VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

Programming for Data Science

CSE 3046

THEORY DIGITAL ASSIGNMENT

Student Name : Pranjal Gupta

Student ID : 19BDS0081

Submitted to: Prof. Tamizharzi T

Date of Submission: 02/12/2021

Marketing Analytics

Performing analysis on success of marketing campaigns conducted using EDA and Classification models

Marketing analytics comprises the processes and technologies that enable marketers to evaluate the success of their marketing initiatives. This is accomplished by measuring performance. It tells a company how effective their marketing programs are and how they are performing.
Marketing Campaigns are a vital part of how company's promote their interests, whether that be raising awareness for a new product or capturing customer feedback. It is important for any company to be able to gauge customer's participation in the marketing campaigns, assess the success of past campaigns, and propose data-driven solutions to increase participation in future campaigns.

In this assignment, we will seek answers to a few chief questions:

- What does the Average customer look like for our company?
- What Products and Channels of Revenue are best performing?
- Which Marketing Campaigns were most successful?
- What factors contribute to the success of our current campaign?

These questions can be through Visualization as well as complex machine learning models to see if we can find contributing factors to the success of our past campaigns.

Dataset – https://bit.ly/32Gx4XO

	A
1	ID;Year_Birth;Education;Marital_Status;Income;Kidhome;Teenhome;Dt_Customer;Recency;MntWines;MntFruits;MntMeatProduct
2	5524;1957;Graduation;Single;58138;0;0;2012-09-04;58;635;88;546;172;88;88;3;8;10;4;7;0;0;0;0;0;0;3;11;1
3	2174;1954;Graduation;Single;46344;1;1;2014-03-08;38;11;1;6;2;1;6;2;1;1;2;5;0;0;0;0;0;0;3;11;0
4	4141;1965;Graduation;Together;71613;0;0;2013-08-21;26;426;49;127;111;21;42;1;8;2;10;4;0;0;0;0;0;0;3;11;0
5	6182;1984;Graduation;Together;26646;1;0;2014-02-10;26;11;4;20;10;3;5;2;2;0;4;6;0;0;0;0;0;0;3;11;0
6	5324;1981;PhD;Married;58293;1;0;2014-01-19;94;173;43;118;46;27;15;5;5;3;6;5;0;0;0;0;0;0;3;11;0
7	7446;1967;Master;Together;62513;0;1;2013-09-09;16;520;42;98;0;42;14;2;6;4;10;6;0;0;0;0;0;0;3;11;0
8	965;1971;Graduation;Divorced;55635;0;1;2012-11-13;34;235;65;164;50;49;27;4;7;3;7;6;0;0;0;0;0;0;3;11;0
9	6177;1985;PhD;Married;33454;1;0;2013-05-08;32;76;10;56;3;1;23;2;4;0;4;8;0;0;0;0;0;0;3;11;0

- ID: the unique identification code for every customer
- Year_Birth: The Year of a customer's birth
- Education: The level of education that a customer completed
- Marital_Status: Status of Marriage
- Income: Annual Income
- Kidhome: # of children under the age of 13 in Customer's household
- Teenhome: # of children between 13-19 in Customer's household
- Dt_Customer: Date of Customer Enrollment
- Recency: # of days since last purchase
- MntWines: Dollar amount of Wines purchased in last 2 years
- MntFruits: Dollar amount of Fruits purchased in last 2 years
- MntMeatProducts: Dollar amount of Meat products purchased in the last 2 years
- MntFishProducts: Dollar amount of Fish products purchased in the last 2 years
- MntSweetProducts: Dollar amount of Sweet products purchased in the last 2 years
- MntGoldProds: Dollar amount of Gold products purchased in the last 2 years
- NumDealsPurchases: # of purchases made with discount
- NumWebPurchases: # of purchases made through the company's website
- NumCatalogPurchases: # of purchases made using the catalog
- NumStorePurchases: # of purchases made directly in-store
- NumWebVisitsMonth: # of visits made through the company's website
- AcceptedCmp1: 1 if customer accepted the offer in the 1st campaign, 0 otherwise
- AcceptedCmp2: 1 if customer accepted the offer in the 2nd campaign, 0 otherwise
- AcceptedCmp3: 1 if customer accepted the offer in the 3rd campaign, 0 otherwise
- AcceptedCmp4: 1 if customer accepted the offer in the 4th campaign, 0 otherwise
- AcceptedCmp5: 1 if customer accepted the offer in the 5th campaign, 0 otherwise
- Complain: 1 if customer complained in the last 2 years, 0 otherwise
- Response: 1 if customer accepted the offer in the last campaign, 0 otherwise

Code -

```
library(tidyverse)
library(dplyr)
library(ggplot2)
require(scales)
```

#Data Cleaning and Manipulation

df <- read.csv('C:\\Users\\Pranjal Gupta\\Downloads\\marketing_campaign.csv',

header = TRUE,sep = ";")

head(df)

str(df)

```
> str(df)
'data.frame':   2240 obs. of  29 variables:
 $ ï..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  "2012-09-04" "2014-03-08" "2013-08-21" "2014-02-10" ...
 $ 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 ...
 $ 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 ...
 $ Z_CostContact      : int  3 3 3 3 3 3 3 3 3 3 ...
 $ Z_Revenue          : int  11 11 11 11 11 11 11 11 11 11 ...
 $ Response           : int  1 0 0 0 0 0 0 0 1 0 ...
```

sum(is.na(df))

colSums(is.na(df))

```
> sum(is.na(df))
[1] 24
> colSums(is.na(df))
              ï..ID          Year_Birth           Education      Marital_Status              Income
                  0                   0                   0                   0                  24
            Kidhome            Teenhome         Dt_Customer             Recency            MntWines
                  0                   0                   0                   0                   0
          MntFruits     MntMeatProducts     MntFishProducts    MntSweetProducts        MntGoldProds
                  0                   0                   0                   0                   0
  NumDealsPurchases     NumWebPurchases NumCatalogPurchases   NumStorePurchases   NumWebVisitsMonth
                  0                   0                   0                   0                   0
       AcceptedCmp3        AcceptedCmp4        AcceptedCmp5        AcceptedCmp1        AcceptedCmp2
                  0                   0                   0                   0                   0
           Complain       Z_CostContact           Z_Revenue            Response
                  0                   0                   0                   0
>
```

We check for unique values in the dataset

sapply(df,function(x) length(unique(x)))

```
sapply(df,function(x) length(unique(x)))
              i..ID          Year_Birth           Education      Marital_Status              Income
               2240                  59                   5                   8                1975
            Kidhome            Teenhome         Dt_Customer             Recency            MntWines
                  3                   3                 663                 100                 776
          MntFruits     MntMeatProducts     MntFishProducts    MntSweetProducts        MntGoldProds
                158                 558                 182                 177                 213
  NumDealsPurchases     NumWebPurchases NumCatalogPurchases   NumStorePurchases   NumWebVisitsMonth
                 15                  15                  14                  14                  16
       AcceptedCmp3        AcceptedCmp4        AcceptedCmp5        AcceptedCmp1        AcceptedCmp2
                  2                   2                   2                   2                   2
           Complain       Z_CostContact           Z_Revenue            Response
                  2                   1                   1                   2
```

