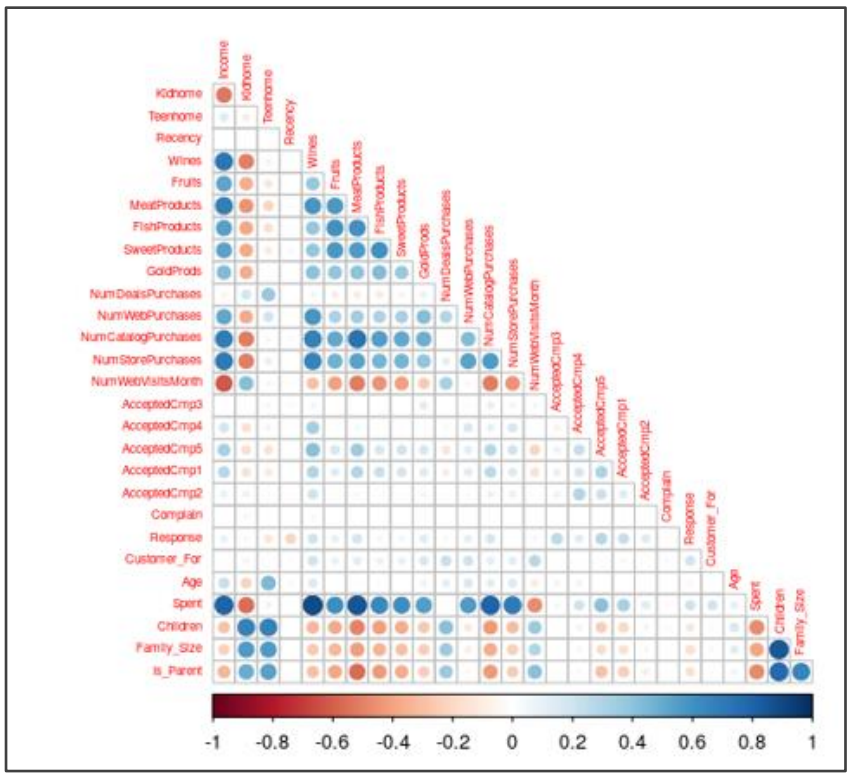
```
##   Education              Income         Kidhome          Teenhome
##  Length:1988        Min.   : 1730   Min.   :0.0000   Min.   :0.0000
##  Class :character   1st Qu.:34069   1st Qu.:0.0000   1st Qu.:0.0000
##  Mode  :character   Median :49107   Median :0.0000   Median :1.0000
##                     Mean   :49412   Mean   :0.4754   Mean   :0.5231
##                     3rd Qu.:65646   3rd Qu.:1.0000   3rd Qu.:1.0000
##                     Max.   :85072   Max.   :2.0000   Max.   :2.0000
##     Recency          Wines            Fruits        MeatProducts
##  Min.   : 0.00   Min.   :   0.0   Min.   :  0.00   Min.   :   0.0
##  1st Qu.:25.00   1st Qu.:  21.0   1st Qu.:  1.00   1st Qu.:  14.0
##  Median :49.00   Median : 154.0   Median :  7.00   Median :  59.0
##  Mean   :49.25   Mean   : 278.2   Mean   : 24.29   Mean   : 144.8
##  3rd Qu.:75.00   3rd Qu.: 456.2   3rd Qu.: 30.00   3rd Qu.: 199.5
##  Max.   :99.00   Max.   :1486.0   Max.   :199.00   Max.   :1725.0
##   FishProducts    SweetProducts      GoldProds      NumDealsPurchases
##  Min.   :  0.00   Min.   :  0.00   Min.   :  0.00   Min.   : 0.000
##  1st Qu.:  2.00   1st Qu.:  1.00   1st Qu.:  8.00   1st Qu.: 1.000
##  Median : 11.00   Median :  7.00   Median : 23.00   Median : 2.000
##  Mean   : 34.91   Mean   : 24.65   Mean   : 43.26   Mean   : 2.404
##  3rd Qu.: 42.00   3rd Qu.: 30.00   3rd Qu.: 56.00   3rd Qu.: 3.000
##  Max.   :259.00   Max.   :198.00   Max.   :321.00   Max.   :15.000
##  NumWebPurchases  NumCatalogPurchases NumStorePurchases NumWebVisitsMonth
##  Min.   : 0.000   Min.   : 0.000      Min.   : 0.000    Min.   : 0.000
##  1st Qu.: 2.000   1st Qu.: 0.000      1st Qu.: 3.000    1st Qu.: 4.000
##  Median : 3.000   Median : 1.000      Median : 5.000    Median : 6.000
##  Mean   : 3.991   Mean   : 2.425      Mean   : 5.661    Mean   : 5.517
##  3rd Qu.: 6.000   3rd Qu.: 4.000      3rd Qu.: 8.000    3rd Qu.: 7.000
##  Max.   :25.000   Max.   :28.000      Max.   :13.000    Max.   :20.000
```

```
## 'data.frame':    1988 obs. of  30 variables:
##  $ Education          : chr  "Graduate" "Graduate" "Graduate" "Graduate" ...
##  $ Income             : int  58138 46344 71613 26646 58293 62513 55635 33454 30351 7500 ...
##  $ Kidhome            : int  0 1 0 1 1 0 0 1 1 0 ...
##  $ Teenhome           : int  0 1 0 0 0 1 1 0 0 0 ...
##  $ Recency            : int  58 38 26 26 94 16 34 32 19 59 ...
##  $ Wines              : int  635 11 426 11 173 520 235 76 14 6 ...
##  $ Fruits             : int  88 1 49 4 43 42 65 10 0 16 ...
##  $ MeatProducts       : int  546 6 127 20 118 98 164 56 24 11 ...
##  $ FishProducts       : int  172 2 111 10 46 0 50 3 3 11 ...
##  $ SweetProducts      : int  88 1 21 3 27 42 49 1 3 1 ...
##  $ GoldProds          : int  88 6 42 5 15 14 27 23 2 16 ...
##  $ NumDealsPurchases  : int  3 2 1 2 5 2 4 2 1 1 ...
##  $ NumWebPurchases    : int  8 1 8 2 5 6 7 4 3 2 ...
##  $ NumCatalogPurchases: int  10 1 2 0 3 4 3 0 0 0 ...
##  $ NumStorePurchases  : int  4 2 10 4 6 10 7 4 2 3 ...
##  $ NumWebVisitsMonth  : int  7 5 4 6 5 6 6 8 9 8 ...
##  $ AcceptedCmp3       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ AcceptedCmp4       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ AcceptedCmp5       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ AcceptedCmp1       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ AcceptedCmp2       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Complain           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Response           : int  1 0 0 0 0 0 0 0 1 0 ...
##  $ Customer_For       : num  663 113 312 139 161 293 593 593 417 388 593 ...
##  $ Age                : num  65 68 57 38 41 55 51 37 48 46 ...
##  $ Spent              : int  1617 27 776 53 422 716 590 169 46 61 ...
##  $ Living_With        : chr  "Alone" "Alone" "Partner" "Partner" ...
##  $ Children           : int  0 2 0 1 1 1 1 1 1 0 ...
```

