Market Target analysis

The Geek Squad {Margaret Gathoni, Wilkister Mbaka, Griffin Buret, Gozzo Evrard Ded}

Table of Contents

Market Target Analysis 1
Introduction 1
Problem Statement 2
Objectives 2
Metrics Of success 2
Data Understanding 2
Data cleaning and Preparation 6
Exploratory Data Analysis 12
1. Uni-variate Analysis 12
2. Bivariate Analysis 22
3. Multivariate Analysis 32
Modeling 34
A. Pre-processing 35
B. Feature selection 37
C. Dealing with class Imbalance 38
KNN Classifier Model 39
Naive Bayes 42
SVM 43
Unsupervised Learning using K-Mean Clustering Method 44
Conclusion 45
Recommendation 45

Market Target Analysis

Introduction

Term deposits are a major source of income for a bank. A term deposit is a cash investment held at a financial institution. Your money is invested for an agreed rate of interest over a fixed amount of time, or term. The bank has various outreach plans to sell term deposits to their customers such as email marketing, advertisements, telephonic marketing, and digital marketing.

Problem Statement

Telephonic marketing campaigns still remain one of the most effective way to reach out to people. However, they require huge investment as large call centers are hired to actually execute these campaigns. Hence, it is crucial to identify the customers most likely to convert beforehand so that they can be specifically targeted via call. The data is related to direct marketing campaigns (phone calls) of a Portuguese banking institution.

Objectives

Main Objective

To build a model that predicts if the client will subscribe to a term deposit or not

Specific Objectives

1. To determine whether having a housing loan affected whether a client subscribed to a term deposit or not.
2. To find out if having a Personal loan affected whether a client subscribed to a term deposit or not.
3. To determine if a previous campaign success led to current campaign success to term deposit subscription.
4. To determine whether having credit on default affects term deposit subscription.
5. To determine if Multiple calls(campaign) contact led to a term deposit or not.

Metrics Of success

1. Define the question, the metric for success, the context, experimental design taken and the appropriateness of the available data to answer the given question.

2. Find and deal with outliers, anomalies, and missing data within the data set.

3. Perform EDA.

4. Building a model to predict if a client will subscribe to a term deposit or not (best model should have a Balanced Accuracy score above 80)

5. From our insights provide a conclusion and recommendation.

Data Understanding

Loading Important Libraries

```
library(data.table)
library(dplyr)
```

```
library(tidyverse)
library(ggplot2)
```

We have two sets of data set i.e train and test , will load them separately as follows:

a. Loading the train data set

```
library(readr)
train <- read_delim("train.csv", delim = ";",
    escape_double = FALSE, trim_ws = TRUE)
```

Previewing first six rows

```
head(train)

## # A tibble: 6 × 17
##     age job    marital educa…¹ default balance housing loan  contact   day
month
##   <dbl> <chr>  <chr>   <chr>   <chr>     <dbl> <chr>   <chr> <chr>   <dbl>
<chr>
## 1    58 manag… married tertia… no         2143 yes     no    unknown     5
may
## 2    44 techn… single  second… no           29 yes     no    unknown     5
may
## 3    33 entre… married second… no            2 yes     yes   unknown     5
may
## 4    47 blue-… married unknown no         1506 yes     no    unknown     5
may
## 5    33 unkno… single  unknown no            1 no      no    unknown     5
may
## 6    35 manag… married tertia… no          231 yes     no    unknown     5
may
## # … with 6 more variables: duration <dbl>, campaign <dbl>, pdays <dbl>,
## #   previous <dbl>, poutcome <chr>, y <chr>, and abbreviated variable name
## #   ¹education
## # ℹ Use `colnames()` to see all variable names
```