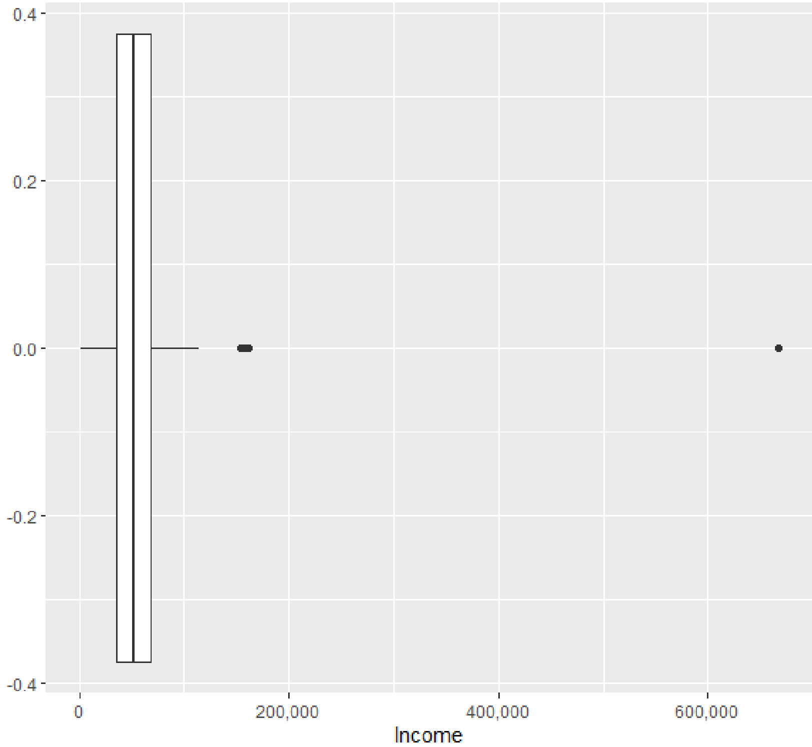
```
df <- df %>%
  mutate(Dt_Customer = as.Date(Dt_Customer)) #create Date column

count(df$Marital_Status)
#there are two many values, hence merging to form better categories
df$Rel_Status[df$Marital_Status %in% c('Alone', 'Divorced', 'Widow', 'Single')] <- 'Single'
df$Rel_Status[df$Marital_Status %in% c('Married', 'Together')] <- 'Together'
df$Rel_Status[df$Marital_Status %in% c('Absurd', 'YOLO')] <- ''

count(df$Education)
# nothing to change in this as everything indicates to a conclusion
```

```
> count(df$Marital_Status)
         x freq
1   Absurd    2
2    Alone    3
3 Divorced  232
4  Married  864
5   Single  480
6 Together  580
7    Widow   77
8     YOLO    2
> #there are two many values, hence merging to form better categories
> df$Rel_Status[df$Marital_Status %in% c('Alone', 'Divorced', 'Widow', 'Single')] <- 'Single'
> df$Rel_Status[df$Marital_Status %in% c('Married', 'Together')] <- 'Together'
> df$Rel_Status[df$Marital_Status %in% c('Absurd', 'YOLO')] <-''
>
> count(df$Education)
           x freq
1   2n Cycle  203
2      Basic   54
3 Graduation 1127
4     Master  370
5        PhD  486
> # nothing to change in this as everything indicates to a conclusion
```

```
summary(df$Income)
ggplot(df, aes(x = Income)) +geom_boxplot()+ scale_x_continuous(labels = comma)
#As we can see a few outliers, it is disturbing the overall value of the column as seen in summary
```

#there are also, 24 missing values, to fill those as well we need to remove outliers

```
> summary(df$Income)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
   1730   35303   51382   52247   68522  666666      24 
> ggplot(df, aes(x = Income)) +geom_boxplot()+ scale_x_continuous(labels = comma)
Warning message:
Removed 24 rows containing non-finite values (stat_boxplot).
```



outliers <- boxplot(df$Income, plot = FALSE)$out

df <- df %>%

  filter(Income < max(outliers) - 1)

now we fill NA values in Income variable with mean

df$Income[is.na(df$Income)] <- mean(df$Income, na.rm = TRUE)

colSums(is.na(df))

```
> outliers <- boxplot(df$Income, plot = FALSE)$out
> df <- df %>%
+   filter(Income < max(outliers) - 1)
> # now we fill NA values in Income variable with mean
> df$Income[is.na(df$Income)] <- mean(df$Income, na.rm = TRUE)
> colSums(is.na(df))
             ï..ID          Year_Birth           Education      Marital_Status              Income
                 0                   0                   0                   0                   0
           Kidhome            Teenhome         Dt_Customer             Recency            MntWines
                 0                   0                   0                   0                   0
         MntFruits     MntMeatProducts     MntFishProducts    MntSweetProducts        MntGoldProds
                 0                   0                   0                   0                   0
 NumDealsPurchases     NumWebPurchases NumCatalogPurchases   NumStorePurchases   NumWebVisitsMonth
                 0                   0                   0                   0                   0
      AcceptedCmp3        AcceptedCmp4        AcceptedCmp5        AcceptedCmp1        AcceptedCmp2
                 0                   0                   0                   0                   0
          Complain       Z_CostContact           Z_Revenue            Response          Rel_Status
                 0                   0                   0                   0                   0
```

summary(df$Z_CostContact)

summary(df$Z_Revenue)

#these columns are not required anymore for further Analysis

```
summary(df$Z_CostContact)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
      3       3       3       3       3       3 
summary(df$Z_Revenue)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
     11      11      11      11      11      11 
```

drop <- c("Marital_Status","Kidhome","Teenhome","Z_CostContact","Z_Revenue")

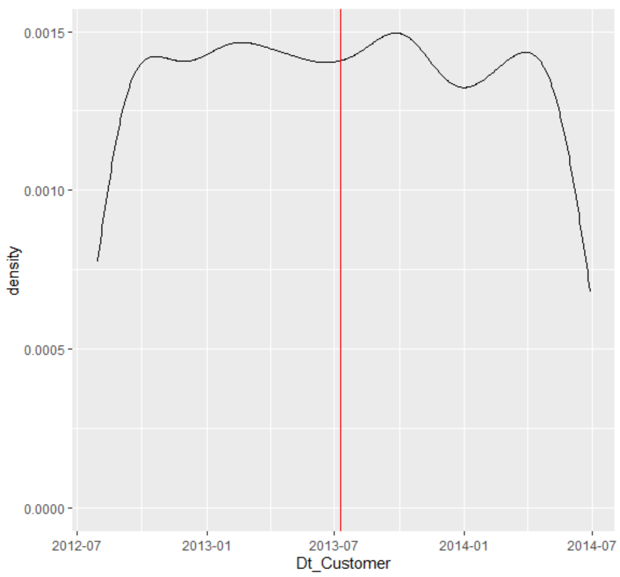
df = df[,!(names(df) %in% drop)]

head(df)

