UCD Michael Smurfit Graduate Business School

MIS40970 : Data Mining for Business Analytics

Assignment 1

Author: Shruti Goyal (16200726)

Professor: Prof. Michael O'Neill

February 15, 2017

Assignment Solution

The objective of this assignment is to explore the given data set called 'orange juice' and to perform some specified functions.

1. **Load the oj.csv data into R.**

   There are various methods to load data from a csv into R. For loading data into R, first, other objects has to be removed from the specified environment. Then a working directory needs to be set to read the file from the specified path.

   a) To load data into R from csv, we can use read.table(). Header = TRUE is set because the excel consists of header variables for each column.

   ```
   > rm(list=ls())
   > getwd()
   > setwd("G:/Smurfit/Course Materials/Data Mining/Assignments Week 4")
   > oj <- read.table("oj.csv",header=TRUE,sep=",")
   ```

   b) We can also use read.csv(). Default separator for this is comma (,)

   ```
   > rm(list=ls())
   > getwd()
   > setwd("G:/Smurfit/Course Materials/Data Mining/Assignments Week 4")
   > oj2 <- read.csv("oj.csv",header=TRUE)
   ```

   c) There is another function called read.csv2(), it used when the separator is semi colon(;). This function will not produced the desired result because the file is comma separated not colon separated.

   ```
   > rm(list=ls())
   > getwd()
   > setwd("G:/Smurfit/Course Materials/Data Mining/Assignments Week 4")
   > oj3 <- read.csv2("oj.csv",header=TRUE)
   ```

   First two command will generate the same output but the third command is generating different output. There is a difference between no of variables read from the file.

   Difference in output can be seen below:

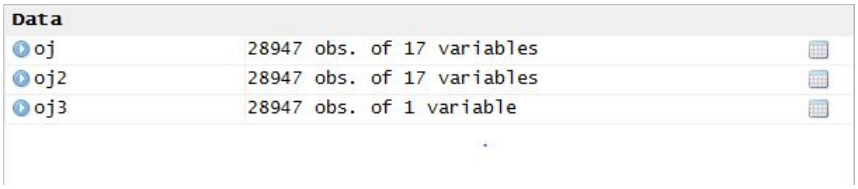

Figure 1: Output from different commands

2. **How many records and how many attributes are in the orange juice dataset?**

   a) To calculate the total number of records in the orange juice dataset, dim('nameofdataset') has to be used. So, for orange juice dataset **dim(oj)** has to be used. It will return total number of rows and number of columns.

   ```
   > rm(list=ls())
   > getwd()
   > setwd("G:/Smurfit/Course Materials/Data Mining/Assignments Week 4")
   > oj <- read.table("oj.csv",header=TRUE,sep=",")
   > dim(oj)
   ```

Output :

```
> dim(oj)
[1] 28947 17
```

Also, we can calculate total number of records by using nrow() command as well whereas NROW() will generate the same result which will treat a vector as a column matrix.

```
> rm(list=ls())
> getwd()
> setwd("G:/Smurfit/Course Materials/Data Mining/Assignments Week 4")
> oj <- read.table("oj.csv",header=TRUE,sep=",")
> nrow(oj)
```

Output :

```
> nrow(oj)
[1] 28947
```

b) To generate attributes of the dataset, attributes('nameofdataset') has to be used. It will remove all attributes, then sets any of the dim attribute and then all other remaining attributes.

```
> rm(list=ls())
> getwd()
> setwd("G:/Smurfit/Course Materials/Data Mining/Assignments Week 4")
> oj <- read.table("oj.csv",header=TRUE,sep=",")
> attributes(oj)
```

Output :

```
$names
 [1] "store"    "brand"    "week"     "logmove"  "feat"     "price"    "AGE60"
 [8] "EDUC"     "ETHNIC"   "INCOME"   "HHLARGE"  "WORKWOM"  "HVAL150"  "SSTRDIST"
[15] "SSTRVOL"  "CPDIST5"  "CPWVOL5"

$class
[1] "tbl_df"     "tbl"        "data.frame"
```

3. **What is the mean, standard deviation and range of the price of orange juice?**

There are two ways in which we can generated the mean of a dataset.

a) We can use standalone commands such as mean('variablename'), sd('variable name') and range('variablename').

```
> rm(list=ls())
> getwd()
> setwd("G:/Smurfit/Course Materials/Data Mining/Assignments Week 4")
> oj <- read.table("oj.csv",header=TRUE,sep=",")
> mean(oj$price)
> sd(oj$price)
> range(oj$price)
```

Output :

```
> mean(oj$price)
[1] 2.282488
> sd(oj$price)
[1] 0.6480007
> range(oj$price)
[1] 0.52 3.87
```

b) By using summary('nameofdataset') we can generate statistical summary of the dataset, it will result in statistics of each variable in the dataset.

```
> rm(list=ls())
> getwd()
> setwd("G:/Smurfit/Course Materials/Data Mining/Assignments Week 4")
> oj <- read.table("oj.csv",header=TRUE,sep=",")
> summary(oj)
```

Output :

```
     store          brand                week          logmove           feat
 Min.   :  2.00   Length:28947       Min.   : 40.0   Min.   : 4.159   Min.   :0.0000
 1st Qu.: 53.00   Class :character   1st Qu.: 70.0   1st Qu.: 8.490   1st Qu.:0.0000
 Median : 86.00   Mode  :character   Median :101.0   Median : 9.034   Median :0.0000
 Mean   : 80.88                      Mean   :100.5   Mean   : 9.168   Mean   :0.2373
 3rd Qu.:111.00                      3rd Qu.:130.0   3rd Qu.: 9.765   3rd Qu.:0.0000
 Max.   :137.00                      Max.   :160.0   Max.   :13.482   Max.   :1.0000
     price            AGE60              EDUC             ETHNIC            INCOME
 Min.   :0.520   Min.   :0.05805   Min.   :0.04955   Min.   :0.02425   Min.   : 9.867
 1st Qu.:1.790   1st Qu.:0.12210   1st Qu.:0.14598   1st Qu.:0.04191   1st Qu.:10.456
 Median :2.170   Median :0.17065   Median :0.22939   Median :0.07466   Median :10.635
 Mean   :2.282   Mean   :0.17313   Mean   :0.22522   Mean   :0.15556   Mean   :10.617
 3rd Qu.:2.730   3rd Qu.:0.21395   3rd Qu.:0.28439   3rd Qu.:0.18776   3rd Qu.:10.797
 Max.   :3.870   Max.   :0.30740   Max.   :0.52836   Max.   :0.99569   Max.   :11.236
    HHLARGE           WORKWOM          HVAL150           SSTRDIST          SSTRVOL
 Min.   :0.01351   Min.   :0.2445   Min.   :0.002509   Min.   : 0.1321   Min.   :0.4000
 1st Qu.:0.09794   1st Qu.:0.3126   1st Qu.:0.123486   1st Qu.: 2.7670   1st Qu.:0.7273
 Median :0.11122   Median :0.3556   Median :0.346154   Median : 4.6507   Median :1.1154
 Mean   :0.11560   Mean   :0.3592   Mean   :0.343766   Mean   : 5.0973   Mean   :1.2073
 3rd Qu.:0.13517   3rd Qu.:0.4023   3rd Qu.:0.528313   3rd Qu.: 6.6506   3rd Qu.:1.5385
 Max.   :0.21635   Max.   :0.4723   Max.   :0.916700   Max.   :17.8560   Max.   :2.5714
    CPDIST5          CPWVOL5
 Min.   :0.7725   Min.   :0.09456
 1st Qu.:1.6262   1st Qu.:0.27167
 Median :1.9634   Median :0.38323
 Mean   :2.1204   Mean   :0.43891
 3rd Qu.:2.5337   3rd Qu.:0.56024
 Max.   :4.1079   Max.   :1.14337
```

4. **What is the median of the log of number of units sold (logmove)?**

   median('variablename') will generate the desired median for 'log of the number of units sold' which is **logmove** in the dataset.

```
> rm(list=ls())
> getwd()
> setwd("G:/Smurfit/Course Materials/Data Mining/Assignments Week 4")
> oj <- read.table("oj.csv",header=TRUE,sep=",")
> median(oj$logmove)
```

Output :

```
> median(oj$logmove)
[1] 9.03408
```

5. **What are the names of the 3 orange juice brands?**

   So, factor('variablename') command has to be used to find out the three categories of orange juice dataset.

```
> rm(list=ls())
> getwd()
> setwd("G:/Smurfit/Course Materials/Data Mining/Assignments Week 4")
> oj <- read.table("oj.csv",header=TRUE,sep=",")
> brands <- factor(oj$brand)
> table(brands)
```

Output :

```
brands
  dominicks minute.maid   tropicana
       9649        9649        9649
```

6. **Create a histogram of prices for each brand of orange juice.**

   There are three categories of orange juice brand i.e. dominicks, minute maid and tropicana. For each category a histogram for price has to be generated. Three separate graphs will be generated.

   a) For brand dominicks

```
> rm(list=ls())
> getwd()
> setwd("G:/Smurfit/Course Materials/Data Mining/Assignments Week 4")
> oj <- read.table("oj.csv",header=TRUE,sep=",")
> brands <- factor(oj$brand)
> hist(oj$price[oj$brand=='domincks'],col=(c("red")),xlab="Price",
  ylab="Frequency",main="Histogram representing price for dominicks")
```

Output :