Checking number of rows and columns

```
dim(train)

## [1] 45211    17
```

We have 45211 rows and 17 columns

Checking the data types

```
str(train)

## spec_tbl_df [45,211 × 17] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ age      : num [1:45211] 58 44 33 47 33 35 28 42 58 43 ...
##  $ job      : chr [1:45211] "management" "technician" "entrepreneur"
"blue-collar" ...
```

```
##  $ marital  : chr [1:45211] "married" "single" "married" "married" ...
##  $ education: chr [1:45211] "tertiary" "secondary" "secondary" "unknown"
...
##  $ default  : chr [1:45211] "no" "no" "no" "no" ...
##  $ balance  : num [1:45211] 2143 29 2 1506 1 ...
##  $ housing  : chr [1:45211] "yes" "yes" "yes" "yes" ...
##  $ loan     : chr [1:45211] "no" "no" "yes" "no" ...
##  $ contact  : chr [1:45211] "unknown" "unknown" "unknown" "unknown" ...
##  $ day      : num [1:45211] 5 5 5 5 5 5 5 5 5 5 ...
##  $ month    : chr [1:45211] "may" "may" "may" "may" ...
##  $ duration : num [1:45211] 261 151 76 92 198 139 217 380 50 55 ...
##  $ campaign : num [1:45211] 1 1 1 1 1 1 1 1 1 1 ...
##  $ pdays    : num [1:45211] -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
##  $ previous : num [1:45211] 0 0 0 0 0 0 0 0 0 0 ...
##  $ poutcome : chr [1:45211] "unknown" "unknown" "unknown" "unknown" ...
##  $ y        : chr [1:45211] "no" "no" "no" "no" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   age = col_double(),
##   ..   job = col_character(),
##   ..   marital = col_character(),
##   ..   education = col_character(),
##   ..   default = col_character(),
##   ..   balance = col_double(),
##   ..   housing = col_character(),
##   ..   loan = col_character(),
##   ..   contact = col_character(),
##   ..   day = col_double(),
##   ..   month = col_character(),
##   ..   duration = col_double(),
##   ..   campaign = col_double(),
##   ..   pdays = col_double(),
##   ..   previous = col_double(),
##   ..   poutcome = col_character(),
##   ..   y = col_character()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

We have a mixture of numeric, and categorical variables

b. Loading the test data set

```
library(readr)
test <- read_delim("test.csv", delim = ";",
    escape_double = FALSE, trim_ws = TRUE)
```

Previewing the first six rows

```
head(test)
```

```
## # A tibble: 6 × 17
##     age job    marital educa…¹ default balance housing loan  contact   day
month
##   <dbl> <chr>  <chr>   <chr>   <chr>     <dbl> <chr>   <chr> <chr>   <dbl>
<chr>
## 1    30 unemp… married primary no         1787 no      no    cellul…    19
oct
## 2    33 servi… married second… no         4789 yes     yes   cellul…    11
may
## 3    35 manag… single  tertia… no         1350 yes     no    cellul…    16
apr
## 4    30 manag… married tertia… no         1476 yes     yes   unknown     3
jun
## 5    59 blue-… married second… no            0 yes     no    unknown     5
may
## 6    35 manag… single  tertia… no          747 no      no    cellul…    23
feb
## # … with 6 more variables: duration <dbl>, campaign <dbl>, pdays <dbl>,
## #   previous <dbl>, poutcome <chr>, y <chr>, and abbreviated variable name
## #   ¹education
## # ℹ Use `colnames()` to see all variable names
```

Checking the number of rows and columns

```
dim(test)

## [1] 4521   17
```

We have 17 columns and 4521 rows

Previewing our test data types

```
str(test)

## spec_tbl_df [4,521 × 17] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ age      : num [1:4521] 30 33 35 30 59 35 36 39 41 43 ...
##  $ job      : chr [1:4521] "unemployed" "services" "management"
"management" ...
##  $ marital  : chr [1:4521] "married" "married" "single" "married" ...
##  $ education: chr [1:4521] "primary" "secondary" "tertiary" "tertiary" ...
##  $ default  : chr [1:4521] "no" "no" "no" "no" ...
##  $ balance  : num [1:4521] 1787 4789 1350 1476 0 ...
##  $ housing  : chr [1:4521] "no" "yes" "yes" "yes" ...
##  $ loan     : chr [1:4521] "no" "yes" "no" "yes" ...
##  $ contact  : chr [1:4521] "cellular" "cellular" "cellular" "unknown" ...
##  $ day      : num [1:4521] 19 11 16 3 5 23 14 6 14 17 ...
##  $ month    : chr [1:4521] "oct" "may" "apr" "jun" ...
##  $ duration : num [1:4521] 79 220 185 199 226 141 341 151 57 313 ...
##  $ campaign : num [1:4521] 1 1 1 4 1 2 1 2 2 1 ...
##  $ pdays    : num [1:4521] -1 339 330 -1 -1 176 330 -1 -1 147 ...
##  $ previous : num [1:4521] 0 4 1 0 0 3 2 0 0 2 ...
```