#Date signed up

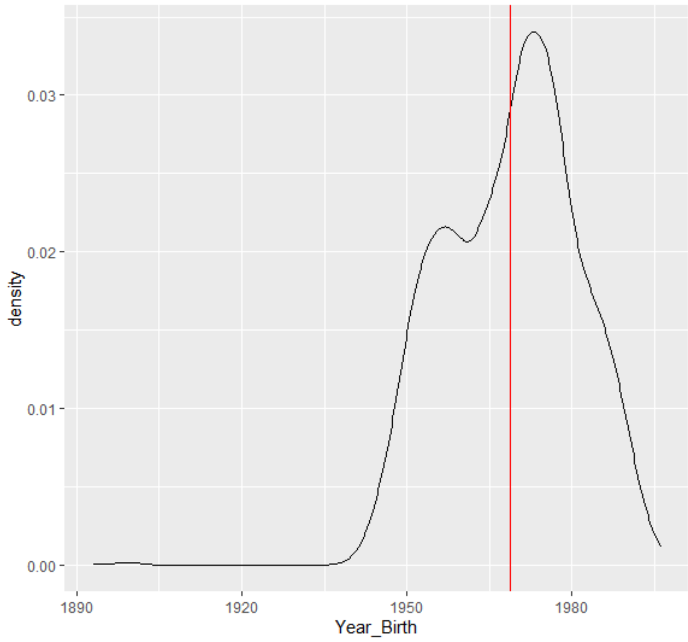
ggplot(df, aes(Dt_Customer)) + geom_density() +

geom_vline(aes(xintercept = mean(Dt_Customer)), color = 'red')



This plot shows the average customer joined around July of 2013. There is a little variation over the time period in when Customer's enrolled with the company, but the data seems to be bound to customers between July of 2012 and July of 2014.

```
#Year born - to find the popular age group of the company
ggplot(df, aes(Year_Birth)) + geom_density() +
  geom_vline(aes(xintercept = mean(Year_Birth)), color = 'red')
```



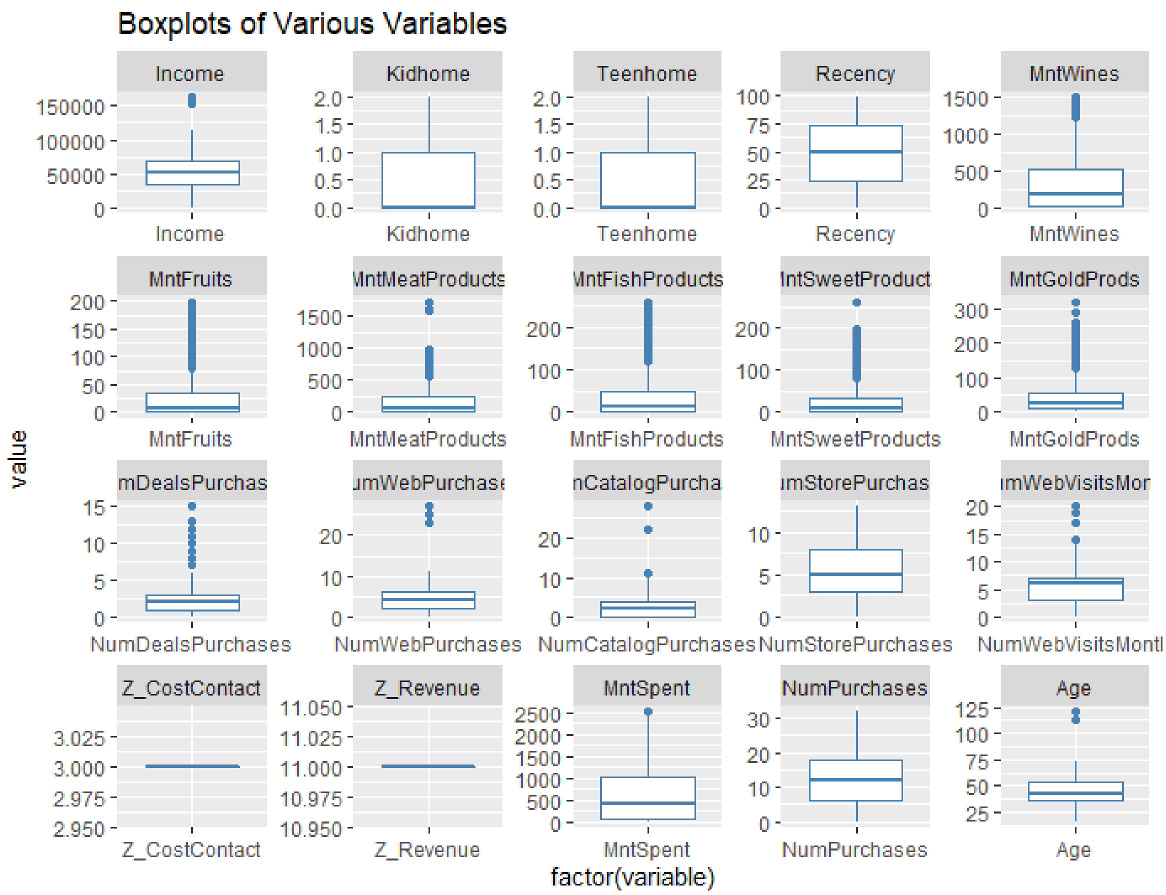
The company seems to be most populated by the people born around 1960s and 1970s, taking a decline when it comes to people born around and after 1980s.

```
df <- df %>%
  #creating new variables based off old ones
  mutate(MntSpent = MntFishProducts + MntMeatProducts + MntFruits +
MntSweetProducts + MntWines + MntGoldProds) %>%
  mutate(NumPurchases = NumCatalogPurchases + NumStorePurchases +
NumWebPurchases) %>%
  mutate(AcceptedCmp = AcceptedCmp1 + AcceptedCmp2 + AcceptedCmp3 +
AcceptedCmp4 + AcceptedCmp5) %>%
  mutate(Age = as.numeric(format(Dt_Customer, format = '%Y')) - Year_Birth)
head(df)
```

```
        6            0            0            0            0
 Response Rel_Status MntSpent NumPurchases AcceptedCmp Age
        1     Single     1617           22           0  55
        0     Single       27            4           0  60
        0   Together      776           20           0  48
        0   Together       53            6           0  30
        0   Together      422           14           0  33
        0   Together      716           20           0  46
```

New columns like - MntSpent (a summation of the amount of money a customer spent products), NumPurchases (a summation of purchases made from the catalogue, web, or in-store), AcceptedCmp (a summation of the previous campaigns a customer participated in), and Age (the age at which a customer became enrolled at the company) were added. Several columns - Kidhome, Marital_Status. Teenhome, Z_CostContact, Z_Revenue were dropped

```
library(reshape) #melt()

#melt data frame into long format

rev <- c('ï..ID','Year_Birth','AcceptedCmp1', 'AcceptedCmp2', 'AcceptedCmp3',

'AcceptedCmp4', 'AcceptedCmp5', 'Complain', 'Education', 'Rel_Status',

'Dt_Customer', 'Response', 'AcceptedCmp')


df_new <- df %>%

  select(-one_of(rev)) %>%

  melt()


ggplot(df_new, aes(factor(variable), value)) + geom_boxplot(color = 'steelblue') +

  facet_wrap(~variable,scale='free') + labs(title = 'Boxplots of Various Variables')
```



The plot looks at the distributions of the numeric variables using Boxplots.

- **Age**: It seems to be an anamoly present in the Age Boxplot. Besides that, the Age variable is normally distributed with the average age being slightly less than 50 years old.
- **Income**: The average salary can easily be seen to be about 50k which is similar to the greater population of people
- **MntFishProducts, MntFruits, MntGoldProds, MntSweetProducts**: This is very right-skewed distribution indicating either mass buying or continued interest in our store
- **MntMeatProducts**: It can be expected that a customer will buy a greater proportion of Meat Products from our store than previous products as the mean is easily around 150 dollars.
- **MntSpent**: The typical amount of money a customer spent in our stores over the past 2 years is 500 dollars, but up to 50% of the customer base spent upwards of 500 - 2500 dollars.