15. Find out the correlation between variables. Create a heatmap to visualize the correlation plot

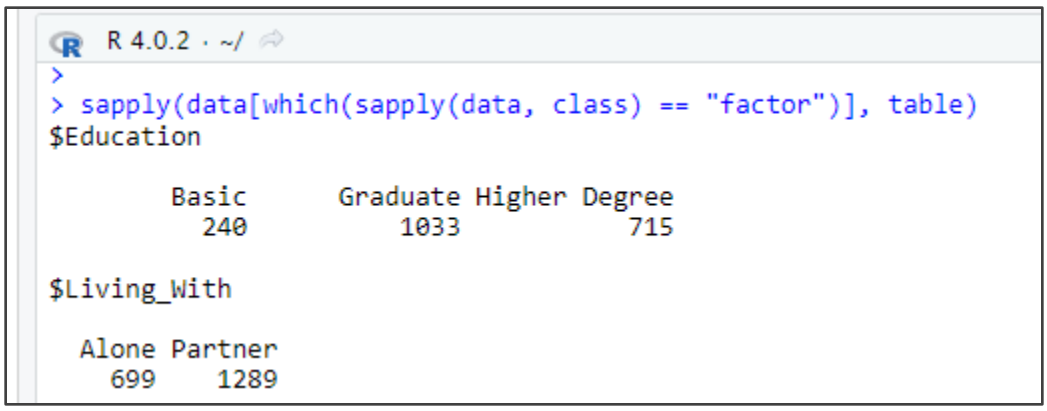
```
drop_cols <- which(sapply(data, class) == "character")

corr_mat <- cor(data[-drop_cols])

corrplot::corrplot(corr_mat, type = "lower",  tl.pos = 'ld',

                   tl.cex = 0.5, diag = FALSE,

                   main = 'Correlation Plot')
```


```
drop_cols <- which(sapply(data, class) == "character")

corr_mat <- cor(data[-drop_cols])

corrplot::corrplot(corr_mat, type = "lower",  tl.pos = 'ld',

                   tl.cex = 0.5, diag = FALSE,

                   main = 'Correlation Plot')
```



16. Prepare the data for cluster analysis. Categorical variables must be incorporated in clustering. Perform necessary encoding techniques to transform the categorical variables

```
data[which(sapply(data, class) == "character")] <- lapply(data[which(sapply(data, class) == "character")], as.factor)

sapply(data[which(sapply(data, class) == "factor")], table)
```




```
data[which(sapply(data, class) == "factor")] <- sapply(data[which(sapply(data, class) == "factor")], as.numeric)
```

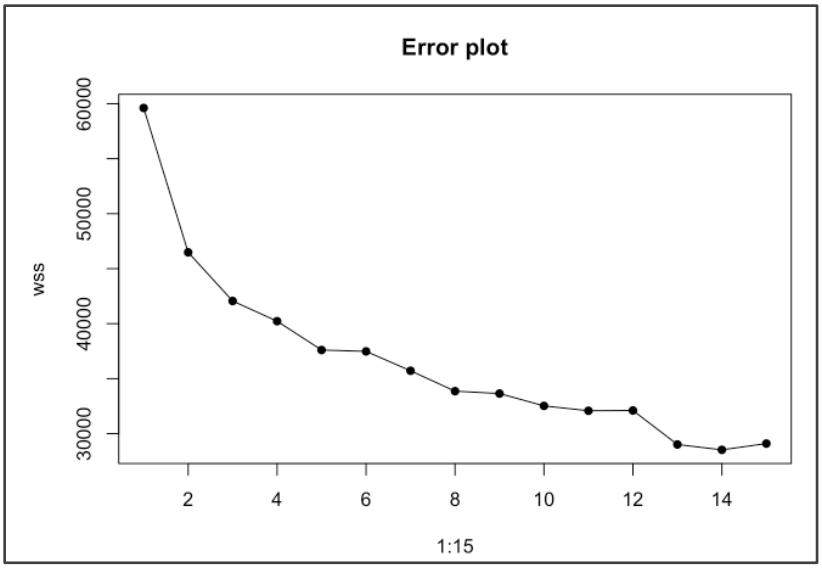
17. Find the appropriate number of clusters using the elbow method

```
scaled_data <- scale(data)

apply(scaled_data, 2, function(x) round(mean(x),1))

apply(scaled_data, 2, function(x) round(sd(x),1))
```

```
##           Education              Income             Kidhome            Teenhome 
##                   0                   0                   0                   0 
##             Recency               Wines             Fruits        MeatProducts 
##                   0                   0                   0                   0 
##       FishProducts       SweetProducts          GoldProds   NumDealsPurchases 
##                   0                   0                   0                   0 
##     NumWebPurchases NumCatalogPurchases   NumStorePurchases   NumWebVisitsMonth 
##                   0                   0                   0                   0 
##       AcceptedCmp3        AcceptedCmp4        AcceptedCmp5        AcceptedCmp1 
##                   0                   0                   0                   0 
##       AcceptedCmp2            Complain            Response        Customer_For 
##                   0                   0                   0                   0 
##                Age               Spent         Living_With            Children 
##                   0                   0                   0                   0 
##        Family_Size           Is_Parent 
##                   0                   0
```

```
##           Education              Income             Kidhome            Teenhome 
##                   1                   1                   1                   1 
##             Recency               Wines             Fruits        MeatProducts 
##                   1                   1                   1                   1 
##       FishProducts       SweetProducts          GoldProds   NumDealsPurchases 
##                   1                   1                   1                   1 
##     NumWebPurchases NumCatalogPurchases   NumStorePurchases   NumWebVisitsMonth 
##                   1                   1                   1                   1 
##       AcceptedCmp3        AcceptedCmp4        AcceptedCmp5        AcceptedCmp1 
##                   1                   1                   1                   1 
##       AcceptedCmp2            Complain            Response        Customer_For 
##                   1                   1                   1                   1 
##                Age               Spent         Living_With            Children 
##                   1                   1                   1                   1 
##        Family_Size           Is_Parent 
##                   1                   1
```

```
wss <- sapply(1:15,
              function(k){kmeans(scaled_data, k , iter.max = 1000)$tot.withinss})
plot(1:15, wss, pch = 16, type = 'o', main = "Error plot")
```



18. Perform cluster analysis using k-means

```
set.seed(1) # to maintain consistency for every execution

k_result <- kmeans(scaled_data, 4, iter.max = 1000 , )

k_result
```

```
## K-means clustering with 4 clusters of sizes 604, 473, 369, 542
##
## Cluster means:
##      Education     Income    Kidhome   Teenhome     Recency      Wines
## 1 -0.245036744 -0.9850068  0.5259227 -0.8856076 -0.00952370 -0.7500692
## 2  0.181876305 -0.3760181  0.6452895  0.9264460  0.01177902 -0.6294915
## 3  0.003470963  1.2015922 -0.8334025 -0.9150227  0.04468982  0.9926229
## 4  0.111981395  0.6077734 -0.5818334  0.8013680 -0.03009171  0.7094343
##       Fruits MeatProducts FishProducts SweetProducts  GoldProds
## 1 -0.4653944   -0.5957489   -0.4625983    -0.4583944 -0.4621680
## 2 -0.5406387   -0.5958257   -0.5530849    -0.5379476 -0.5313951
## 3  1.0741748    1.5770392    1.1914824     1.0393914  0.6986028
## 4  0.2591325    0.1102037    0.1870138     0.2726643  0.5031641
##   NumDealsPurchases NumWebPurchases NumCatalogPurchases NumStorePurchases
## 1        -0.2715096      -0.6229140          -0.6913953        -0.7601125
## 2         0.2195058      -0.5551362          -0.6222889        -0.6210298
## 3        -0.6493030       0.3173337           1.2620605         0.8231740
## 4         0.5530597       0.9625892           0.4543267         0.8286049
##   NumWebVisitsMonth AcceptedCmp3 AcceptedCmp4 AcceptedCmp5 AcceptedCmp1
## 1        0.57093891   0.04629311   -0.2361351   -0.2227228  -0.20974113
## 2        0.23331560  -0.06736694   -0.1251493   -0.2227228  -0.21770274
## 3       -1.14156326   0.04464909    0.1593470    0.8750824   0.69453795
## 4       -0.06267258  -0.02319555    0.2638786   -0.1531973  -0.04912816
##   AcceptedCmp2    Complain    Response Customer_For        Age      Spent
## 1  -0.11051612  0.02675649 -0.03924405  -0.03914937 -0.6780709 -0.7805842
```