```
##  $ poutcome : chr [1:4521] "unknown" "failure" "failure" "unknown" ...
##  $ y        : chr [1:4521] "no" "no" "no" "no" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   age = col_double(),
##   ..   job = col_character(),
##   ..   marital = col_character(),
##   ..   education = col_character(),
##   ..   default = col_character(),
##   ..   balance = col_double(),
##   ..   housing = col_character(),
##   ..   loan = col_character(),
##   ..   contact = col_character(),
##   ..   day = col_double(),
##   ..   month = col_character(),
##   ..   duration = col_double(),
##   ..   campaign = col_double(),
##   ..   pdays = col_double(),
##   ..   previous = col_double(),
##   ..   poutcome = col_character(),
##   ..   y = col_character()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

Data cleaning and Preparation

For cleaning will start cleaning the train data set

Train data set

a. Checking for null values

```
is.null(train)

## [1] FALSE

colSums(is.na(train))

##       age       job   marital education   default   balance   housing 
loan 
##         0         0         0         0         0         0         0 
0 
##   contact       day     month  duration  campaign     pdays  previous 
poutcome 
##         0         0         0         0         0         0         0 
0 
##         y 
##         0
```

We have no null values

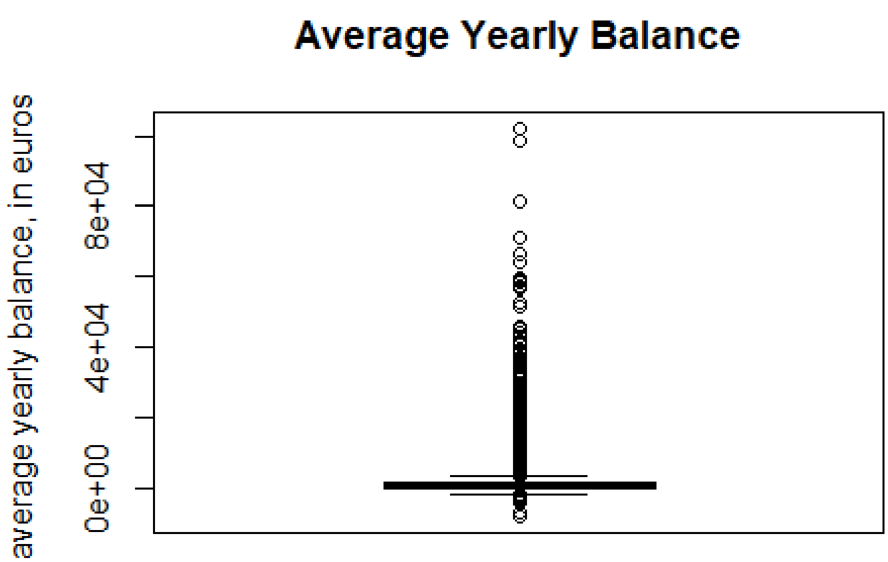
b. checking for duplicates

```
duplicated_rows <- train[duplicated(train),]
duplicated_rows

## # A tibble: 0 × 17
## # … with 17 variables: age <dbl>, job <chr>, marital <chr>, education
<chr>,
## #   default <chr>, balance <dbl>, housing <chr>, loan <chr>, contact
<chr>,
## #   day <dbl>, month <chr>, duration <dbl>, campaign <dbl>, pdays <dbl>,
## #   previous <dbl>, poutcome <chr>, y <chr>
## # ℹ Use `colnames()` to see all variable names
```