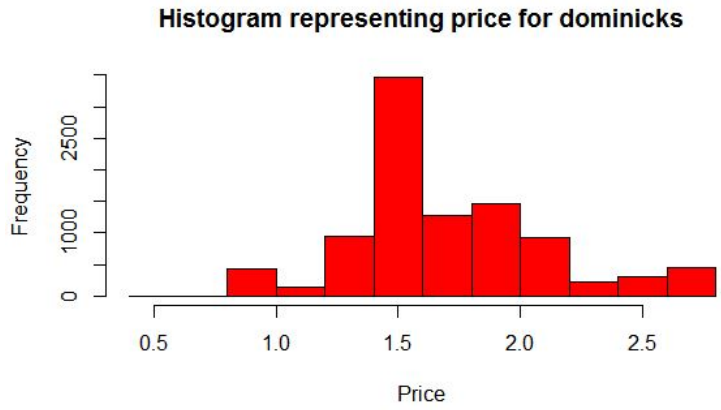

Figure 2: Histogram of price for brand 'dominicks'

   b) For brand minute maid

```
> rm(list=ls())
> getwd()
> setwd("G:/Smurfit/Course Materials/Data Mining/Assignments Week 4")
> oj <- read.table("oj.csv",header=TRUE,sep=",")
> brands <- factor(oj$brand)
> hist(oj$price[oj$brand=='minute.maid'],col=(c("red")),xlab="Price",
  ylab="Frequency",main="Histogram representing price for minute maid")
```

Output :

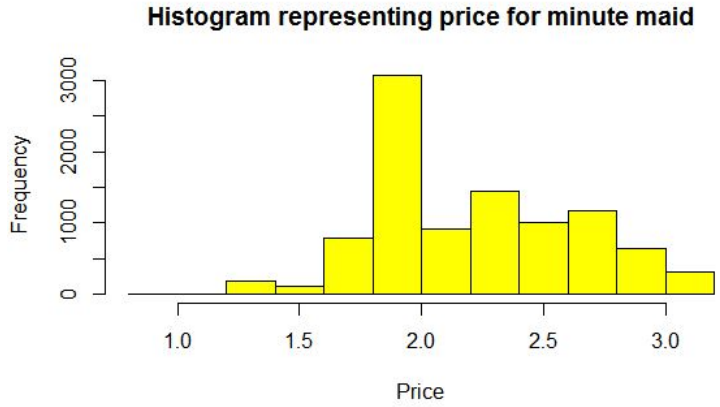

Figure 3: Histogram of price for brand 'minute maid'

c) For brand tropicana

```
> rm(list=ls())
> getwd()
> setwd("G:/Smurfit/Course Materials/Data Mining/Assignments Week 4")
> oj <- read.table("oj.csv",header=TRUE,sep=",")
> brands <- factor(oj$brand)
> hist(oj$price[oj$brand=='tropicana'],col=(c("red")),xlab="Price",
  ylab="Frequency",main="Histogram representing price for tropicana")
```

Output :

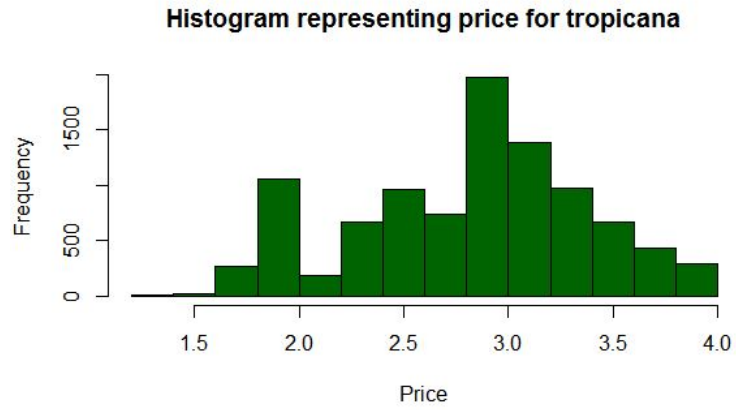

Figure 4: Histogram of price for brand 'tropicana'

d) To generate the histogram for price variable for each brand can also be done using the same histogram. Use of **ggplot** can fulfil the purpose. For that we need to install the ggplot2 package and then histogram can be generated.

```
> rm(list=ls())
> getwd()
> setwd("G:/Smurfit/Course Materials/Data Mining/Assignments Week 4")
> oj <- read.table("oj.csv",header=TRUE,sep=",")
> brands <- factor(oj$brand)
> install.packages("ggplot2")
> library(ggplot2)
> ggplot(oj, aes(x= oj$price, fill = brands)) + geom_histogram(binwidth=1,
  position = "dodge")
```

Output :

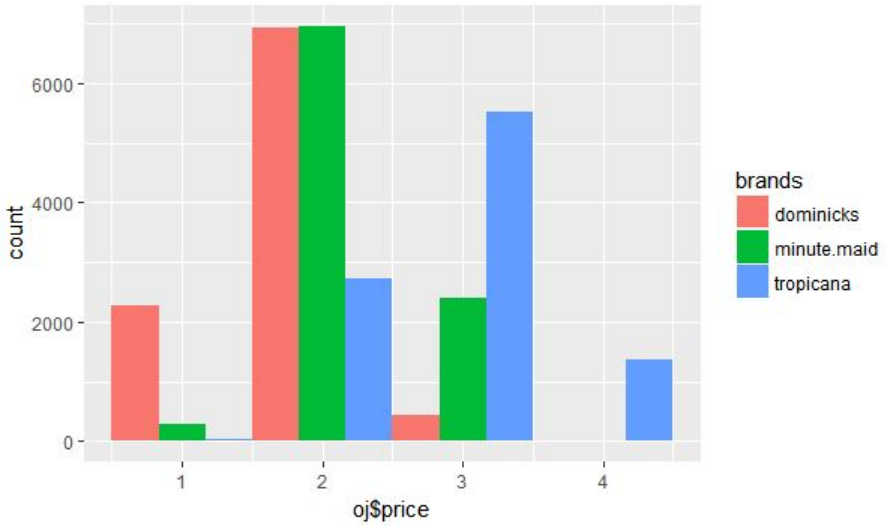

Figure 5: Histogram representing price for each brand

7. **Generate a boxplot, which includes a separate plot for the prices of each brand**

To generate separate boxplots for the prices of each brand. There are three brands that are dominicks, minute maid and tropicana. using boxplot command will do the job.

```
> rm(list=ls())
> getwd()
> setwd("G:/Smurfit/Course Materials/Data Mining/Assignments Week 4")
> oj <- read.table("oj.csv",header=TRUE,sep=",")
> brands <- factor(oj$brand)
> boxplot(price~brand, data=oj, col=(c("green","gold","pink")),
  main="Boxplot",xlab="Price and brand")
```

Output :

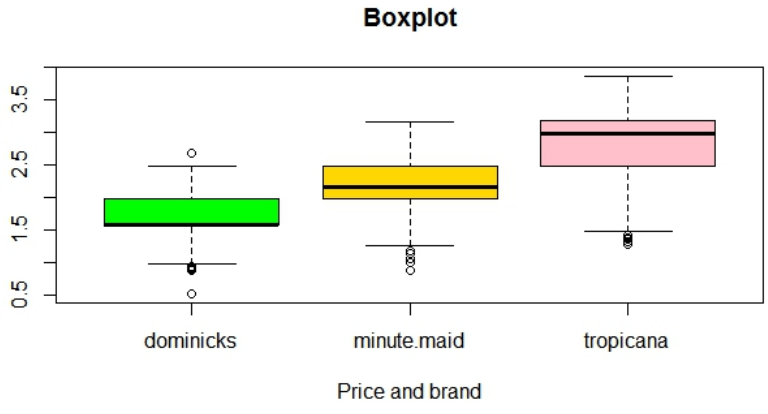

Figure 6: Boxplot representing price for each brand

8. **What does the boxplot tell us about the relative prices of each brand?**

The boxplot diagrams in Q7 shows significant difference in patterns for each type of brands. Boxplot for Dominicks tells us that majority of the price range is more than the

median price (which is 1.6 approx) that varies from 1.6 - 2. Also, there are few outliers as well on both the sides of the whiskers, which are 0.5,1 and 2.75. Boxplot for minute maid tells us that there is not much difference in price range, values lies around the median price (2.25 approx) and price varies from 2 - 2.5 but still there are predominant outliers around the lower whisker below price of 1.2 approx and outliers values are also have varied price values. Now boxplot for tropicana shows price falls below the median price(3), price varies from 2.5 - 3.2 and there is not much difference between the prices. There are also few outliers around the lower whisker below price of 1.5.