- **MntWines**: The average customers are expected to spend more on Wines than Meat Products meaning it may be the top source of revenue.
- **NumCatalogPurchases**: The average number of catalogue purchases a customer makes is around 5, but some customer's enjoy purchasing many items from the catalogue.
- **NumPurchases**: It is very normally distributed with mean greater than 10 and a range of anywhere between 0 and 30 purchases made.
- **NumStorePurchases**: It is also normally distributed with an average of about 5 in-store purchases and a range of anywhere between 0 and 13.
- **NumWebPurchases, NumWebVisitsMonth**: Some customers enjoy the website for their purchases much more and make more purchases there.
- **Recency**: Nearly perfectly normally distributed, the average number of days a customer has gone with making a purchase is 50 days (or nearly 2 months) while the maximum number of days a customer has gone without purchasing a product is 100 days (slightly more than 3 months).

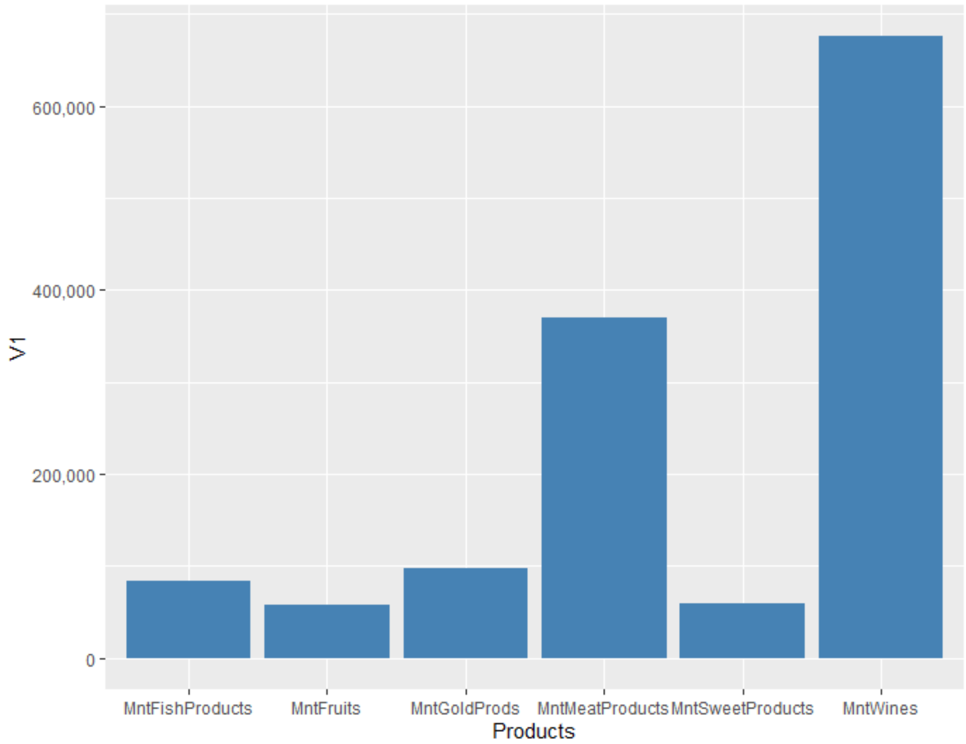
```
#remove outliers from age variable as seen in the boxplot
outliers <- boxplot(df$Age, plot = FALSE)$out
df <- df %>%
  filter(Age < min(outliers))
```

EDA using visualization

```
#list of products
products <- c('MntWines', 'MntFruits', 'MntMeatProducts', 'MntFishProducts',
'MntSweetProducts', 'MntGoldProds')

#sum amounts spent on products and set these values in df
products_df <- df %>%
  select(products) %>%
  summarize_each(sum) %>%
  t() %>% #to calculate transpose of a matrix or Data Frame.
  as.data.frame() %>%
```

```
  rownames_to_column('Products')
products_df

ggplot(products_df,aes(x=Products, y=V1)) + geom_bar(fill="steelblue",stat = "identity") + scale_y_continuous(labels = comma)
```

```
> products_df
          Products     V1
1         MntWines 676074
2        MntFruits  58391
3  MntMeatProducts 370045
4  MntFishProducts  83397
5 MntSweetProducts  59895
6     MntGoldProds  97415
```

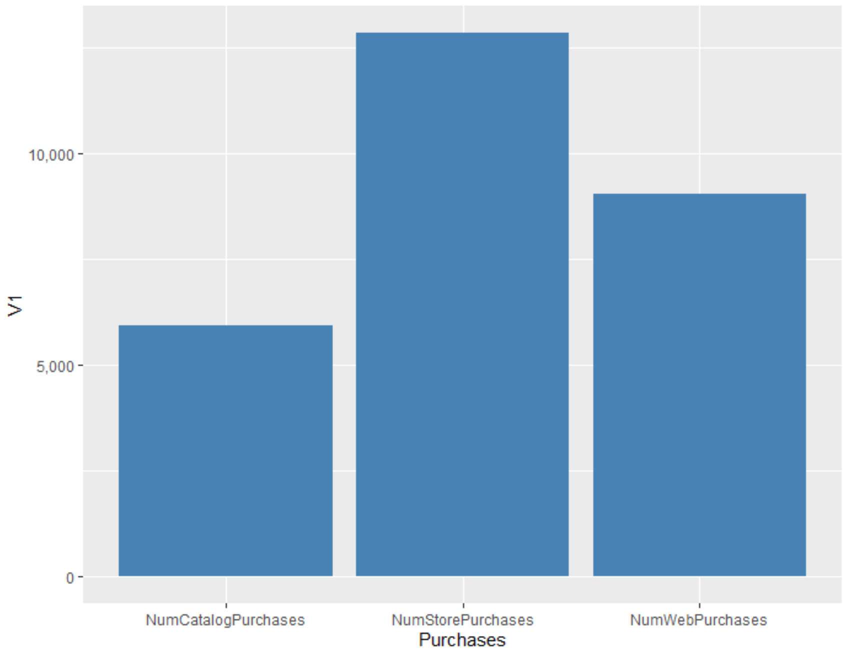


As we can see in the graph, Wines easily account for a majority of total sales, with meat products being a second with nearly half the sales as Wines, Other products accrue a similar amount of sales. Total Sales in the past 2 years sits at 1341984 dollar.

```
#list of purchases
purchases <- c('NumCatalogPurchases', 'NumStorePurchases',
'NumWebPurchases')
```

```
#sum amounts spent on purchases and set these values in df
purchases_df <- df %>%
  select(purchases) %>%
  summarize_each(sum) %>%
  t() %>% #to calculate transpose of a matrix or Data Frame.
  as.data.frame() %>%
  rownames_to_column('Purchases')
purchases_df

ggplot(purchases_df,aes(x=Purchases, y=V1)) + geom_bar(fill="steelblue",stat =
"identity") + scale_y_continuous(labels = comma)
```

```
> purchases_df
            Purchases    V1
1 NumCatalogPurchases  5918
2   NumStorePurchases 12852
3     NumWebPurchases  9050
```