We have no duplicates

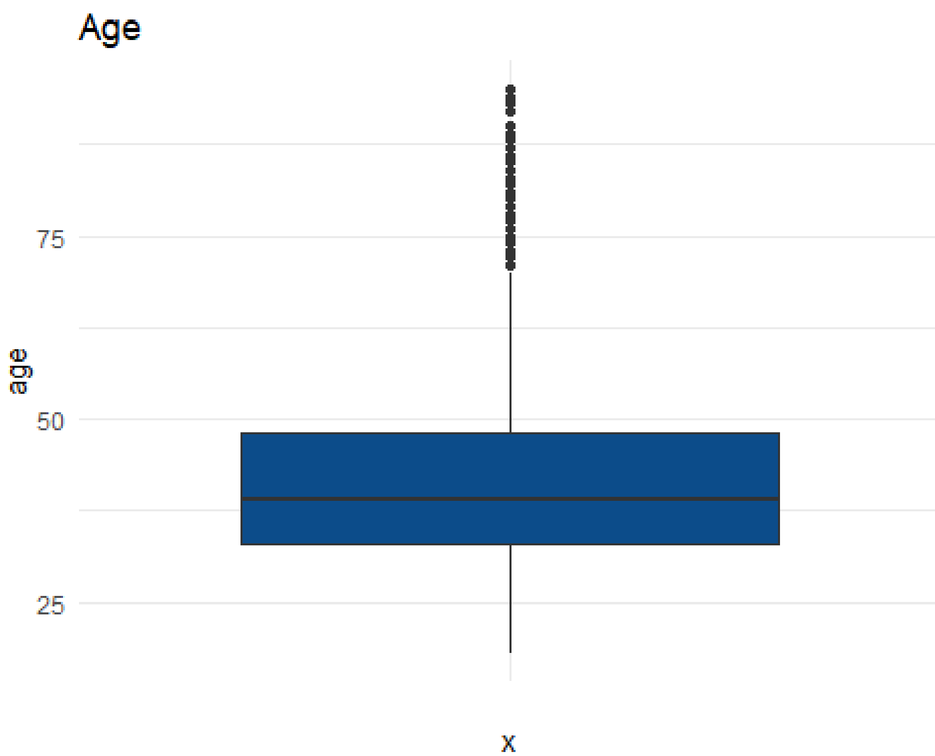
c. Checking for outliers

```
boxplot(train$balance, ylab = "average yearly balance, in euros ", main =
'Average Yearly Balance')
```



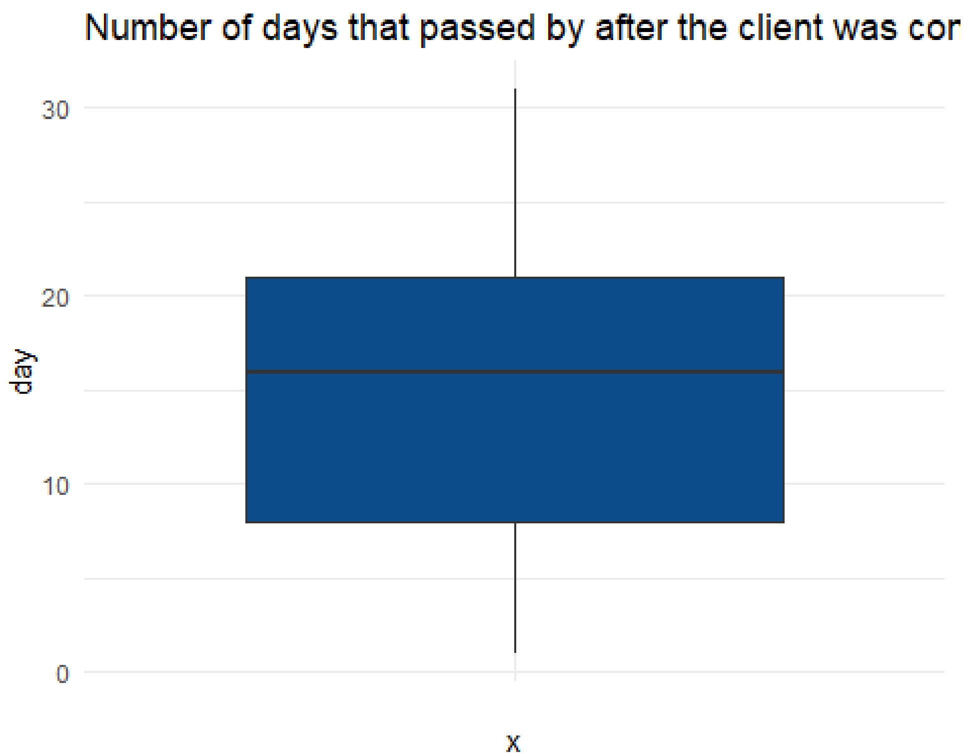
We have outliers on the balance column.

```
ggplot(train) +
  aes(x = "", y =age) +
  geom_boxplot(fill = "#0c4c8a") +
  theme_minimal() + labs(title = 'Age')
```