9. **Generate a scatterplot of the logmove compared to price, and color the points according to their brand**

   a) Creating plots using pairs() command

```
> rm(list=ls())
> getwd()
> setwd("G:/Smurfit/Course Materials/Data Mining/Assignments Week 4")
> oj <- read.table("oj.csv",header=TRUE,sep=",")

> brands <- factor(oj$brand)

> pairs(oj$logmove~oj$price,main="Plot for logmove against price for each brand",
  pch=21,col=rainbow(3)[unclass(brands)])
```

Output :

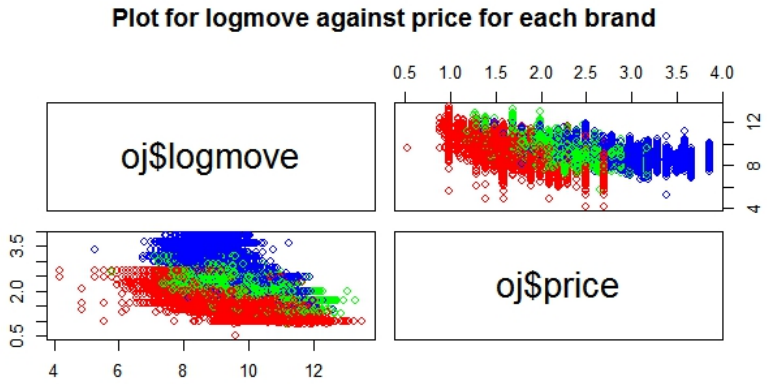

Figure 7: Scatterplot using pairs() representing logmove against price for each brand

Red color for dominicks, dark green for minute maid and gold for tropicana

   b) Creating plots using plot() command

```
> setwd("G:/Smurfit/Course Materials/Data Mining/Assignments Week 4")
> oj <- read.table("oj.csv",header=TRUE,sep=",")
> brands <- factor(oj$brand)

> plot(oj$logmove,oj$price,type="p",main="Plot for logmove against price for
  each brand",xlab="Price",ylab="logmove",pch=21,col=rainbow(3)[unclass(brands)])

> legend("bottomleft",inset = .005, title = "Brands", c("Dominicks","Minute
  maid","Tropicana"),fill=rainbow(3),horiz=FALSE)
```

Output :

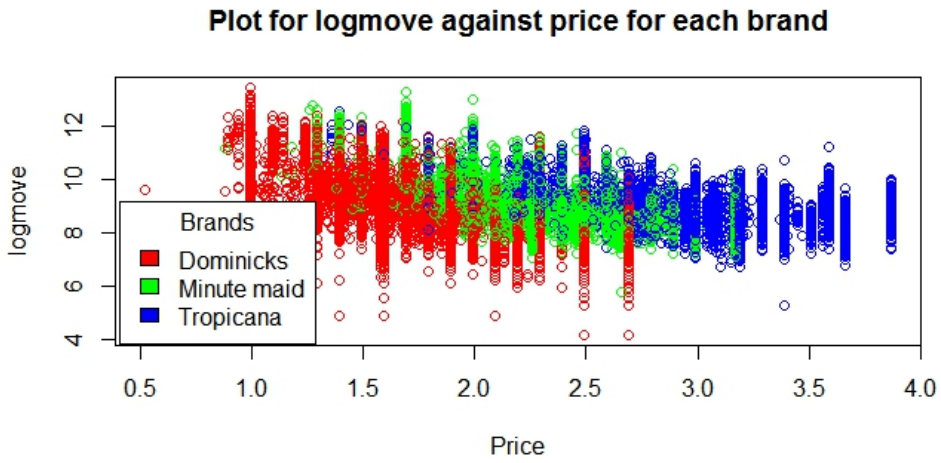

Figure 8: Scatterplot using plot() representing logmove against price for each brand

Red color for dominicks, green for minute maid and blue for tropicana

10. **Based on what you observe in the scatterplot, what can we say about the price of each brand of chilled orange juice and the volume of sales?**

    logmove = log of number of units sold
    price = price of brand

    Scatterplot shows the correlation pattern between price and number of units sold. Data is not evenly distributed into columns and rows. From scatterplot we can interpret information for continuous variables rather than discrete variables. We can interpret that maximum variation in number of units sold and price is for dominicks brand as price ranges from 0.5 - 2.6 with corresponding units sold between 4 - 12 units. For dominicks brand, when the price is lowest, number of units sold is highest. Minute maid brand has more similar pattern to dominicks, it just has little variations in the price range and number of units sold. Mostly, for all brands price varies in the range 1 - 3.5 against number of units sold in the range 8 - 11. From the graph we can say that, highest price value is for tropicana (4 approx)and average number of units sold is between 8 - 10 units. If we have to consider the average price range for all three brands together then we can say that for prices in range 9 - 10 then approx 2 - 2.5 units of items has been sold .

11. **Calculate the mean price of orange juice sold each week, and create a line plot of this timeseries.**

    We need to calculate the mean price using tapply() and store the values in a data object and then while using plot.ts() we can generate the time series plot.

```
> oj <- read.table("oj.csv",header=TRUE,sep=",")
> meanprice <- tapply(oj$price,oj$week,FUN=mean,na.rm=TRUE)

> plot.ts(meanprice,xaxt="n",pch=16,type="l",col="green",
  main="Time series graph between mean price and week",xlab = "Week",
  ylab="Mean Price")
```

Output :

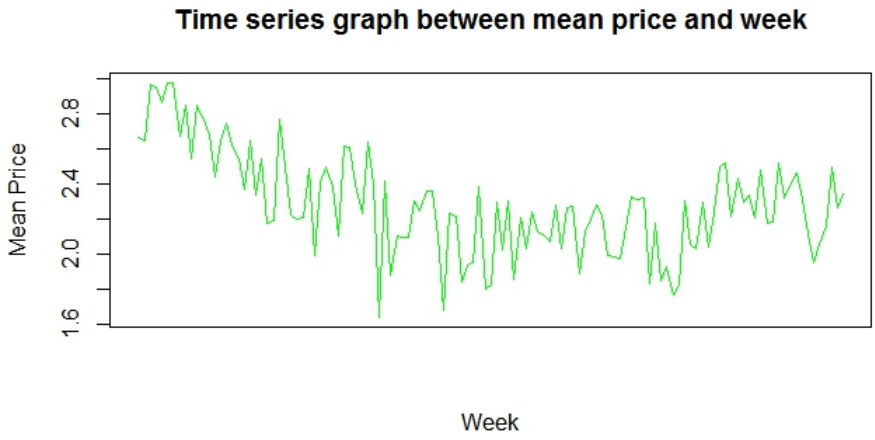

Figure 9: Time series plot for Mean Weekly Price

12. **Extract the mean weekly price of orange juice sold each week according to each brand**

    To calculate the mean weekly price of dataset for each week according to brand, tapply() is used but for indexing it for each brand, INDEX option must be used.

```
> rm(list=ls())
> getwd()
> setwd("G:/Smurfit/Course Materials/Data Mining/Assignments Week 4")
> oj <- read.table("oj.csv",header=TRUE,sep=",")
> brands <- factor(oj$brand)
> tapply(oj$price, INDEX=list(oj$week,brands),FUN=mean,na.rm=TRUE)
```

Output :

```
    dominicks minute.maid tropicana
40   1.590000    2.890000  3.509315
41   2.447015    1.990000  3.514627
42   2.461940    2.911343  3.536567
43   2.452597    2.892597  3.517792
44   2.463506    2.590000  3.534156
45   2.467838    2.908649  3.541486
46   2.471053    2.913158  3.547500
47   2.086800    2.390000  3.534000
48   2.086883    2.907013  3.540000
49   2.086842    1.990000  3.545000
50   2.083924    2.911392  3.539114
51   1.889000    2.921625  3.552375
52   1.888987    2.913797  3.243544
53   1.888987    2.192025  3.243544
54   1.790000    2.908765  3.245062
55   1.779610    2.909481  3.542857
56   1.790000    2.909114  3.169114
57   2.469000    1.990000  3.170375
58   1.202469    2.657160  3.250494
59   1.590000    2.912375  3.432375
60   1.590000    1.979367  3.431772
```

13. **Create a plot which compares the mean weekly price of orange juice for all brands versus each individual brand.**

To generate a plot to compare weekly price of of all brands verses each individual brand

We first need to use tapply() for all brands and for each brand (dominicks, minute maid, tropicana) to calculate the mean price on weekly basis and store the results into data objects. But these two data objects are of different lengths, so we need to store the data first into a data frame and then by using ggplot() we can create a line graph for each brand category and all brands.

```
> rm(list=ls())
> getwd()
> setwd("G:/Smurfit/Course Materials/Data Mining/Assignments Week 4")
> oj <- read.table("oj.csv",header=TRUE,sep=",")
> indbrand <- tapply(oj$price, INDEX=list(oj$week,oj$brand),FUN=mean,
  na.rm=TRUE)
> allbrands <- tapply(oj$price,oj$week,FUN=mean,na.rm=TRUE)
> frame <- cbind(allbrands,indbrand)
> write.table(df, file = 'brand.csv',sep = ",",row.names = T)
> library(readr)
> branddata <- read_csv("brand.csv",col_names = FALSE, skip = 1)
> ggplot(data = branddata, aes(branddata$X1,xlab="Week")) +
     geom_line(aes(y = branddata$X2, colour = "All Brands")) +
      geom_line(aes(y = branddata$X3, colour = "Dominicks")) +
     geom_line(aes(y = branddata$X4, colour = "Minute Maid")) +
     geom_line(aes(y = branddata$X5, colour = "Tropicana")) +
     xlab("Week") +ylab("Price")
```

Output :