Most of our sales do come from our store, but our web portal and catalogue are far from underutilized. Total number of purchases we've gotten in the past 2 years is 27,757.

library(ggcorrplot)

df_new1 <- df %>%

 select(-one_of(rev))

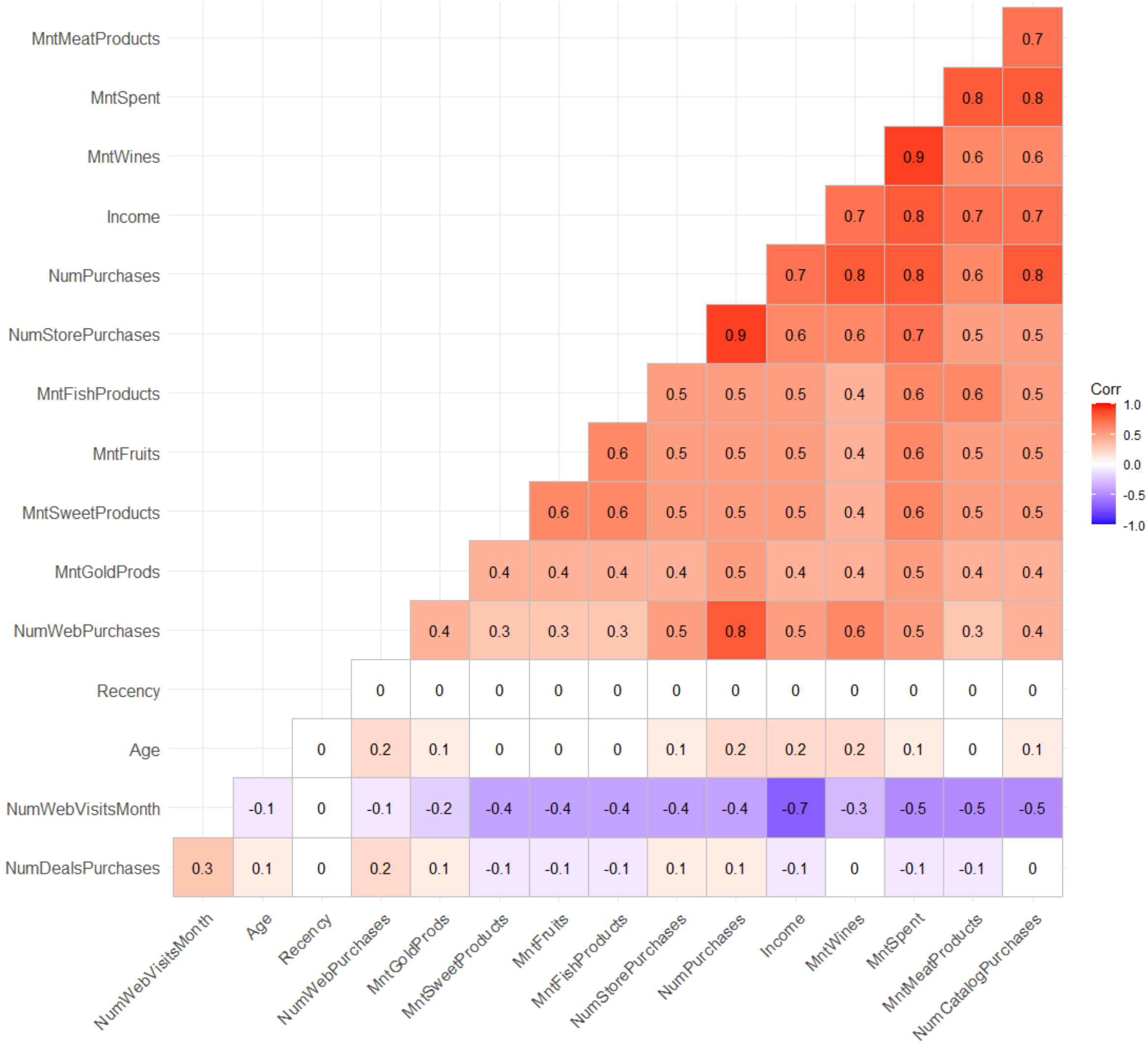
head(df_new1)

```
> #library(GGally) #ggcorr() and ggpairs()
> library(ggcorrplot)
> df_new1 <- df %>%
+   select(-one_of(rev))
> head(df_new1)
  Income Recency MntWines MntFruits MntMeatProducts MntFishProducts MntSweetProducts MntGoldProds
1  58138      58      635        88             546             172               88           88
2  46344      38       11         1               6               2                1            6
3  71613      26      426        49             127             111               21           42
4  26646      26       11         4              20              10                3            5
5  58293      94      173        43             118              46               27           15
6  62513      16      520        42              98               0               42           14
  NumDealsPurchases NumWebPurchases NumCatalogPurchases NumStorePurchases NumWebVisitsMonth MntSpent
1                 3               8                  10                 4                 7     1617
2                 2               1                   1                 2                 5       27
3                 1               8                   2                10                 4      776
4                 2               2                   0                 4                 6       53
5                 5               5                   3                 6                 5      422
6                 2               6                   4                10                 6      716
  NumPurchases Age
1           22  55
2            4  60
3           20  48
4            6  30
5           14  33
6           20  46
> correlation_matrix <- round(cor(df_new1),1)
> ggcorrplot(correlation_matrix, hc.order =TRUE, type ="lower", method ="square",lab =TRUE)
```

correlation_matrix <- round(cor(df_new1),1)

ggcorrplot(correlation_matrix, hc.order =TRUE, type ="lower", method ="square",lab

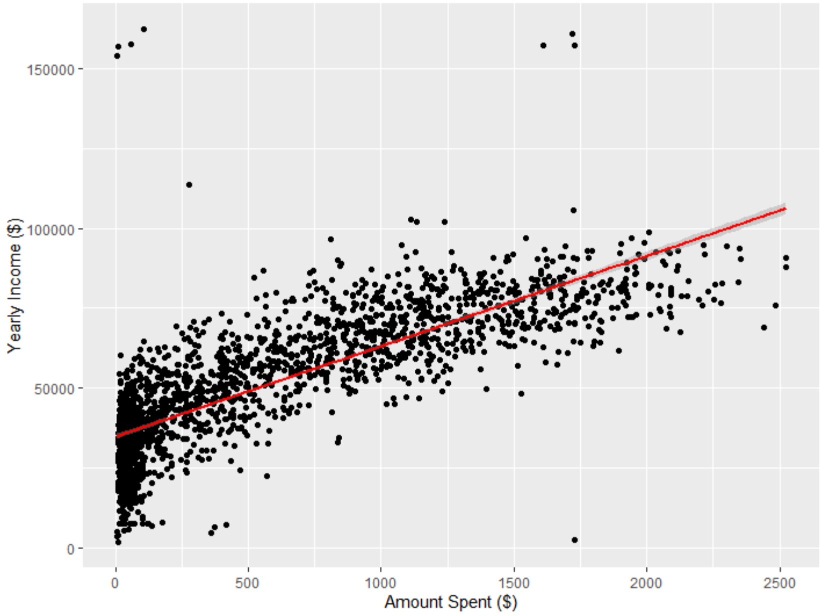
=TRUE)