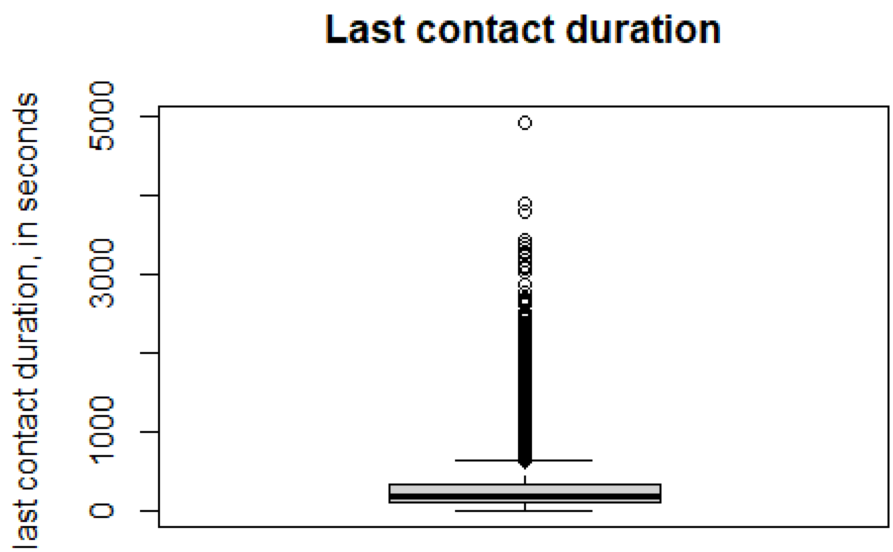
We have outlier in age

```
ggplot(train) +
  aes(x = "", y =day) +
  geom_boxplot(fill = "#0c4c8a") +
  theme_minimal() + labs(title = 'Number of days that passed by after the client was contacted from previous campaign')
```