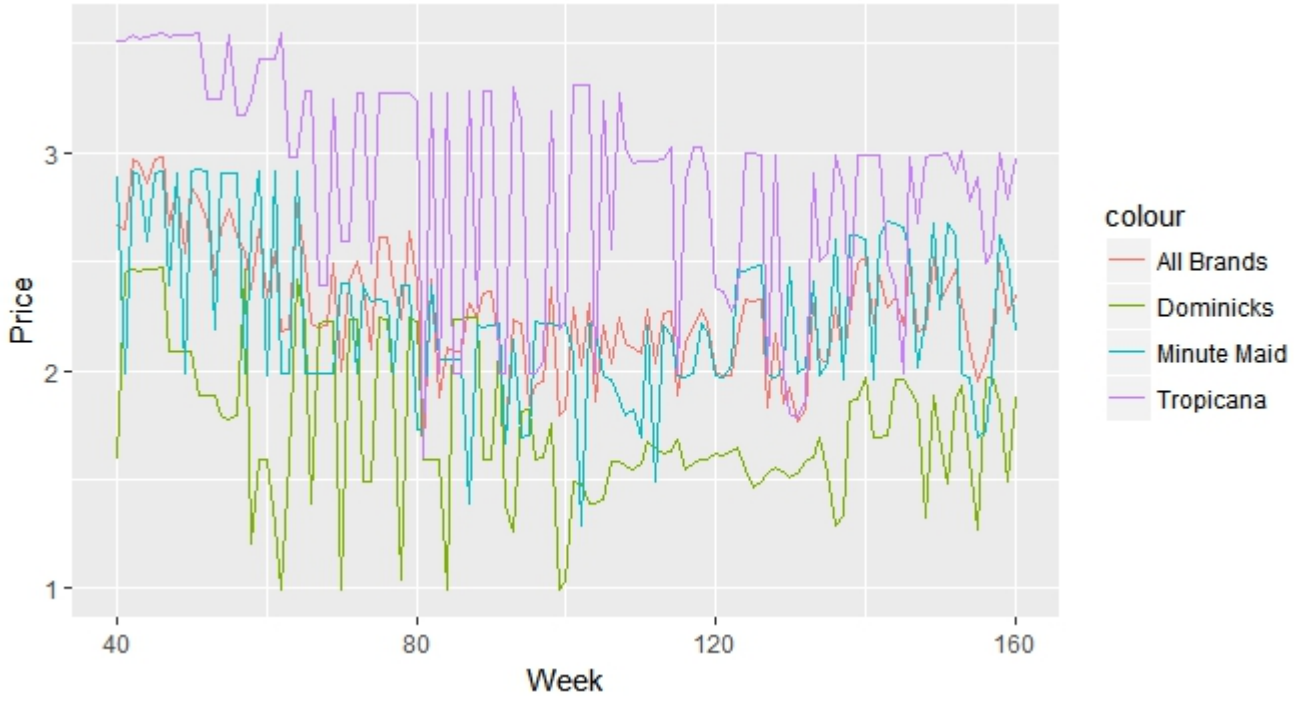

Figure 10: Line plot for mean weekly prices for all brands versus each brand

14. **When there is an advertising campaign for orange juice does it impact on the number of units sold?**

To determine impact when there is an advertising campaign for orange juice we have to use factor() command. factor(feat) will determine the levels of the orange juice advertisement which is either 1 or 0. Lastly, we can use tapply() to calculate the total number of units sold.

```
> rm(list=ls())
> getwd()
> setwd("G:/Smurfit/Course Materials/Data Mining/Assignments Week 4")
> oj <- read.table("oj.csv",header=TRUE,sep=",")
> adcap <- factor(oj$feat)
> tapply(oj$logmove,adcap,FUN=sum,na.rm=TRUE)
> #Affected sales by advertisement based on brands
> sales <- tapply(exp(oj$logmove), oj[,c("feat","brand")], sum)
> print(sales)
> mosaicplot(salestable,col=rainbow(3),
  main = "Affected sales on using advertisement",
  xlab = "Advertisement (0- without promotion, 1- With promotion)", ylab = "Brand")
```

Output :

```
> tapply(oj$logmove,adcap,FUN=sum,na.rm=TRUE)
       0        1
195685.4  69696.8

> print(sales)
    brand
feat dominicks minute.maid tropicana
   0  84120384    57478272  80762624
   1 107263296   118504640  53007424
```

From the output generated we can interpret that for level 0 (no advertisement) total number of units sold are more than level 1 (advertisement). So there was not much impact of the advertisement campaign on the number of units sold. Using the figure below we can analyse the affect on sales.

Output :

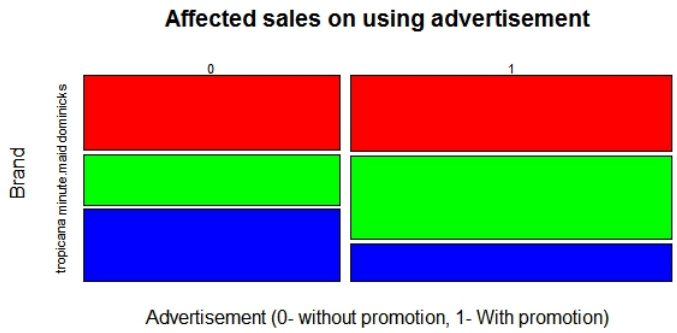

Figure 11: Affected sales using Advertisement

15. **Can you create a line plot of the mean weekly units sold without a promotion overlayed with the mean weekly units sold with a promotion? What is interesting about this plot?**

    To plot graph between mean weekly price without a promotion and with a promotion

    First we need to determine the mean weekly values for both level values of feat (i.e. 0 and 1). To do this, tapply() has to be use with list() function where both weeks and feat will be used.

```
> pdata <- tapply(oj$logmove,list(Week=oj$week,oj$feat),FUN=mean,na.rm=TRUE)
> df1 <- cbind(pdata)
> write.table(df1, file = 'Promotion.csv',sep = ",",row.names = T)
> library(readr)
> prom <- read_csv("G:/R_programs_git/R_Progams/Promotion.csv",
  col_names = FALSE, skip = 1)
> plot(prom$X1,prom$X2,type="l",col="red",
  main="Line plot between mean weekly price without promotion and with promotion",
  xlab = "Week", ylab = "Promotions")
> lines(prom$X1,prom$X3,col="green")
> legend("bottomright",c("Without Promotion","With Promotion"),
  fill= c("red","green"),horiz=FALSE)
```

Output :

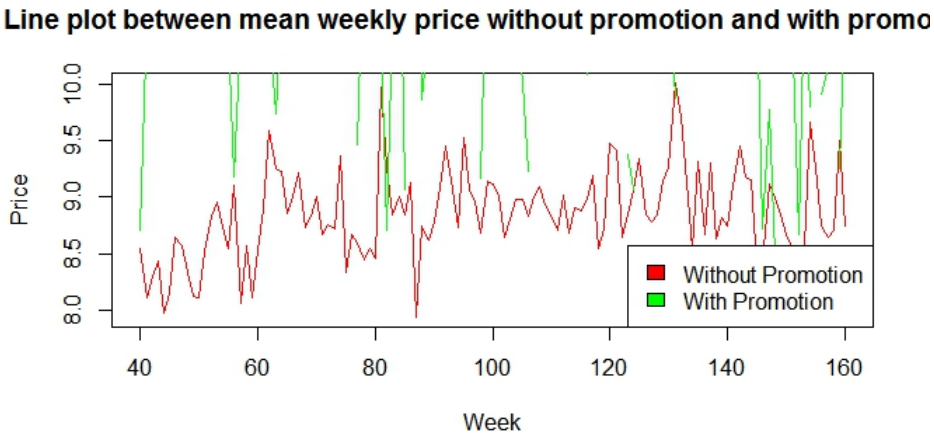

Figure 12: Mean weekly price between without promotion and with promotion

From the above graph, we can interpret that there are significant difference over the weeks between sales without promotion and sales with promotion. For example, during week 40-60, sales without promotion are continuously varying over different prices but for sales with promotion the values are more discrete. We cannot find a fixed pattern for sales with promotion. For some weeks, there is not any affect on sales for promotions.

16. **Consider the demographic and competitive variables. Using descriptive analytics are there patterns you can observe that might suggest the potential for profiling individual stores or customers, which might then be used for marketing purposes?**

Using descriptive statistics we describe and summarize the information from data. From the orange juice dataset we have 11 demographic and competitive variables such as AGE60, EDUC, ETHNIC, INCOME, etc.

```
> store <- tapply(oj$logmove, INDEX = list(oj$store,brands), FUN = sum,
  na.rm =TRUE)
> df2 <- cbind(store)
> write.table(df2, file = "store.csv",sep = ",",row.names = T)
> library(readr)
> sdata <- read_csv("G:/R_programs_git/R_Progams/store.csv",
  col_names = FALSE, skip = 1)
> ggplot(sdata, aes(sdata$X1)) +
  geom_point(aes(y = sdata$X2, color = "Dominicks")) +
  geom_point(aes(y = sdata$X3, color = "Minute Maid")) +
  geom_point(aes(y = sdata$X4, color = "Tropicana")) +
  xlab("Store") + ylab("Number of units sold")
```

From the generated output we can see that the data is more clustered on the right side of the graph. We can see that most of the sales has been occurred in the stores (store 75 - store 150). Marketing team can generate offers or discount coupons to increase the sales at all stores and focus more on other stores.