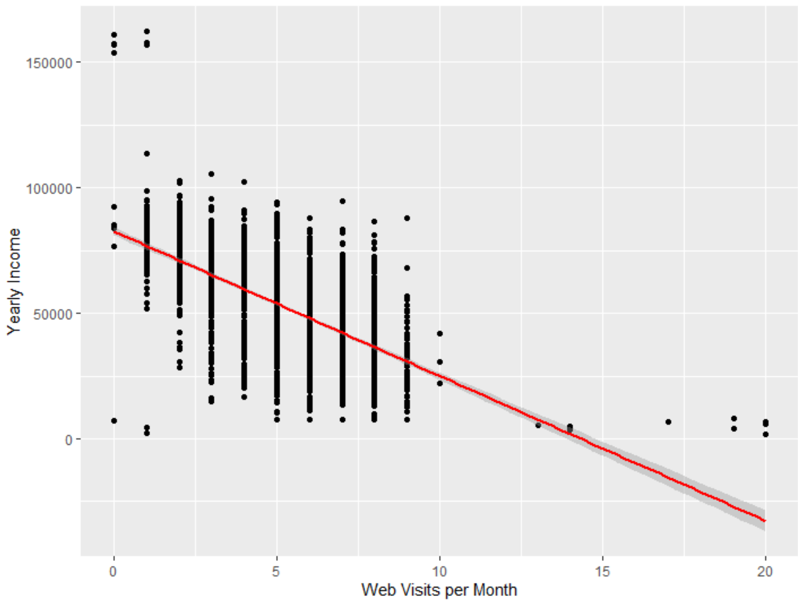
The most positively correlated data include Income to MntSpent, suggesting that as a customer's Income increases it is expected of them to spend more on the products. MntMeatProducts and NumCatalogPurchases are also correlated together, suggesting that many customers purchase meat products from the catalogue and not in-store or website.

The only negatively correlated relationship is between Income and NumWebVisitsMonth. However, Income and NumWebPurchases are not negatively correlated. This indicates that customers with lower incomes are expected to visit the website more but make a similar number of purchases as their higher income customers.

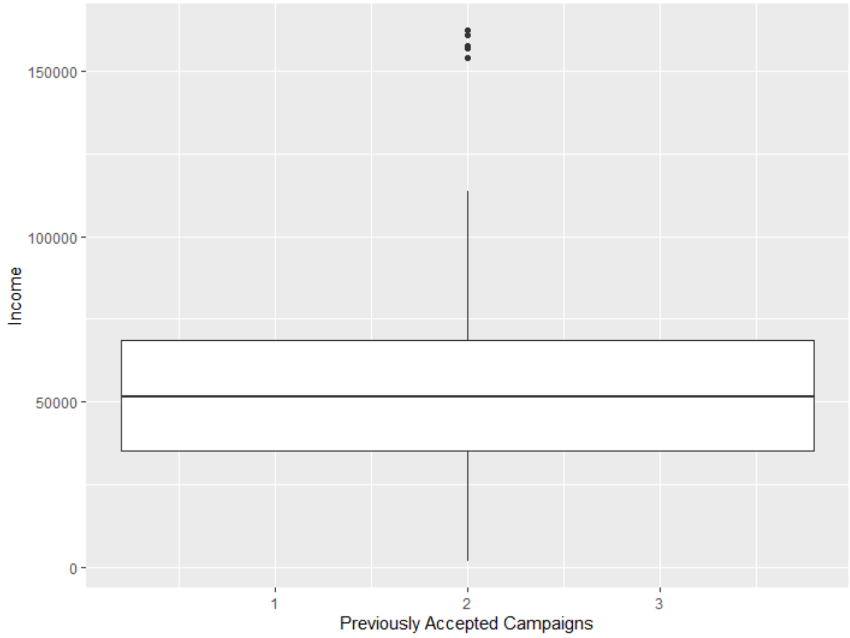
```
#income v/s mntspent
ggplot(df, aes(x = MntSpent, y = Income)) + geom_point() + geom_smooth(method = lm,color='red') +
  labs(x = 'Amount Spent ($)', y = 'Yearly Income ($)')
```



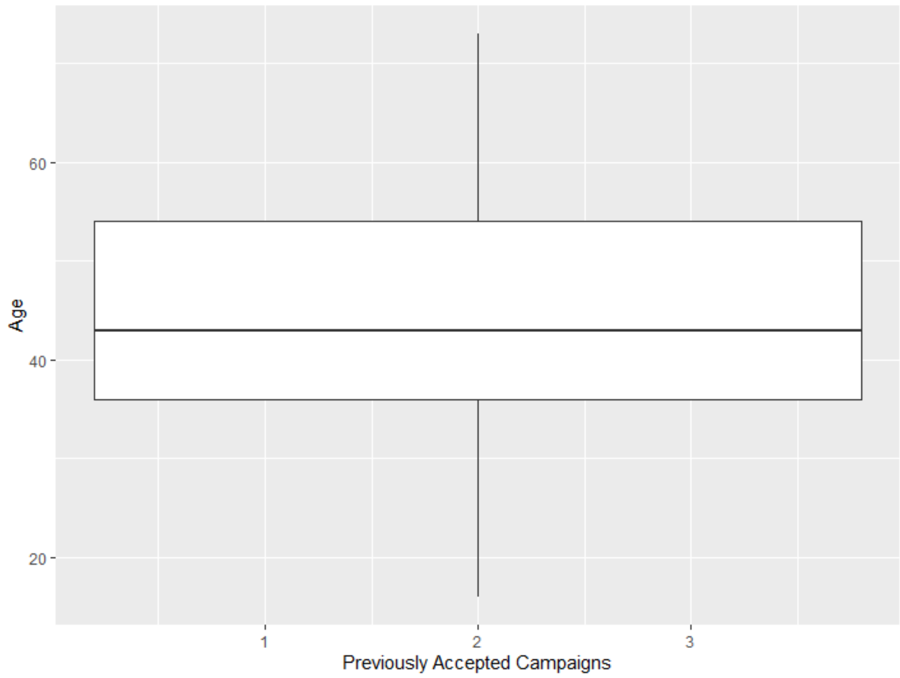

```
#income v/s age
ggplot(df, aes(x = NumWebVisitsMonth, y = Income)) + geom_point() +
geom_smooth(method = lm,color='red') +
  labs(x = 'Web Visits per Month', y = 'Yearly Income')
```