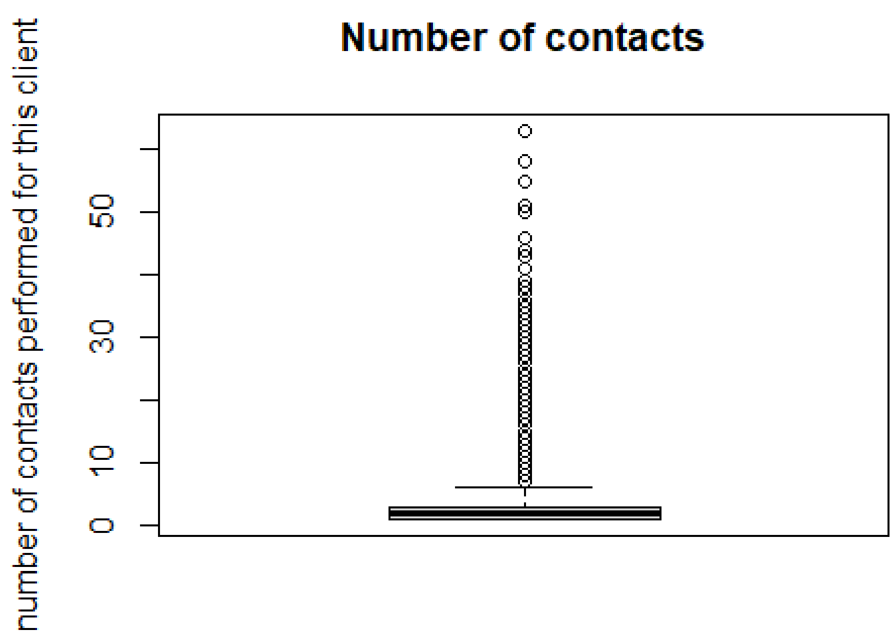


There are no outliers on day column.

```
boxplot(train$duration, ylab = "last contact duration, in seconds ", main =
'Last contact duration')
```




```
boxplot(train$campaign, ylab = "number of contacts performed for this client", main = 'Number of contacts')
```



Most of the numeric columns have outliers but will not drop them since they are significant for our analysis.

Test data set

a. checking for null values

```
is.null(test)

## [1] FALSE

colSums(is.na(test))

##       age       job   marital education   default   balance   housing 
loan 
##         0         0         0         0         0         0         0 
0 
##   contact       day     month  duration  campaign     pdays  previous 
poutcome 
##         0         0         0         0         0         0         0 
0 
##         y 
##         0
```

We have no null values

b. checking for duplicates

```
duplicated_rows <- test[duplicated(test),]
duplicated_rows

## # A tibble: 0 × 17
## # … with 17 variables: age <dbl>, job <chr>, marital <chr>, education 
<chr>,
## #   default <chr>, balance <dbl>, housing <chr>, loan <chr>, contact 
<chr>,
## #   day <dbl>, month <chr>, duration <dbl>, campaign <dbl>, pdays <dbl>,
## #   previous <dbl>, poutcome <chr>, y <chr>
## # ℹ Use `colnames()` to see all variable names
```

We have no duplicates

Will combine the two tables for EDA

```
df <- rbind(train, test)
head(df)

## # A tibble: 6 × 17
##     age job    marital educa…¹ default balance housing loan  contact   day 
month
##   <dbl> <chr>  <chr>   <chr>   <chr>     <dbl> <chr>   <chr> <chr>   <dbl> 
<chr>
## 1    58 manag… married tertia… no         2143 yes     no    unknown     5 
may
```

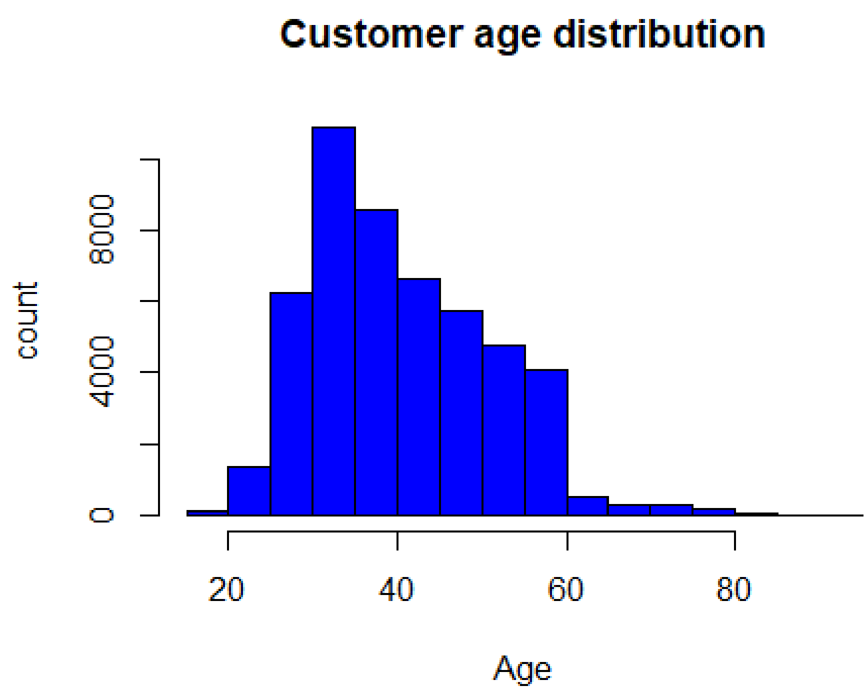
```
## 2     44 techn… single   second… no             29 yes      no     unknown      5
may
## 3     33 entre… married second… no              2 yes      yes    unknown      5
may
## 4     47 blue-… married unknown no           1506 yes      no     unknown      5
may
## 5     33 unkno… single   unknown no              1 no       no     unknown      5
may
## 6     35 manag… married tertia… no            231 yes      no     unknown      5
may
## # … with 6 more variables: duration <dbl>, campaign <dbl>, pdays <dbl>,
## #   previous <dbl>, poutcome <chr>, y <chr>, and abbreviated variable name
## #   ¹education
## # ℹ Use `colnames()` to see all variable names
```