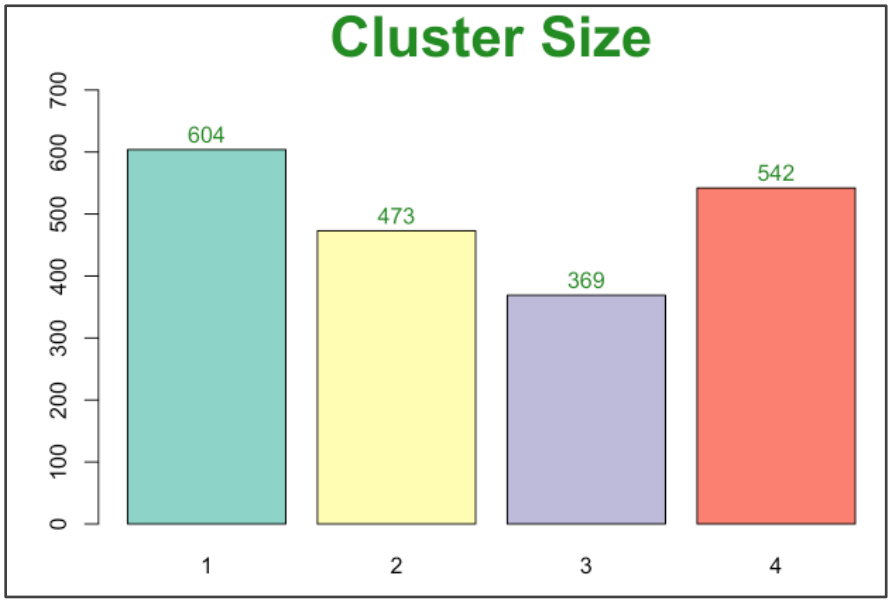
```
## 3 -0.10359104 -1.3033537  -1.1281943 -1.64706031
## 4  0.02152721  0.1645917   0.1470386  0.53115811
##
## Clustering vector:
##    1    2    3    4    5    6    7    8    9   12   13   14   15   17   20   21
##    3    2    3    1    1    4    4    1    1    1    3    2    1    2    1    1
##   22   24   25   26   27   29   30   31   32   33   34   36   37   38   39   41
##    3    4    4    1    2    1    3    1    1    2    2    2    4    1    1    4
##   42   43   45   46   47   48   50   51   52   53   54   55   56   57   58   60
##    1    2    1    3    1    1    4    4    3    1    3    4    3    3    1    4
##   61   62   63   64   66   67   69   70   71   73   74   75   76   77   78   79
##    3    4    4    4    1    2    4    4    3    4    4    1    1    3    3    1
##   80   81   82   83   84   85   86   87   88   89   90   94   95   96   97   98
##    4    1    1    1    1    3    2    2    4    3    2    1    1    1    4    2
##   99  100  101  102  106  107  108  109  111  112  113  115  116  118  119  120
##    3    1    1    2    1    2    4    1    4    3    4    1    2    4    2    1
##  121  122  123  124  126  127  128  130  131  132  133  135  136  137  138  139
##    4    1    1    1    3    3    2    4    4    4    3    2    3    1    2    2
##  140  142  144  145  146  147  148  149  150  151  152  153  154  155  157  158
##    2    4    4    1    4    1    1    2    1    2    4    4    1    4    4    2
##  159  160  161  163  164  166  167  168  169  170  171  172  173  174  175  176
##    2    3    2    1    3    2    3    1    3    2    2    1    2    2    1    4
##  178  179  180  181  182  184  185  186  187  188  189  190  191  192  194  195
##    2    2    3    2    1    1    1    1    2    4    3    2    1    3    1    2
##  196  197  198  200  201  202  203  205  206  207  208  209  210  211  212  213
##    2    3    3    2    4    3    4    2    1    1    2    2    4    2    3    2
##  214  215  216  217  219  220  221  222  223  224  225  226  227  228  229  230
##    2    3    1    2    2    4    1    3    4    1    4    1    4    4    3    1
##  231  232  233  234  235  236  237  238  239  241  242  243  244  245  246  247
```

```
## 2113 2114 2115 2116 2117 2118 2119 2120 2121 2122 2123 2124 2125 2126 2127 2128
##    2    2    4    4    1    4    3    1    1    2    1    4    2    3    4    3
## 2129 2130 2131 2132 2134 2135 2136 2137 2138 2139 2140 2141 2142 2143 2144 2145
##    3    2    1    3    1    3    4    2    1    2    2    1    2    2    1    4
## 2146 2147 2148 2149 2150 2151 2152 2153 2154 2155 2156 2157 2158 2159 2160 2161
##    3    1    2    2    2    2    3    1    2    1    1    2    2    4    4    4
## 2162 2163 2165 2166 2167 2170 2171 2172 2173 2174 2175 2176 2177 2178 2179 2180
##    1    4    1    1    4    2    4    3    4    4    4    3    3    4    4    2
## 2181 2182 2183 2184 2185 2186 2187 2188 2189 2190 2192 2194 2195 2196 2197 2198
##    1    4    1    1    2    4    4    3    3    1    2    3    4    2    1    4
## 2199 2200 2201 2202 2203 2204 2205 2206 2207 2208 2209 2210 2211 2214 2215 2216
##    2    2    1    4    4    4    1    2    4    2    1    2    4    3    1    2
## 2217 2219 2220 2221 2222 2223 2224 2225 2226 2227 2228 2230 2231 2232 2233 2235
##    1    1    1    4    3    1    2    4    4    4    4    2    1    4    1    1
## 2236 2238 2239 2240
##    4    3    4    2
##
## Within cluster sum of squares by cluster:
## [1]  8123.625  6210.406 12803.295 12521.151
##  (between_SS / total_SS =  33.5 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

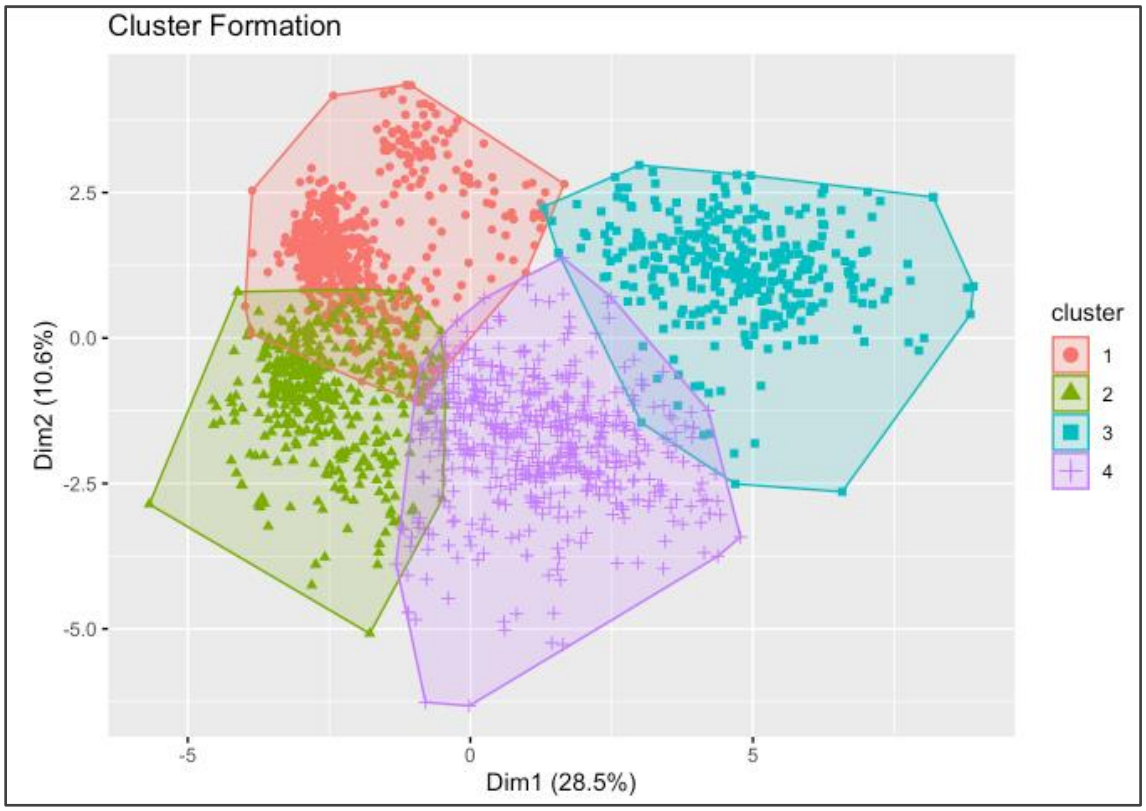
19. After clusters are formed, the business must define the clusters, perform cluster profiling, and describe each cluster in detail. Use appropriate visualizations to support your views