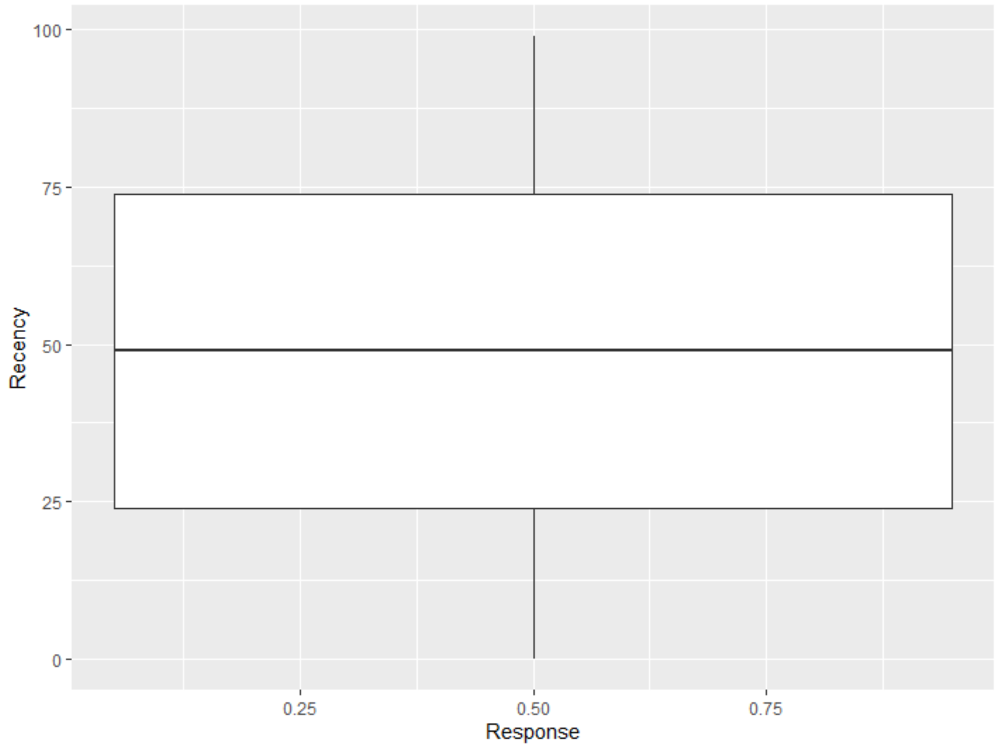
```
unique(df[c("AcceptedCmp")])

#boxplot Income by accepted previous
ggplot(df, aes(x = AcceptedCmp, y = Income)) + geom_boxplot()+
  labs(x = 'Previously Accepted Campaigns')
```




```
#boxplot Age by accepted previous
ggplot(df, aes(x = AcceptedCmp, y = Age)) + geom_boxplot()+
  labs(x = 'Previously Accepted Campaigns')
```

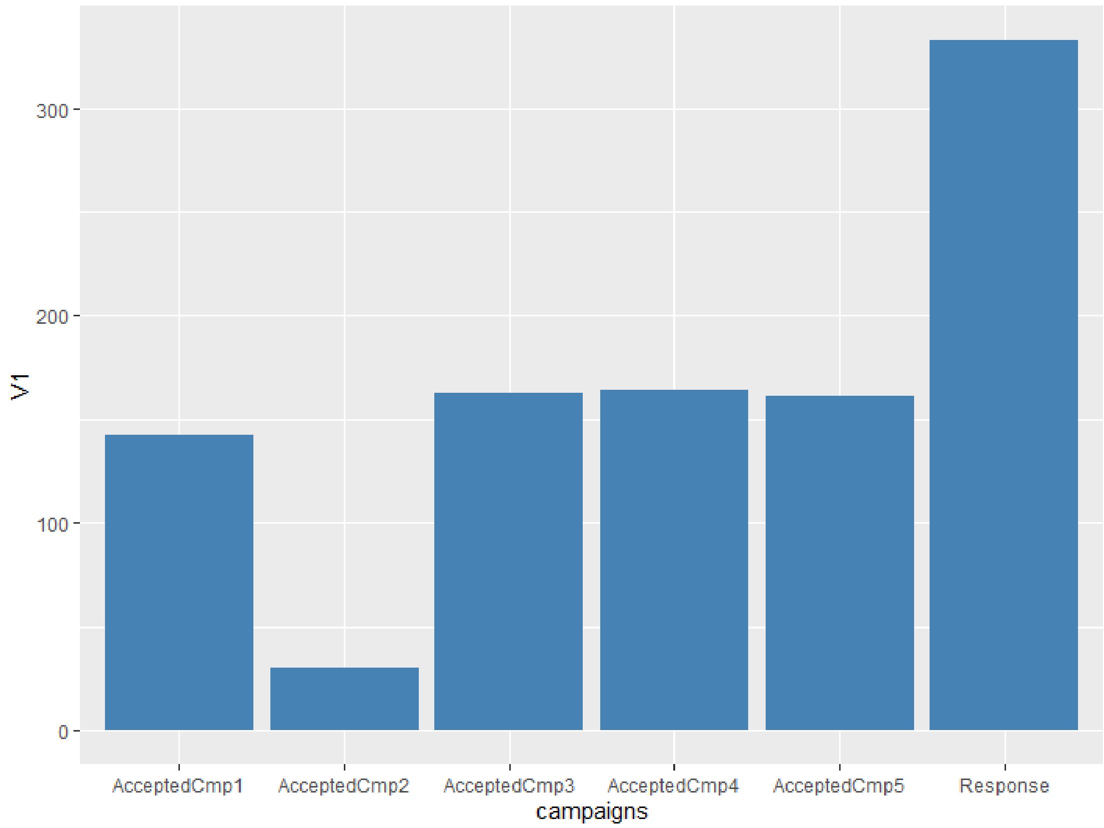


```
#boxplot Recency by Response
ggplot(df, aes(x = Response, y = Recency)) + geom_boxplot() +
  labs(x = 'Response', y = 'Recency')
```




```
#bar chart of most successful marketing campaign
campaigns <- c('AcceptedCmp1', 'AcceptedCmp2', 'AcceptedCmp3',
'AcceptedCmp4', 'AcceptedCmp5','Response')

campaign_df <- df %>%
  select(campaigns) %>%
  summarize_each(sum) %>%
  t() %>% #to calculate transpose of a matrix or Data Frame.
  as.data.frame() %>%
  rownames_to_column('campaigns')
campaign_df

ggplot(campaign_df,aes(x=campaigns, y=V1)) + geom_bar(fill="steelblue",stat =
"identity") + scale_y_continuous(labels = comma)
```

```
> campaign_df
      campaigns  V1
1 AcceptedCmp1 142
2 AcceptedCmp2  30
3 AcceptedCmp3 163
4 AcceptedCmp4 164
5 AcceptedCmp5 161
> ggplot(campaign_df,aes(x=campaigns, y=V1)) + geom_bar(fill="steelblue",stat = "identity") +
+   scale_y_continuous(labels = comma)
```



In this graph, the second campaign did the least well, failing to engage enough customers. Other previous campaigns have had similar engagement to one another, having been accepted by almost less customers.

Unlike the other campaigns, the current campaign has done wonders better, engaging most customers. The current campaign is the most successful while the second campaign was the least successful.