Exploratory Data Analysis

1. Uni-variate Analysis

Age distribution of the customers

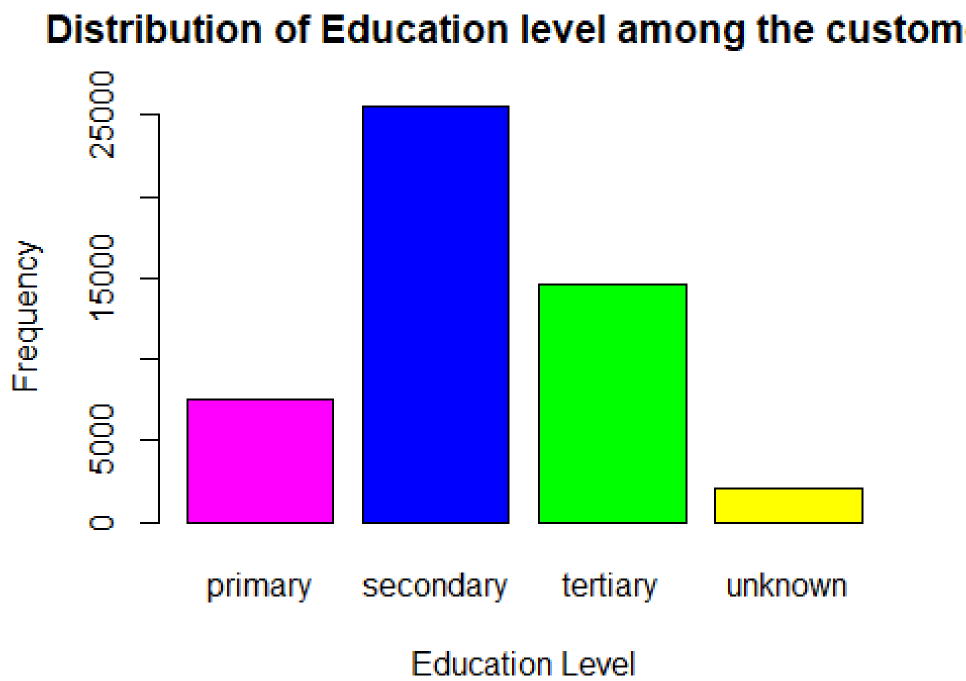
```
hist((df$age),
main = "Customer age distribution",
     xlab = 'Age',
     ylab = 'count',
     col = "blue")
```



The age bracket of most clients was 35 years, there was an extreme of 95 years and 18 years