```
data_copy <- data

data_copy$cluster_label<- k_result$cluster
```

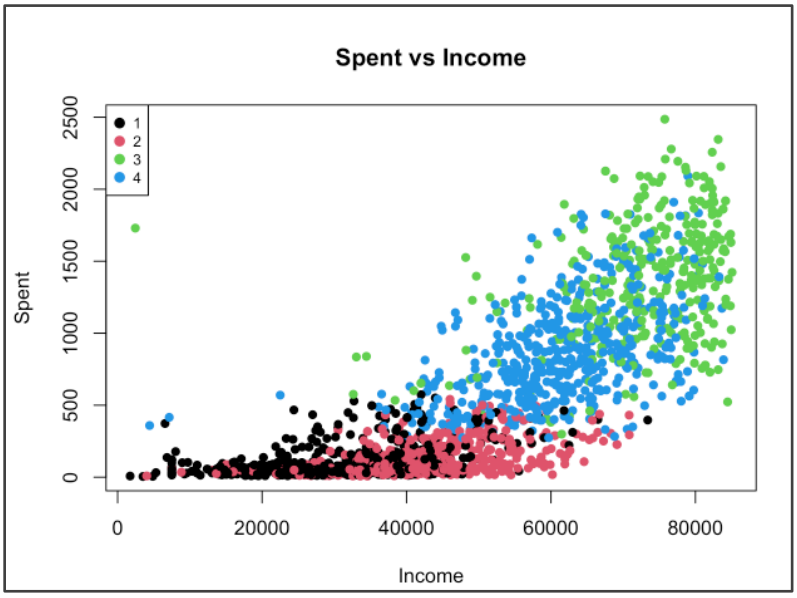
```
par(mar = c(5,4,4,2))

count <- table(data_copy$cluster_label)

myplot <- barplot(count, col = brewer.pal(4,'Set3'), ylim = c(0, 700))

text(myplot, count + 25, count, cex = 1, col = 'forestgreen')

title( list("Cluster Size", cex = 2.5, col = 'forestgreen'))
```




```
fviz_cluster(k_result, data = scaled_data, geom = "point",

             main = 'Cluster Formation')
```


Cluster Formation
2.5
0.0
-2.5
-5.0
Dim2 (10.6%)
-5
0
5
Dim1 (28.5%)
cluster
1
2
3
4

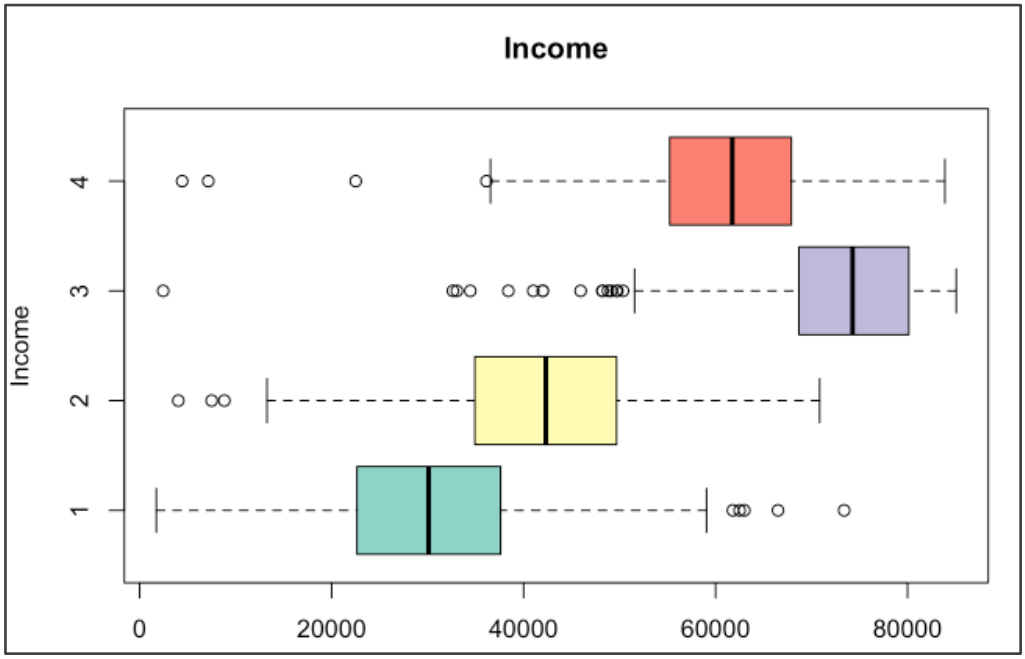
```
par(mar = c(5,4,4,4))
plot(data_copy$Income,
     data_copy$Spent,
     col = factor(data_copy$cluster_label), pch = 16,
     main = 'Spent vs Income', ylab = 'Spent', xlab = 'Income')
legend('topleft', legend = levels(factor(data_copy$cluster_label)),
       pch = 16, col =
         factor(levels(factor(data_copy$cluster_label))),
       pt.cex =1.2, cex= 0.75)
```