Output :

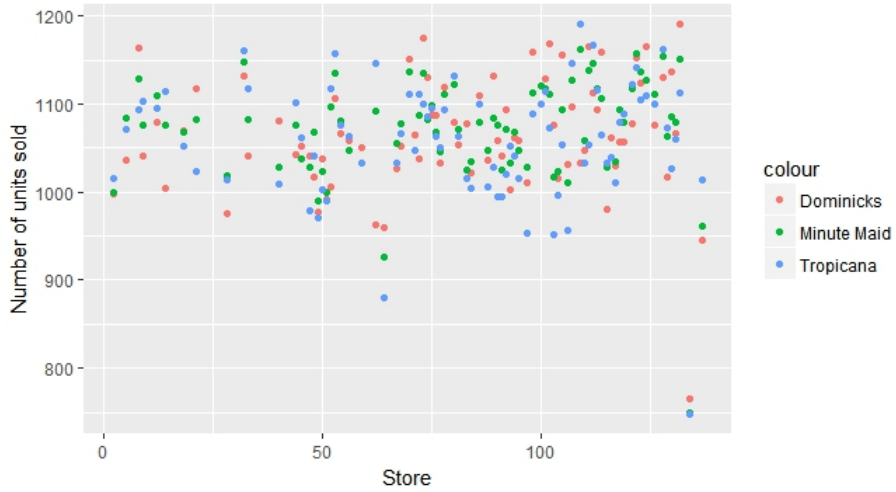

Figure 13: Trends for each brand between store and number of units sold

From summary table of descriptive analytics we can know the basic statistics such as mean, median, range, etc. We can generate the summary table by two ways as discussed below :

Method 1 :

```
> df <- unique(oj[c(7:17)])
  #Generating a list of subset of columns
> Des_stats <- do.call(data.frame,
                list(Average = apply(df, 2, mean),
                    Standard_Deviation = apply(df, 2, sd),
                     Median = apply(df, 2, median),
                     Minimum = apply(df, 2, min),
                     Maximum = apply(df, 2, max)))
> print(Des_stats)
```

Output :

```
            Average Standard_Deviation      Median    Minimum    Maximum
AGE60     0.1729724         0.06221013  0.17065481 0.05805397  0.3073979
EDUC      0.2257762         0.11114318  0.22939040 0.04955029  0.5283620
ETHNIC    0.1546351         0.18782770  0.07465643 0.02424657  0.9956908
INCOME   10.6176755         0.28344624 10.63532646 9.86708287 11.2361965
HHLARGE   0.1156640         0.03035444  0.11122120 0.01350636  0.2163543
WORKWOM   0.3591536         0.05280191  0.35563513 0.24446267  0.4723083
HVAL150   0.3446528         0.24092599  0.34615385 0.00250941  0.9166995
SSTRDIST  5.0969450         3.48872894  4.65068687 0.13209682 17.8559508
SSTRVOL   1.2104260         0.53090387  1.11538461 0.40000000  2.5714286
CPDIST5   2.1184928         0.73761928  1.96341236 0.77252973  4.1079018
CPWVOL5   0.4393132         0.22067903  0.38322679 0.09456175  1.1433666
```

Method 2 :

```
> df <- unique(oj[c(1,7:17)])
> install.packages("stargazer")
> library(stargazer)
> stargazer(df, type = "text", median = TRUE, digits = 2)
```

Output :

```
==============================================
Statistic N  Mean  St. Dev.  Min  Median  Max
----------------------------------------------
AGE60     83 0.17    0.06   0.06   0.17  0.31
EDUC      83 0.23    0.11   0.05   0.23  0.53
ETHNIC    83 0.15    0.19   0.02   0.07  1.00
INCOME    83 10.62   0.28   9.87  10.64  11.24
HHLARGE   83 0.12    0.03   0.01   0.11  0.22
WORKWOM   83 0.36    0.05   0.24   0.36  0.47
HVAL150   83 0.34    0.24   0.003  0.35  0.92
SSTRDIST  83 5.10    3.49   0.13   4.65  17.86
SSTRVOL   83 1.21    0.53   0.40   1.12  2.57
CPDIST5   83 2.12    0.74   0.77   1.96  4.11
CPWVOL5   83 0.44    0.22   0.09   0.38  1.14
----------------------------------------------
```

Statistics values generated from the summary table can be used to find out the prior beta values to perform clustering, t-test, classification or other functions in Bayesm package.

Correlation :
To see more details we can generate correlation methods so that we can establish the mutual relationship amongst demographic variables and we can forsee the trends among different variables. Correlation plot can be generated by creating scatterplot matrix. There are two methods through which we can generate a scatterplot matrix described as follow:

Method 1 :

```
> corr = apply(df,2,quantile,probs=c(0.05,.1,.5,.95,.99))
> print(corr)
> corr1 <- cor(df,use = "complete.obs",method="kendall")
> print(round(corr1,digits=3))

> pairs(df,panel = points,main="Correlation plot among variables of dataset")
```

Output :

```
> print(corr)

        AGE60       EDUC     ETHNIC   INCOME    HHLARGE   WORKWOM
5%   0.08976661 0.07129084 0.02645457 10.09390 0.07536104 0.2891824
10%  0.10148080 0.08605602 0.03272247 10.20248 0.08853972 0.2938360
50%  0.17065481 0.22939040 0.07465643 10.63533 0.11122120 0.3556351
95%  0.28258626 0.41904145 0.43983920 10.99819 0.16141721 0.4400615
99%  0.30156013 0.51966864 0.98683932 11.23374 0.20697561 0.4655878

        HVAL150   SSTRDIST   SSTRVOL   CPDIST5   CPWVOL5
5%  0.009970923  0.9781841 0.5217391 0.9995484 0.1484053
10% 0.044155255  1.6813092 0.6030769 1.2982643 0.1843759
50% 0.346153846  4.6506869 1.1153846 1.9634124 0.3832268
95% 0.748465786 12.1663330 2.2480769 3.3366440 0.8140872
99% 0.885625473 16.9362742 2.4813187 3.9385360 0.9606725


> print(round(corr1,digits=3))

           store  AGE60   EDUC ETHNIC INCOME HHLARGE WORKWOM HVAL150 SSTRDIST
store      1.000 -0.165  0.016  0.157 -0.027   0.121   0.016  -0.057    0.212
AGE60     -0.165  1.000 -0.234 -0.113 -0.144  -0.250  -0.462  -0.106    0.059
EDUC       0.016 -0.234  1.000 -0.233  0.565  -0.246   0.442   0.676   -0.074
ETHNIC     0.157 -0.113 -0.233  1.000 -0.492   0.151  -0.083  -0.355    0.262
INCOME    -0.027 -0.144  0.565 -0.492  1.000  -0.089   0.319   0.550   -0.238
HHLARGE    0.121 -0.250 -0.246  0.151 -0.089   1.000  -0.155  -0.306    0.044
WORKWOM    0.016 -0.462  0.442 -0.083  0.319  -0.155   1.000   0.339   -0.152
HVAL150   -0.057 -0.106  0.676 -0.355  0.550  -0.306   0.339   1.000   -0.078
SSTRDIST   0.212  0.059 -0.074  0.262 -0.238   0.044  -0.152  -0.078    1.000
SSTRVOL    0.025 -0.067 -0.074  0.154 -0.135   0.128  -0.022  -0.184    0.082
CPDIST5    0.019  0.040 -0.097 -0.098  0.106   0.088  -0.058  -0.116   -0.047
CPWVOL5   -0.188 -0.075  0.210 -0.274  0.305  -0.097   0.221   0.199   -0.277

          SSTRVOL CPDIST5 CPWVOL5
store       0.025   0.019  -0.188
AGE60      -0.067   0.040  -0.075
EDUC       -0.074  -0.097   0.210
ETHNIC      0.154  -0.098  -0.274
INCOME     -0.135   0.106   0.305
HHLARGE     0.128   0.088  -0.097
WORKWOM    -0.022  -0.058   0.221
HVAL150    -0.184  -0.116   0.199
SSTRDIST    0.082  -0.047  -0.277
SSTRVOL     1.000  -0.048   0.187
CPDIST5    -0.048   1.000   0.060
CPWVOL5     0.187   0.060   1.000
```

Method 2 :

```
> library(s20x)

> pairs(df,col="green",pch = 20)

> pairs20x(df)
```

Scatterplot Matrix Output:

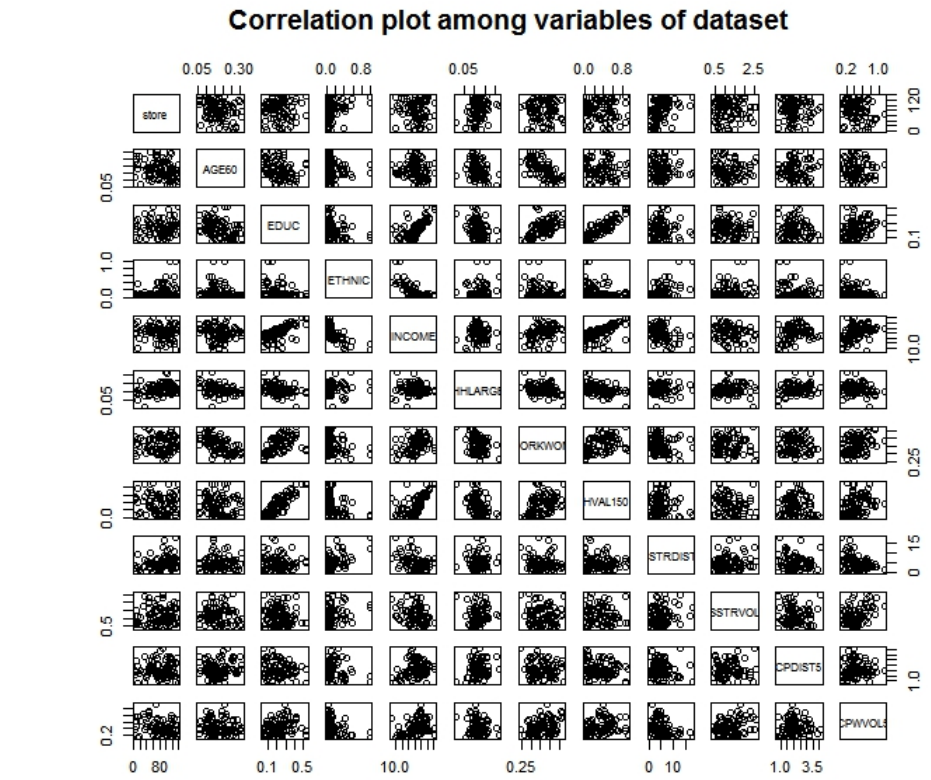

Figure 14: Correlation scatterplot using Method 1

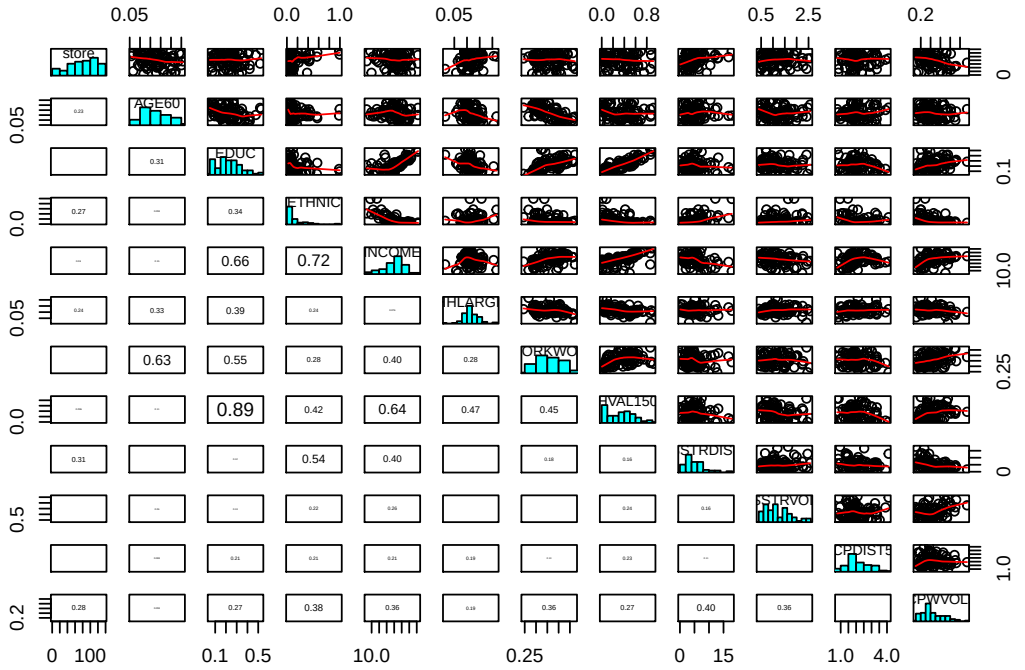

Figure 15: Scatterplot using Method 2

Linear Models :

Now we can generate the linear models for each of demographic and competitive variables so that we can see the positive or negative trends.

```
> library(lattice)
> par(mfrow = c(2,2))
> df <- unique(oj[c(1,7:17)])
> var_list<- names(df)[2:12]
> models <- lapply(var_list,
          function(x)
             {
              mod = lm(substitute(store ~ i, list(i = as.name(x))), data = df)
              })
> par(mfrow = c(2,2))
> invisible(lapply(models,plot))
```

Output :