#Model Predictions

library(randomForest)

library(party)

library(datasets)

library(party)

```
library(dplyr)
library(magrittr)
library(e1071)
library(caTools)
library(class)
library(caret)

#Spliting Data into 70% and 30%
sample_data = sample.split(df, SplitRatio = 0.7)
train_data <- subset(df, sample_data == TRUE)
test_data <- subset(df, sample_data == FALSE)

# ID3 and Decision tree doesn't support categorical binary data, hence we use
Randomforest decison tree
set.seed(101)
rf <- randomForest(Response ~ ., data = train_data,importance=TRUE)
print(rf)
```

```
> rf <- randomForest(Response ~ ., data = train_data,importance=TRUE)
Warning message:
In randomForest.default(m, y, ...) :
  The response has five or fewer unique values.  Are you sure you want to do regression?
> print(rf)

Call:
 randomForest(formula = Response ~ ., data = train_data, importance = TRUE)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 9

          Mean of squared residuals: 0.08332432
                    % Var explained: 35.4
```

From the first call of randomForest, only about 34% of the variance in the data, which isn't even moderately good for gaining useful insights form this model. Additionally, only 9 variables where used to split the data, meaning that many of the variables are not useful in the randomForest model.

```
out.importance <- round(importance(rf), 2)
print(out.importance)
```

```
> out.importance <- round(importance(rf), 2)
> print(out.importance )
                    %IncMSE IncNodePurity
ï..ID                 -0.58          6.82
Year_Birth             7.26          5.78
Education             10.63          2.75
Income                18.83          9.89
Dt_Customer           19.30         14.73
Recency               31.23         18.47
MntWines              19.02          8.67
MntFruits              9.49          4.93
MntMeatProducts       21.21          9.09
MntFishProducts        8.81          5.24
MntSweetProducts      13.47          5.88
MntGoldProds          13.28          7.09
NumDealsPurchases      7.96          4.80
NumWebPurchases        7.63          3.05
NumCatalogPurchases   11.18          4.19
NumStorePurchases     16.28          5.18
NumWebVisitsMonth     14.77          7.26
AcceptedCmp3          15.19          5.33
AcceptedCmp4           2.84          0.62
AcceptedCmp5          14.75          7.51
AcceptedCmp1           9.46          3.12
AcceptedCmp2           7.36          0.81
Complain               1.00          0.01
Rel_Status            13.70          4.51
MntSpent              18.87          8.82
NumPurchases          12.43          4.09
AcceptedCmp           31.16         17.66
Age                    7.95          6.14
```

Importance – predicts the importance of all the predictor variables from the data frame.

#KNN Model

```
#KNN utilize Euclidean space so categorical variables will have to leave
train <- train_data %>% dplyr::select(where(is.numeric))
test <- test_data %>% dplyr::select(where(is.numeric))

train.knn <- as.data.frame(train)
test.knn <- as.data.frame(test)

# Fitting KNN Model
# to training dataset
kn <- knn(train.knn, test.knn, train.knn$Response, k = 1)
kn
misClassError <- mean(kn != test.knn$Response)
print(paste('Accuracy =', 1-misClassError))
```

```
> #KNN Model
> train <- train_data %>% dplyr::select(where(is.numeric))
> test <- test_data %>% dplyr::select(where(is.numeric))
>
> train.knn <- as.data.frame(train)
> test.knn <- as.data.frame(test)
>
> # Fitting KNN Model
> # to training dataset
> kn <- knn(train.knn, test.knn, train.knn$Response, k = 1)
> kn
  [1] 0 0 0 0 1 1 0 0 0 0 0 0 0 0 1 0 1 1 0 1 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
 [48] 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0
 [95] 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0
[142] 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 1 0 0 0 1 0 0 0 0
[189] 0 0 1 0 0 1 0 1 1 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
[236] 0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 0 0 1 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0
[283] 0 1 0 0 0 1 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0
[330] 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
[377] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 1 0 0 0 0 0 0
[424] 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
[471] 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
[518] 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 0 0 0
[565] 0 0 0 0 0 1 1 0 1 0 1 0 1 0 0 1 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 1 1 0 0 1 0 0 0 0 1 0 0 1 0 0
[612] 0 1 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 1 0 1 1 0 0 0 1 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 1 0 0 0 0
[659] 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0
Levels: 0 1
> misClassError <- mean(kn != test.knn$Response)
> print(paste('Accuracy =', 1-misClassError))
[1] "Accuracy = 0.758369723435226"
>
```

K = 5

kn <- knn(train.knn, test.knn, train.knn$Response, k = 5)

misClassError <- mean(kn != test.knn$Response)

print(paste('Accuracy =', 1-misClassError))

K = 15

kn <- knn(train.knn, test.knn, train.knn$Response, k = 15)

misClassError <- mean(kn != test.knn$Response)

print(paste('Accuracy =', 1-misClassError))

K = 25

kn <- knn(train.knn, test.knn, train.knn$Response, k = 25)

misClassError <- mean(kn != test.knn$Response)

print(paste('Accuracy =', 1-misClassError))

#K=25 reached the highest from the last four, hence increasing the value wont affect the accuracy much

#confusion matrix

cm <- table(test.knn$Response, kn)

cm

```
>
> # K = 5
> kn <- knn(train.knn, test.knn, train.knn$Response, k = 5)
> misClassError <- mean(kn != test.knn$Response)
> print(paste('Accuracy =', 1-misClassError))
[1] "Accuracy = 0.842794759825327"
>
> # K = 15
> kn <- knn(train.knn, test.knn, train.knn$Response, k = 15)
> misClassError <- mean(kn != test.knn$Response)
> print(paste('Accuracy =', 1-misClassError))
[1] "Accuracy = 0.851528384279476"
>
> # K = 25
> kn <- knn(train.knn, test.knn, train.knn$Response, k = 25)
> misClassError <- mean(kn != test.knn$Response)
> print(paste('Accuracy =', 1-misClassError))
[1] "Accuracy = 0.851528384279476"
>
> #K=25 reached the highest from the last four, hence increasin
ch
> #confusion matrix
> cm <- table(test.knn$Response, kn)
> cm
   kn
      0   1
  0 579   7
  1  95   6
```

```
> confusionMatrix(table(kn, test.knn$Response), positive = '1')
Confusion Matrix and Statistics


kn    0   1
  0 579  95
  1   7   6

               Accuracy : 0.8515
                 95% CI : (0.8227, 0.8773)
    No Information Rate : 0.853
    P-Value [Acc > NIR] : 0.569

                  Kappa : 0.0742

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.059406
            Specificity : 0.988055
         Pos Pred Value : 0.461538
         Neg Pred Value : 0.859050
             Prevalence : 0.147016
         Detection Rate : 0.008734
   Detection Prevalence : 0.018923
      Balanced Accuracy : 0.523730

       'Positive' Class : 1
```

The accuracy for this model is 85%, hence this is better option than randomForest. Many customers have not participated in the current campaign (about 85%) and therefore, classification methods will naturly favor classfying any given customer as not having participated.

Conclusion

- What does the Average customer look like for the company? - From the visualizations, the average customer is someone born around 1970 or before, who's earning around 50k/year and has a high school education. They probably enrolled in the company in July of 2013.

- What Products and Channels of Revenue are best performing? - Wines are vastly overperforming any other product, with Meat products coming in a far second while other products making up a relatively low proportion of the total revenue. The channels of revenue are all similar, with in-store only being slightly more popular than use of the catalogue or website purchasing venues.
- Which Marketing Campaigns were most successful? – The current marketing campaign is by far the most successful, while past campaigns all seem to have performed similarly (besides the 2nd).

- What factors contribute to the success of the current campaign? – The RandomForest model found that the most influential variables in its classification were the amount of campaigns that were previously accepted, the recency of purchases from our store, the date enrolled, and income.

- The KNN Model helps to draw the conclusions based on the customers and campaigns classifications in Clusters with k value of 25.
- Customers that participate in previous campaigns are more likely to participate in new ones.
- Customers with lower incomes are less likely to participate in the current and past campaigns.
- Customers that have been in our stores recently are more likely to have participated in our current campaign.

Given more context, it could make incredible decisions for the business and find out more better ways and insights to engage more customers to the marketing campaigns.

Name : Pranjal Gupta

Reg no. : 19BDS0081