```
pers_quant_cont <-  c("Income",  "Recency", "Age" )

for (var in pers_quant_cont){

 cat('\n                    ',toupper(var), '\n\n')  agg <- aggregate(data_copy[,var], by =
list(data_copy$cluster_label), mean)

  names(agg) <- c('cluster', paste(var, 'mean', sep = '_'))

  agg['std'] <- aggregate(data_copy[,var], by = list(data_copy$cluster_label),
sd)[2]

  agg['data_range'] <- aggregate(data_copy[var], by = list(data_copy$cluster_label),

                        range)[2]print(agg)

  boxplot(data_copy[,var] ~data_copy$cluster_label, main = var,

          col = brewer.pal(4, 'Set3'), horizontal = T,

          xlab = 'Cluster label', ylab = var)

}
```

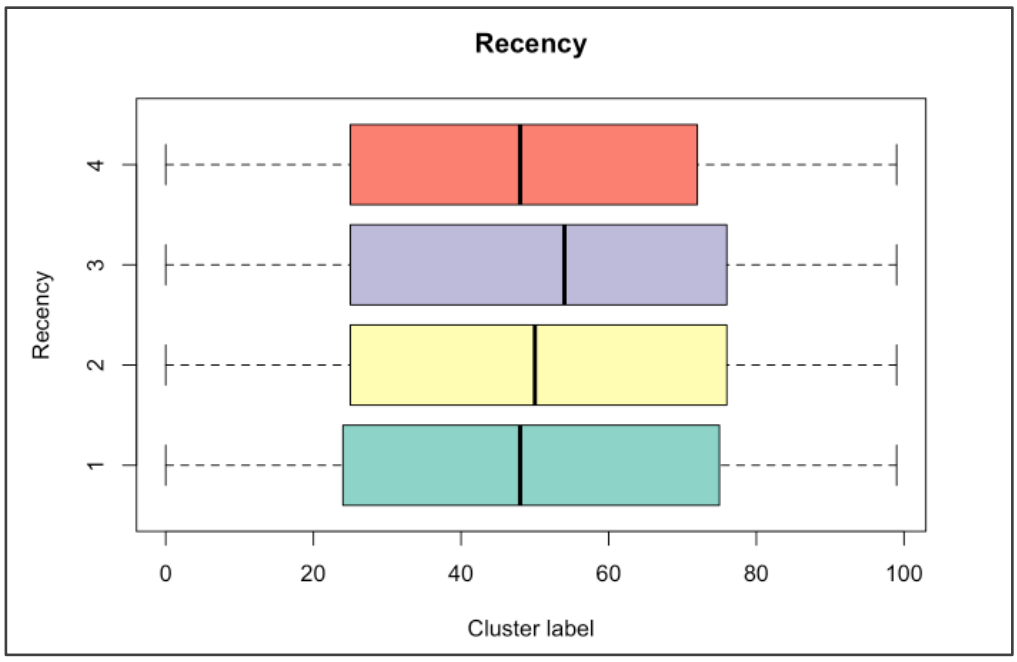
INCOME

	cluster	Income_mean	std	data_range.1	data_range.2
1	1	30215.22	11052.36	1730	73395
2	2	42083.92	11230.75	4023	70844
3	3	72830.28	10025.81	2447	85072
4	4	61257.23	10366.64	4428	83891



RECENCY

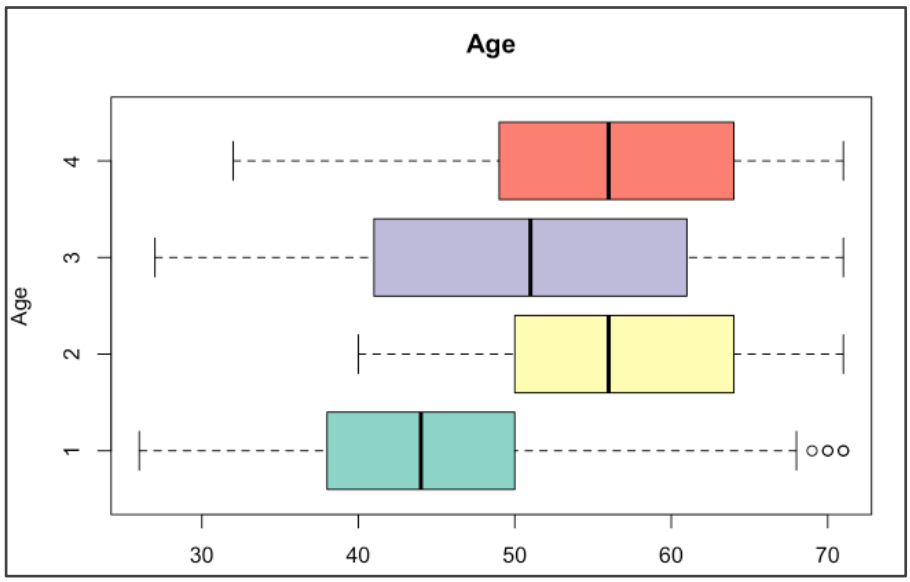
	cluster	Recency_mean	std	data_range.1	data_range.2
1	1	48.97351	28.72040	0	99
2	2	49.59197	29.38430	0	99
3	3	50.54743	29.68665	0	99
4	4	48.37638	28.65904	0	99



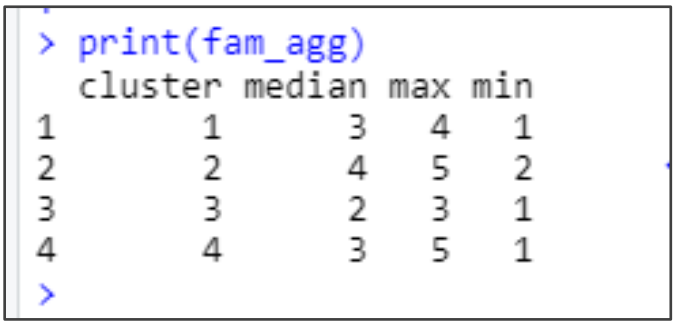
AGE

	cluster	Age_mean	std	data_range.1	data_range.2
1	1	44.63079	8.439686	26	71
2	2	56.68499	8.171113	40	71
3	3	51.08130	11.982501	27	71
4	4	56.01661	8.881938	32	71

>

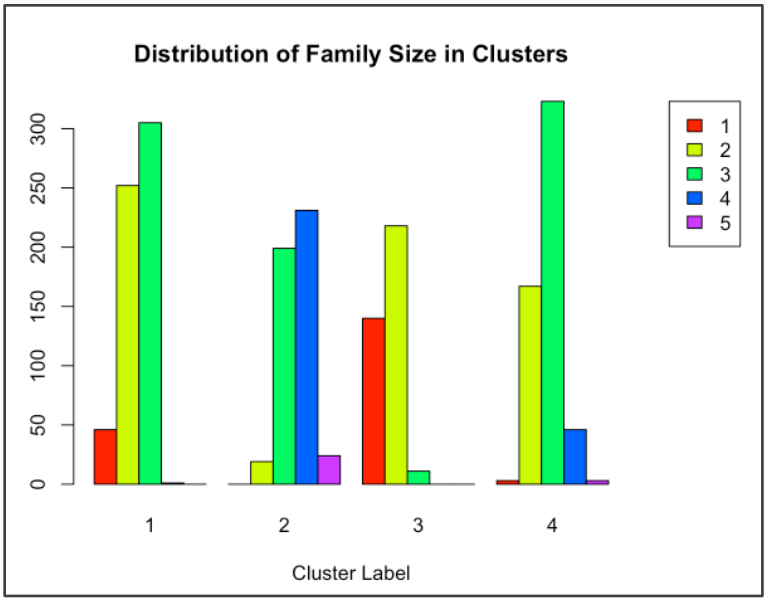