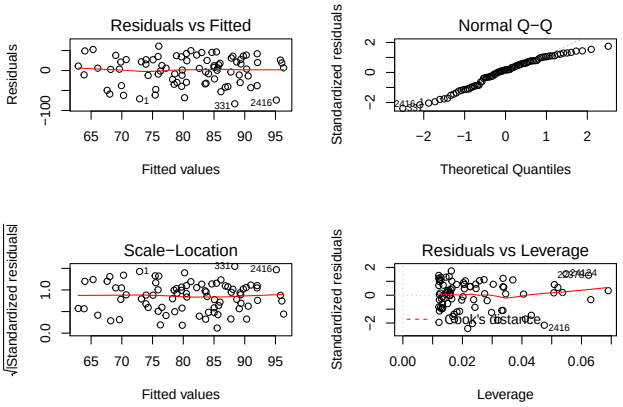

Figure 16: Regression models for AGE60

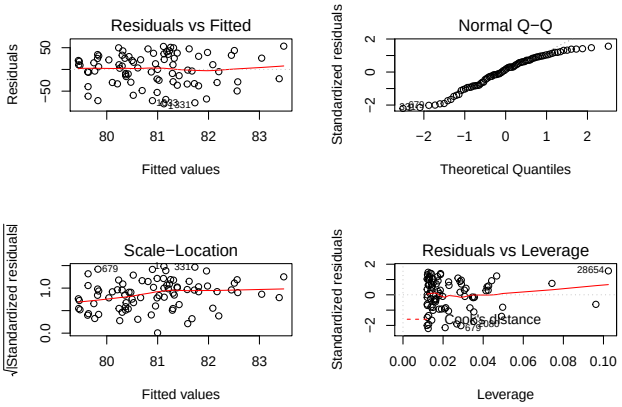

Figure 17: Regression models for EDUC

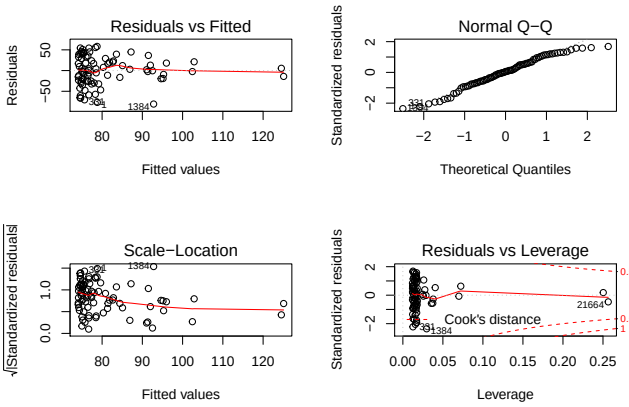

Figure 18: Regression models for ETHNIC

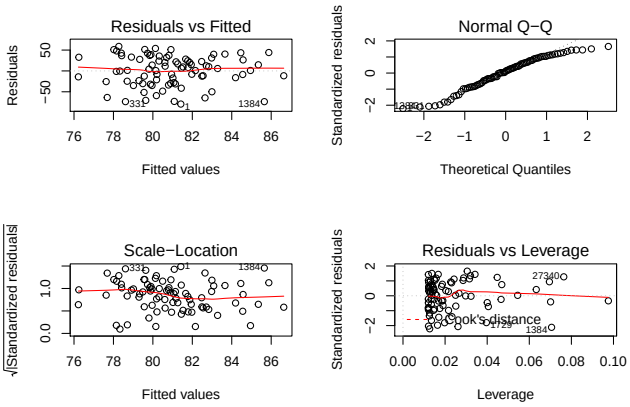

Figure 19: Regression models for INCOME

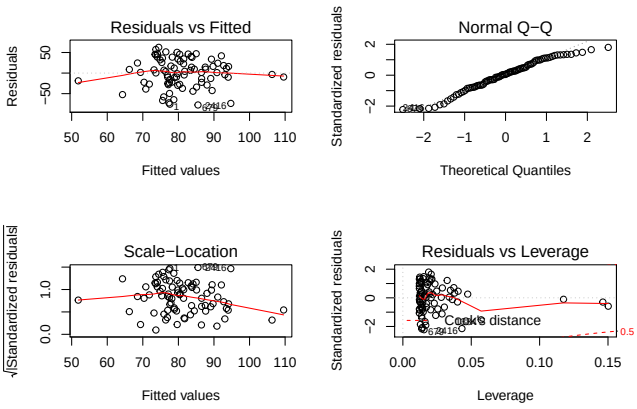

Figure 20: Regression models for HHLARGE

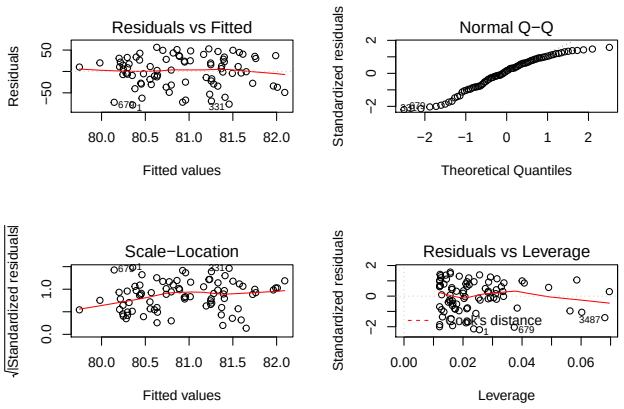

Figure 21: Regression models for WORKWOM

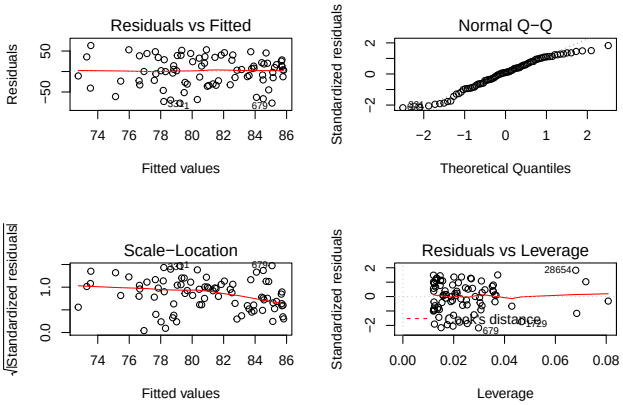

Figure 22: Regression models for HVAL150

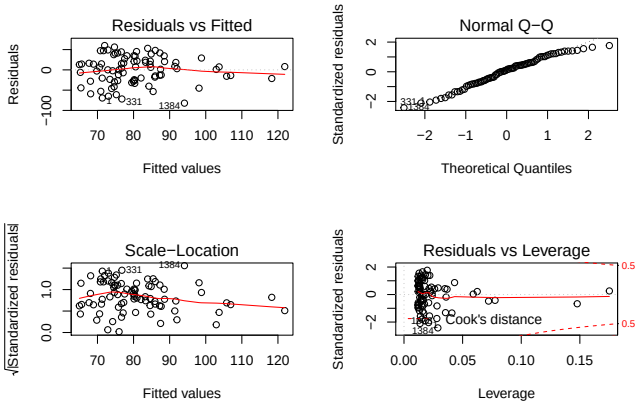

Figure 23: Regression models for STRDIST

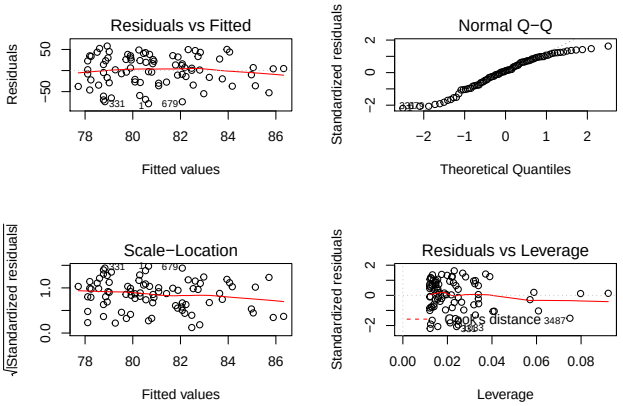

Figure 24: Regression models for SSTRVOL

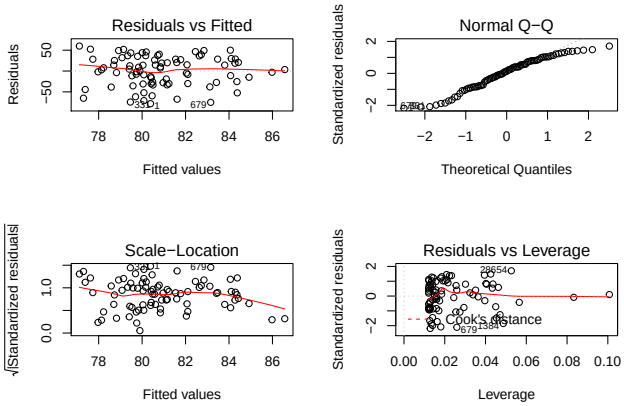

Figure 25: Regression models for CPDIST5

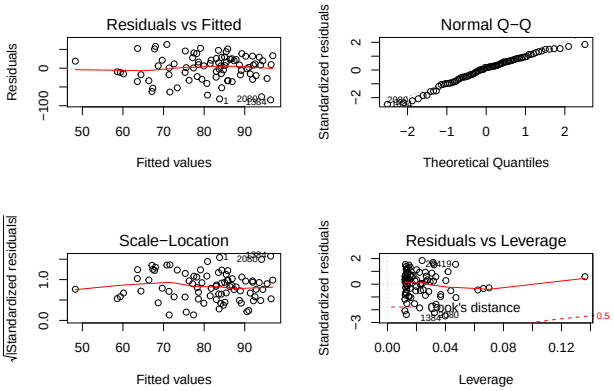

Figure 26: Regression models for CPWVOL5

Histograms : By using histograms and density curve, we can interpret the distribution of the demographic and competitive variables whether the distribution is normal distribution or not.

```
> library(ggplot2)
> df <- unique(oj[c(1,7:17)])
> var_list<- names(df)[2:12]
> plots <- function(i){
              ggplot(df, aes(x = df[i])) +
              geom_histogram(aes(y = ..density..),colour = "black",
                             fill = "yellow",binwidth = 0.02) +
              geom_density(alpha = 0.2, fill = "blue",linetype = "dashed") +
              geom_vline(aes(xintercept=mean(df[i], na.rm = TRUE)),
                             color = "black",linetype = "dashed", size = 1) +
              xlab(names(df)[i]) +
              ggtitle(paste("Histogram plot of ",names(df)[i]))}
> par(mfrow = c(2,2))
> lapply(2:12,FUN = plots)
```

Output :

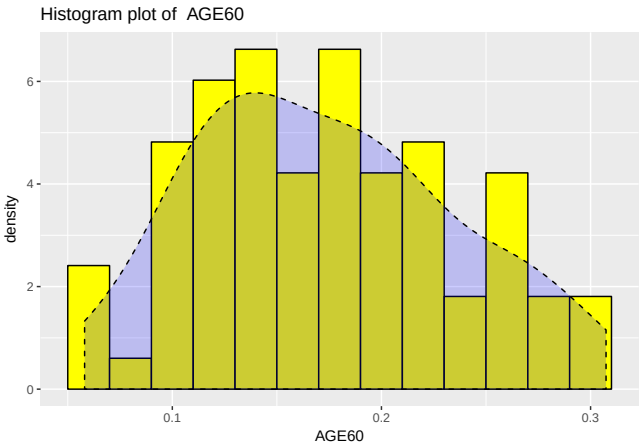

Figure 27: Histogram and Density plots for AGE60

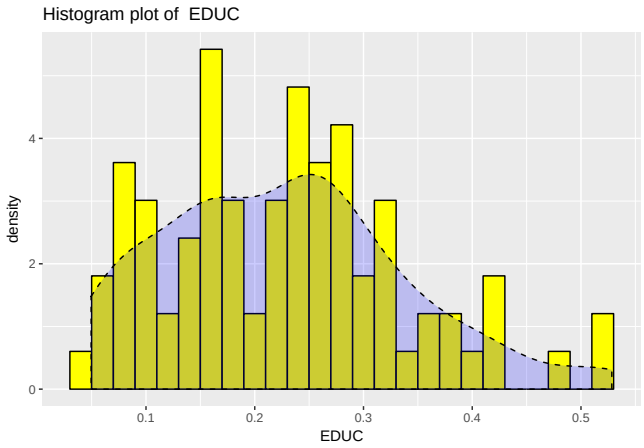

Figure 28: Histogram and Density plots for EDUC

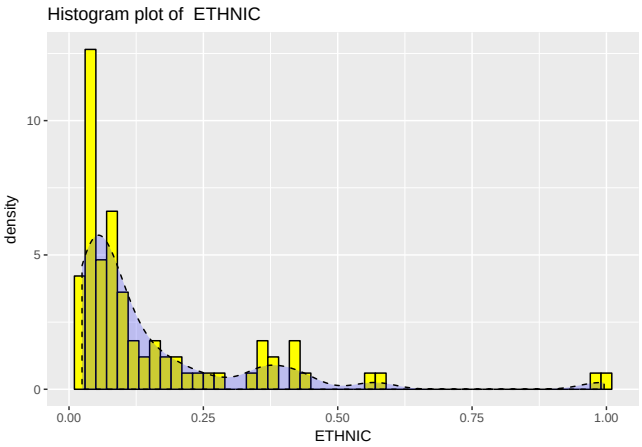

Figure 29: Histogram and Density plots for ETHNIC

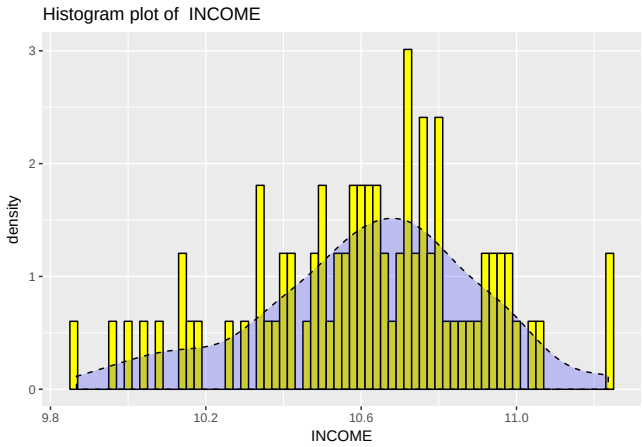

Figure 30: Histogram and Density plots for INCOME

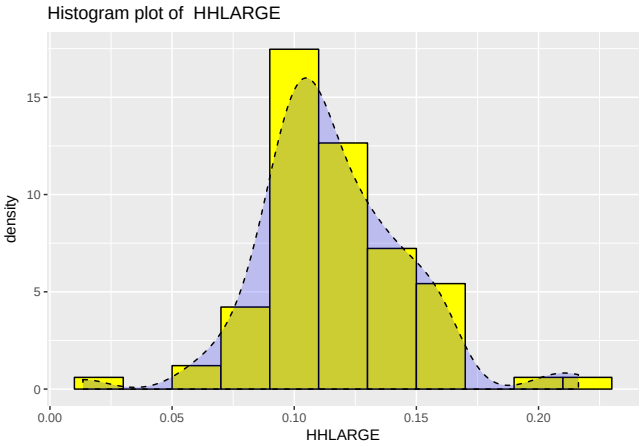

Figure 31: Histogram and Density plots for HHLARGE

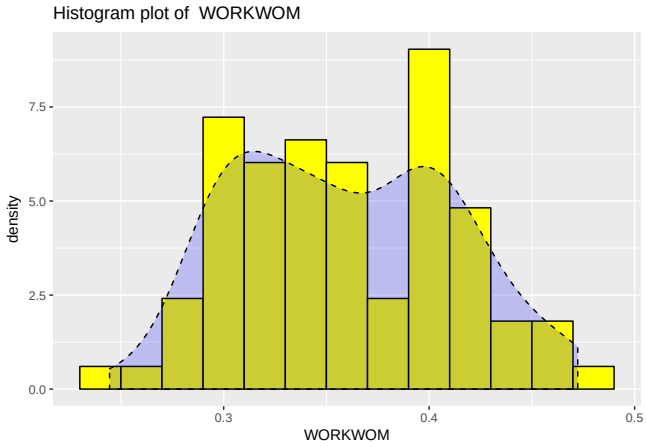

Figure 32: Histogram and Density plots for WORKWOM

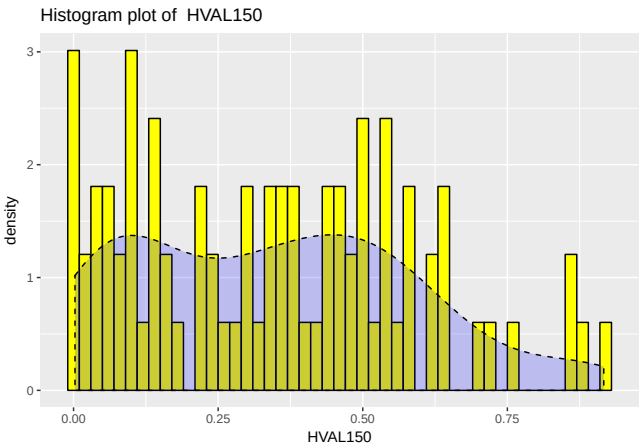

Figure 33: Histogram and Density plots for HVAL150

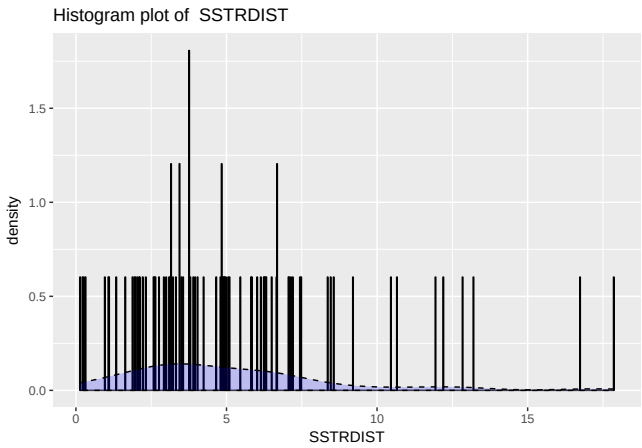

Figure 34: Histogram and Density plots for STRDIST

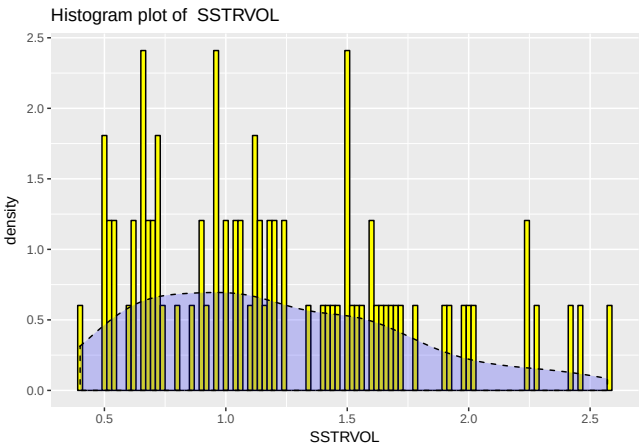

Figure 35: Histogram and Density plots for SSTRVOL

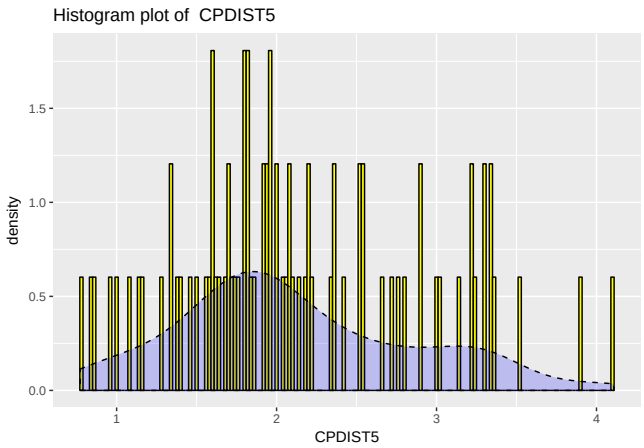

Figure 36: Histogram and Density plots for CPDIST5

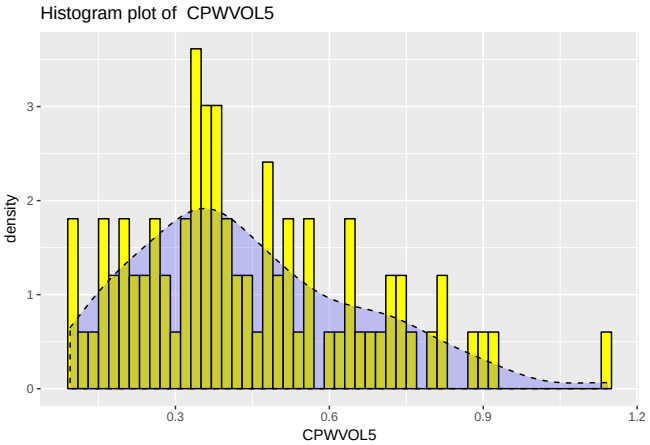

Figure 37: Histogram and Density plots for CPWVOL5

From Figure 15, we can see that the correlation coefficients between EDUC and INCOME, EDUC and HVAL150, AGE60 and WORKWOM are 0.66, 0.89, 0.63 respectively, which means they have influences to the sales, so we can use regression model to make use of the information by using the following code:

```
>reg <- lm(df$store~df$AGE60+df$INCOME+df$EDUC+df$ETHNIC+
           df$HVAL150+df$HHLARGE+df$WORKWOM+df$SSTRDIST+
           df$SSTRVOL+df$CPDIST5+df$CPWVOL5)
>summary(reg)
```

Output :

```
Call:
lm(formula = df$store ~ df$AGE60 + df$INCOME + df$EDUC + df$ETHNIC +
```

```
    df$HVAL150 + df$HHLARGE + df$WORKWOM + df$SSTRDIST + df$SSTRVOL +
    df$CPDIST5 + df$CPWVOL5)

Residuals:
    Min      1Q  Median      3Q     Max
-73.087 -22.128  -0.087  22.716  64.968

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -208.779    310.956  -0.671   0.5041
df$AGE60     -38.375    120.740  -0.318   0.7515
df$INCOME     21.789     31.483   0.692   0.4911
df$EDUC       77.623     95.245   0.815   0.4178
df$ETHNIC     35.161     35.236   0.998   0.3217
df$HVAL150   -25.203     39.299  -0.641   0.5234
df$HHLARGE   161.804    217.625   0.743   0.4596
df$WORKWOM    65.330    138.795   0.471   0.6393
df$SSTRDIST    1.759      1.379   1.275   0.2064
df$SSTRVOL     8.738      9.196   0.950   0.3452
df$CPDIST5     4.716      5.900   0.799   0.4268
df$CPWVOL5   -47.787     24.259  -1.970   0.0528 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 33.17 on 71 degrees of freedom
Multiple R-squared:  0.2623, Adjusted R-squared:  0.148
F-statistic: 2.295 on 11 and 71 DF,  p-value: 0.01816
```

The p-value for all the variables are grater than 0.05, it seems we do not have the strong evidence that the regression model we have created will show the significant factors explaining the model and so it is a bad choice to include this combination in a model. So we should choose the variables whose p-value will be less than 0.05 so that it will provide evidences for sales.

Summaryby :

To gain more insight regarding mean of the type of people buy the orange juice from which store, we can create summary by using cross table with function summaryBy.

```
> library(doBy)
> summaryBy(AGE60 ~ store, data = oj, FUN = c(mean), na.rm =TRUE)
```

Output :

```
   store AGE60.mean
1      2 0.23286473
2      5 0.11736803
3      8 0.25239404
4      9 0.26911902
5     12 0.17834141
6     14 0.21394927
7     18 0.27231337
8     21 0.06689646
9     28 0.21330879
10    32 0.25495303
11    33 0.13416997
12    40 0.18185180
```

t-Test :

Welch t-test can also help in determining comparison between two groups by defining confidence interval for the variables. Below example is defining t-test between AGE60 and EDUC ans shows that p - value is much less than 0.05, so we can consider that NULL hypothesis is true and it provides strong evidence that these variable are factors that drives sales of orange juice.

```
> t.test(df$AGE60,df$EDUC)
```

Output :

```
  Welch Two Sample t-test

data:  df$AGE60 and df$EDUC
t = -3.7769, df = 128.79, p-value = 0.0002415
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.08046517 -0.02514245
sample estimates:
mean of x mean of y
0.1729724 0.2257762
```

Conclusion :

To identify patterns in the dataset, data cleansing is must before commencement of any function. Data cleansing help in removing outliers and removing any missing data. Because if there are any junk data present in dataset then models will not produce accurate results. Second, using basic descriptive statistics values such as mean, median, standard deviation can be found; that can be further used to determined beta values for t-test, clustering (k means method), etc. Lastly, different models can be used to establish relationships among the demographic variables.

Approach :**Data Cleansing => Descriptive Statistics => Model Applications**

Marketing team can make use of result obtained from correlations, regression, t-test so that they can use highly correlated variables such as EDUC, INCOME and HHVAL150. From this we can interpret that potential customers are well educated and receive high income because plot has shown positive correlation. Then team can form strategies to target more of these customers by using correlation patterns among other variables as well. Also from the results generated from histograms, team can plot the probability distribution of the variables and can predict the future results. To sum up, descriptive statistics is very helpful in determining the trends to retain old customers and how to target potential customers.