```
fam_agg <- aggregate(data_copy["Family_Size"],
                     by = list(data_copy$cluster_label), median)
names(fam_agg) <- c('cluster', 'median')
fam_agg$max <- aggregate(data_copy["Family_Size"],
                         by = list(data_copy$cluster_label), max)[,2]
fam_agg$min <- aggregate(data_copy["Family_Size"],
                         by = list(data_copy$cluster_label), min)[,2]
print(fam_agg)
```

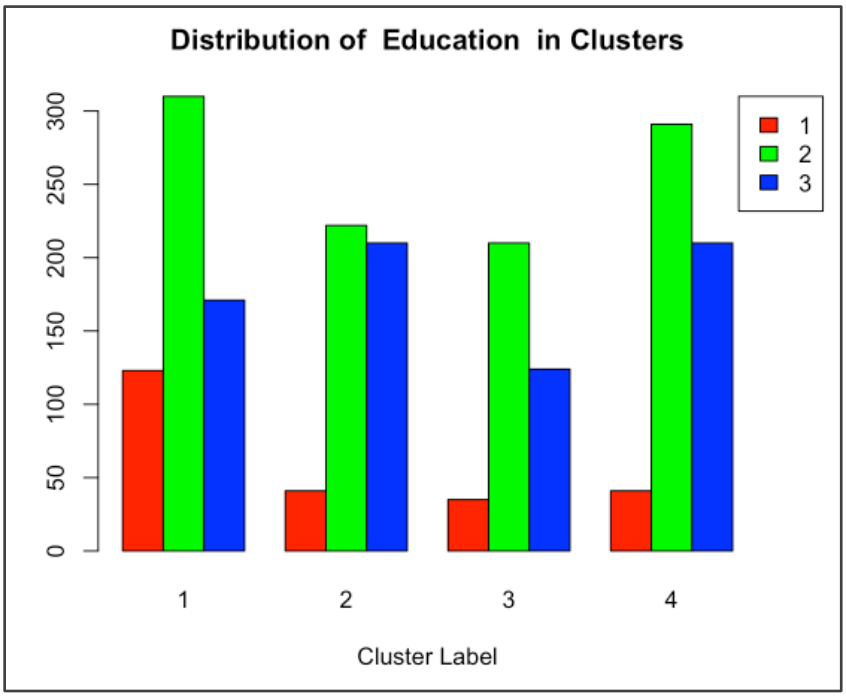


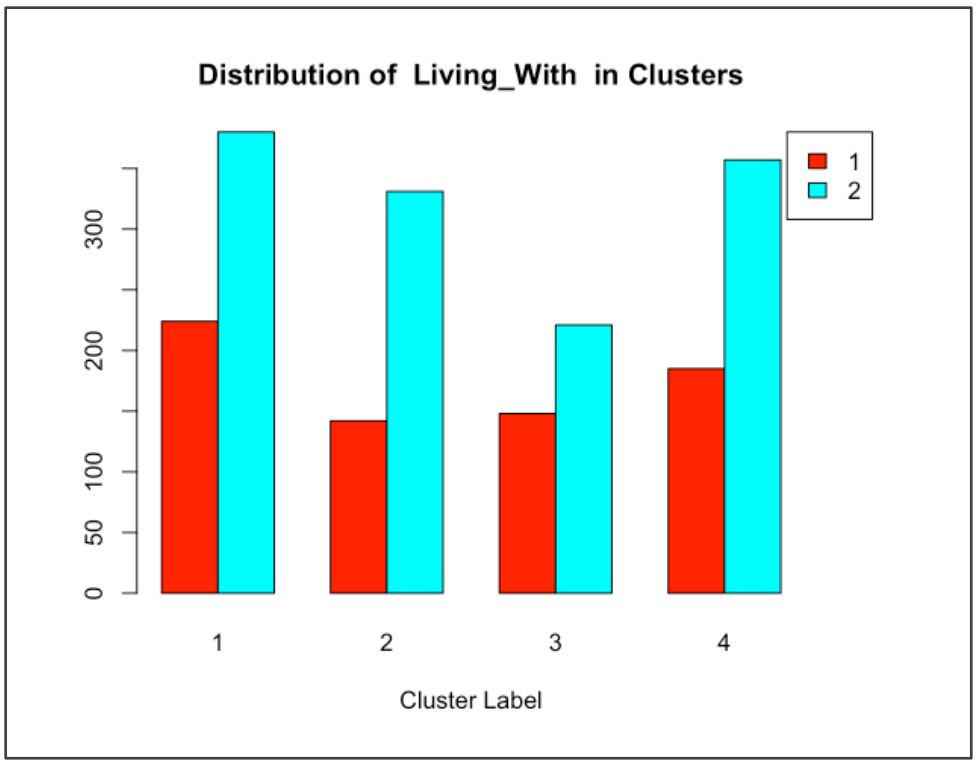
```
> print(fam_agg)
  cluster median max min
1       1      3   4   1
2       2      4   5   2
3       3      2   3   1
4       4      3   5   1
>
```

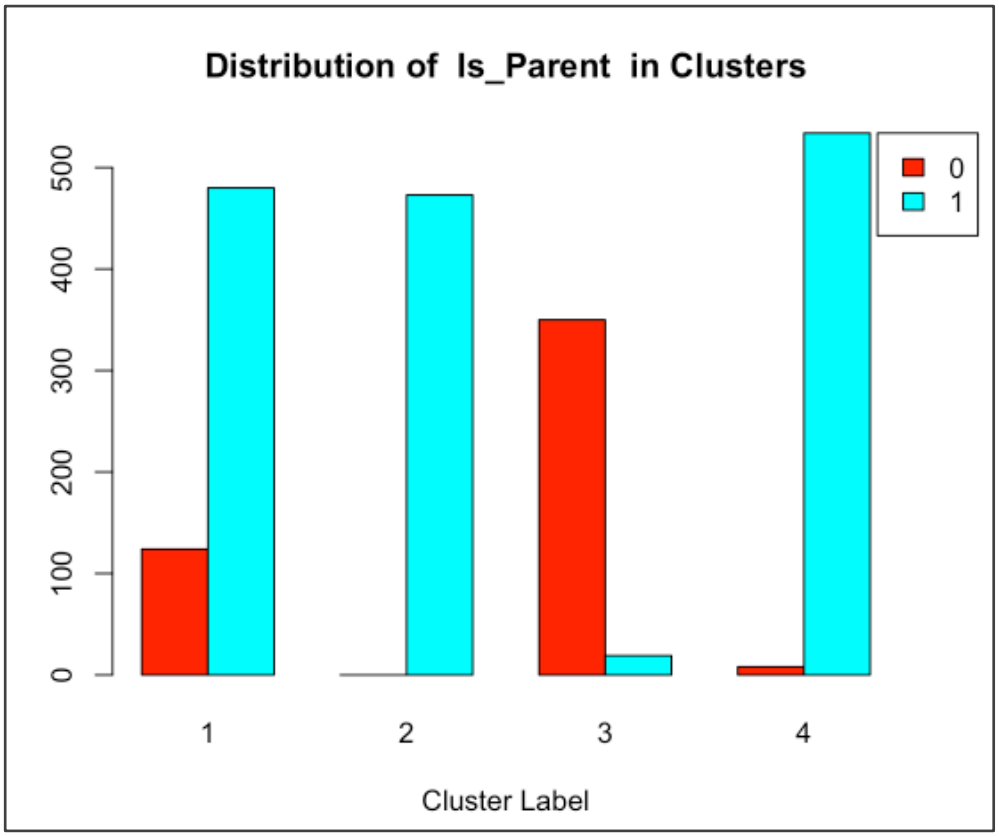
```
cnt <- table(data_copy[,"Family_Size"], data_copy$cluster_label)

par(mar = c(5,4,4,8))

barplot(cnt, beside = TRUE, col =
rainbow(length(unique(data_copy[,'Family_Size']))),

        main = paste('Distribution of Family Size in Clusters'),

        xlab = 'Cluster Label',

        legend = factor(levels(factor(data_copy[,'Family_Size']))),

        args.legend = list(x = 'topright', inset = c(-0.2,0)))
```



```
pers_cat <- c("Education", "Living_With","Is_Parent" par(mar = c(5,4,4,8)) for (var
in pers_cat){n <- length(unique(data_copy[,var]))  cnt <- table(data_copy[, var],
data_copy$cluster_label)  barplot(cnt, beside = TRUE, col = rainbow(n),

        main = paste('Distribution of ',var,' in Clusters'),

        xlab = 'Cluster Label',     legend = factor(levels(factor(data_copy[,var]))),
args.legend = list(x = 'topright', inset = c(-0.1, 0))) }

  }
```




Distribution of Living_With in Clusters
0
50
100
200
300
1
2
3
4
Cluster Label
1
2


Distribution of Is_Parent in Clusters
0
100
200
300
400
500
1
2
3
4
Cluster Label
0
1

```
product_cols <- c("Wines","Fruits","MeatProducts","FishProducts" ,
                  "SweetProducts", "GoldProds")
```

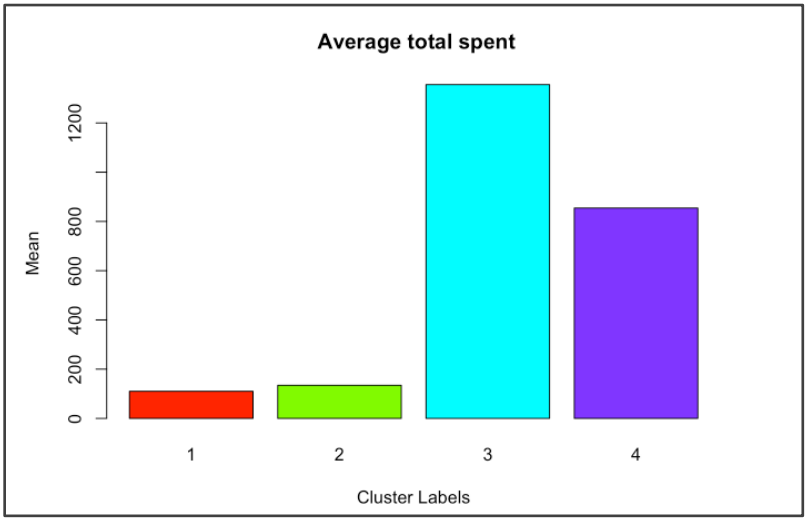
```
for (var in product_cols){
  agg <- aggregate(data_copy[,var], list(data_copy$cluster_label), mean)
  names(agg) <- c('Cluster_Label', paste(var,'_mean'))
  agg[paste(var,'_std')] <- aggregate(data_copy[,var],
                                      list(data_copy$cluster_label), sd)[2]
  print(agg)
}
```

```
##   Cluster_Label Wines _mean Wines _std
## 1             1    38.66556   60.08648
## 2             2    77.16702   82.13458
## 3             3   595.12195  318.37714
## 4             4   504.69742  276.25813
##   Cluster_Label Fruits _mean Fruits _std
## 1             1     6.773179    9.765911
## 2             2     3.940803    7.113240
## 3             3    64.726287   49.190550
## 4             4    34.046125   38.634355
##   Cluster_Label MeatProducts _mean MeatProducts _std
## 1             1           28.10596          33.12800
## 2             2           28.09091          29.26917
## 3             3          453.76694         224.09012
## 4             4          166.40590         110.17988
##   Cluster_Label FishProducts _mean FishProducts _std
## 1             1          10.407285         16.927012
## 2             2           5.615222          9.407666
## 3             3          98.005420         67.569449
## 4             4          44.809963         50.641639
##   Cluster_Label SweetProducts _mean SweetProducts _std
## 1             1            6.976821           9.628047
## 2             2            3.909091           6.477976
## 3             3           64.734417          50.262841
## 4             4           35.167897          40.885163
##   Cluster_Label GoldProds _mean GoldProds _std
## 1             1        19.28808       26.42722
## 2             2        15.69767       18.56697
```

```
agg <- aggregate(data_copy[,'Spent'], list(data_copy$cluster_label), mean)

names(agg) <- c('Cluster_Label', 'mean')

agg['std'] <- aggregate(data_copy[,'Spent'],

                        list(data_copy$cluster_label), sd)[2]

print(agg)
```

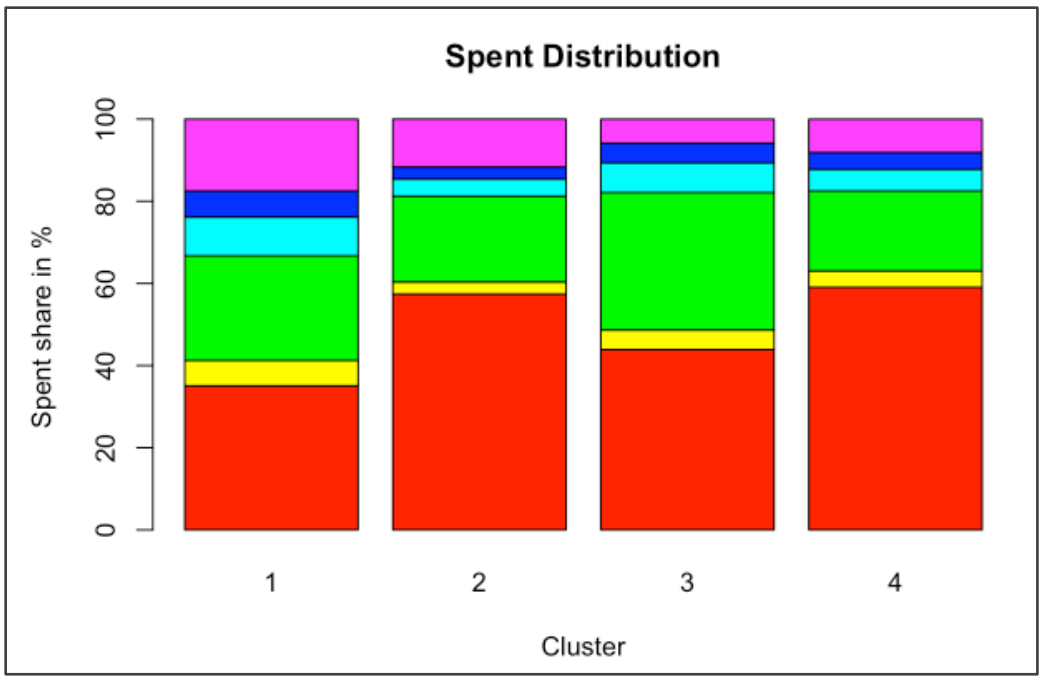
```
> agg <- aggregate(data_copy['Spent'], list(data_copy$cluster_label), mean)
> names(agg) <- c('Cluster_Label', 'mean')
> agg['std'] <- aggregate(data_copy['Spent'],
+                         list(data_copy$cluster_label), sd)[2]
> print(agg)
  Cluster_Label      mean      std
1             1  110.2169 115.0549
2             2  134.4207 119.0464
3             3 1355.8455 408.2443
4             4  854.4815 350.8762
>
```

```
par(mfrow = c(1,1))

barplot(agg$mean~agg$Cluster_Label, col = rainbow(4),

        xlab = 'Cluster Labels', ylab = 'Mean',

        main = ' Average total spent'

)
```



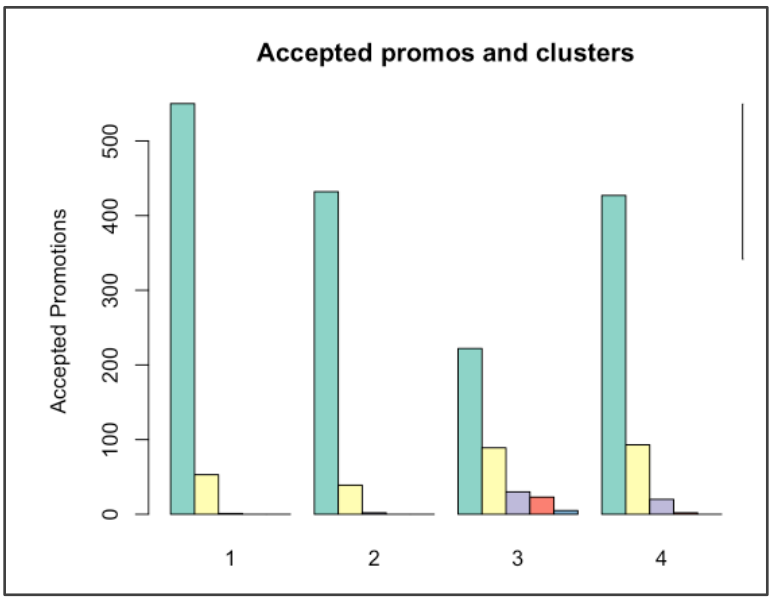
```
agg <- data.frame('Cluster_Label' = sort(unique(data_copy$cluster_label)))
for (var in product_cols){
  agg[var] <- aggregate(data_copy[,var], list(data_copy$cluster_label), sum)[2]
}
```

```
agg <- t(agg)
par(mar = c( 8, 4, 4, 4))
plot_data <- round(t(t(agg[-1,])/ colSums(agg[-1,]) * 100), 2)
colnames(plot_data) <- agg['Cluster_Label',]
barplot(plot_data, col = rainbow(6), beside = F, xlab = 'Cluster',
        ylab = 'Spent share in %' ,
        main = 'Spent Distribution', legend = row.names(plot_data),
        args.legend = list(x = 'bottom', inset = c(-0, -0.8), horiz = T ,  cex = 0.65))
```




```
data_copy$total_promos <- apply(data_copy[grep('Accepted', names(data_copy))],1, sum)
```

```
cnt <- table(data_copy$total_promos , data_copy$cluster_label)
par(mar = c(5,4,4,8))
barplot(cnt, beside = TRUE, ylab = 'Accepted Promotions',
        main = 'Accepted promos and clusters',
        col= brewer.pal(5, 'Set3'))
legend('topright', legend = levels(factor(data_copy$total_promos)),
       pch = 16, col = brewer.pal(5, 'Set3'),
       pt.cex =1.2, cex= 1, inset = c(-0.1,0))
```

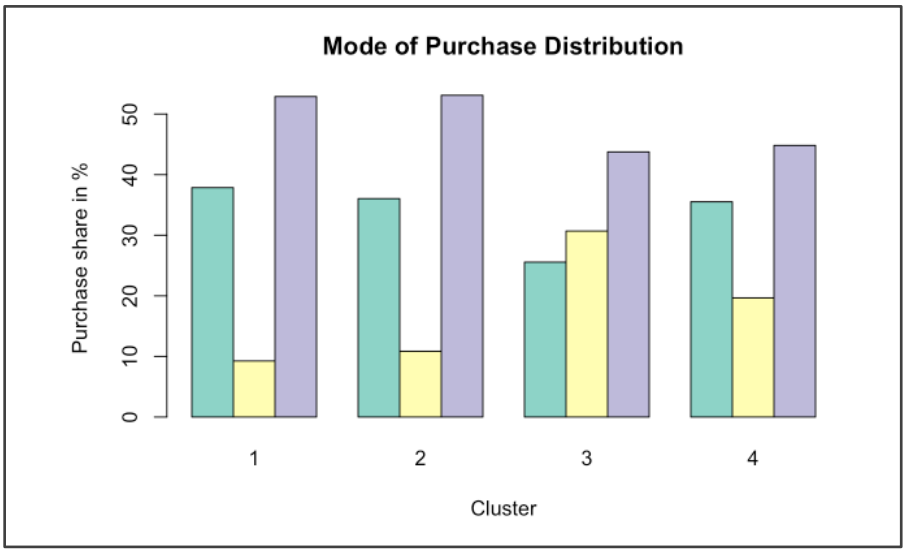


```
mode_of_purchase <- c("NumWebPurchases", "NumCatalogPurchases",
                      "NumStorePurchases")
```

```
agg <- data.frame(Cluster_Label = sort(unique(data_copy$cluster_label)))
for (var in mode_of_purchase){
  temp <- aggregate(data_copy[,var], list(data_copy$cluster_label), sum)
  temp <-  temp[order(temp$Group.1 ),]
  var_name <- sub('Num', '', var)
  agg[var_name] <- temp[,2]
}
print(agg)
```

```
> print(agg)
  Cluster_Label WebPurchases CatalogPurchases StorePurchases
1             1         1395               341           1949
2             2         1179               355           1737
3             3         1789              2148           3062
4             4         3572              1977           4507
>
```

```
agg <- t(agg)

par(mar = c( 8, 4, 4, 4))

plot_data <- round(t(t(agg[-1,])/ colSums(agg[-1,]) * 100), 2)

colnames(plot_data) <- agg['Cluster_Label',]

par(mar = c( 8, 4, 4, 4))

barplot(plot_data, col = brewer.pal(nrow(plot_data), 'Set3'), beside = T,

        xlab = 'Cluster', ylab = 'Purchase share in %' ,

        main = 'Mode of Purchase Distribution',

        legend = row.names(plot_data),

 args.legend = list(x = 'bottom',

                    inset = c(0, -.75), horiz = T, cex = 0.85))
```


Mode of Purchase Distribution
Purchase share in %
0
10
20
30
40
50
1
2
3
4
Cluster