Education level Distribution of the customers

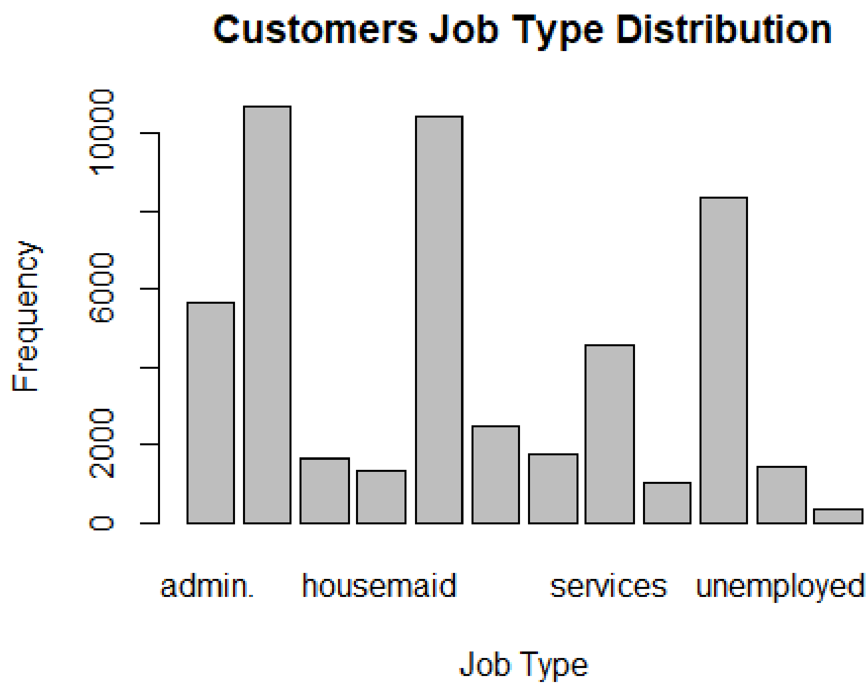
```
edu <- (df$education)
edu.frequency <- table(edu)
edu.frequency

## edu
##   primary secondary  tertiary   unknown
##      7529     25508     14651      2044

barplot(edu.frequency,
  main="Distribution of Education level among the customer",
  xlab="Education Level",
  ylab = "Frequency",
  col=c("magenta","blue", "green", "yellow"),
  )
```



Most of our customers had a form of education with highest having already reached secondary education followed by tertiary level and the least were those who did not disclose their level of education.

Job types distribution

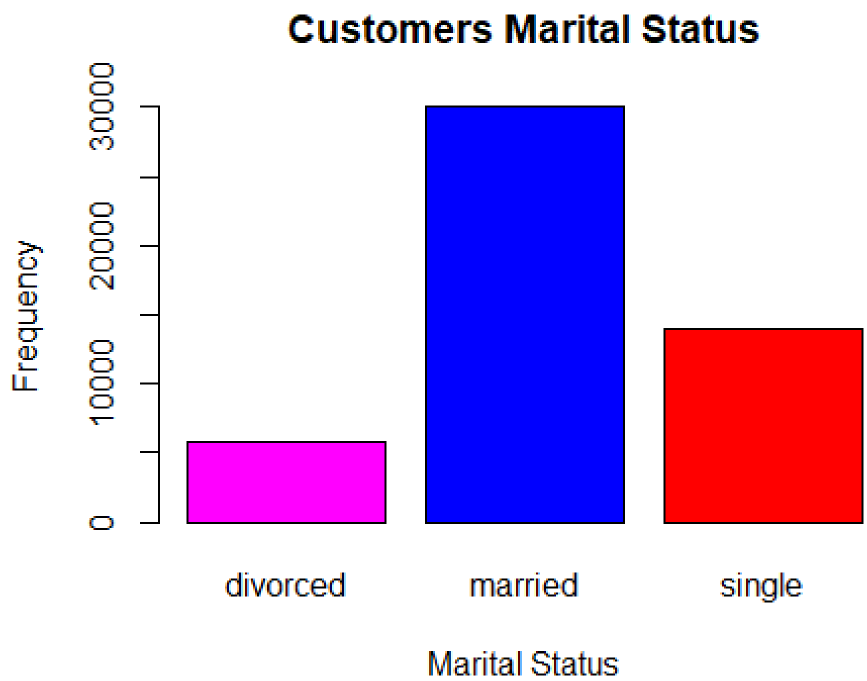
```
job <- (df$job)
job.frequency <- table(job)
job.frequency

## job
##        admin.   blue-collar  entrepreneur     housemaid    management
##          5649         10678          1655          1352         10427
##       retired self-employed      services       student    technician
##          2494          1762          4571          1022          8365
##    unemployed       unknown
##          1431           326

barplot(job.frequency,
  main="Customers Job Type Distribution",
  xlab="Job Type",
  ylab = "Frequency")
```



The clients for the campaign involved most personnel working in blue collar jobs, management and administrative levels with the least being students and thosw who didn't disclose their jobs.

Marital status

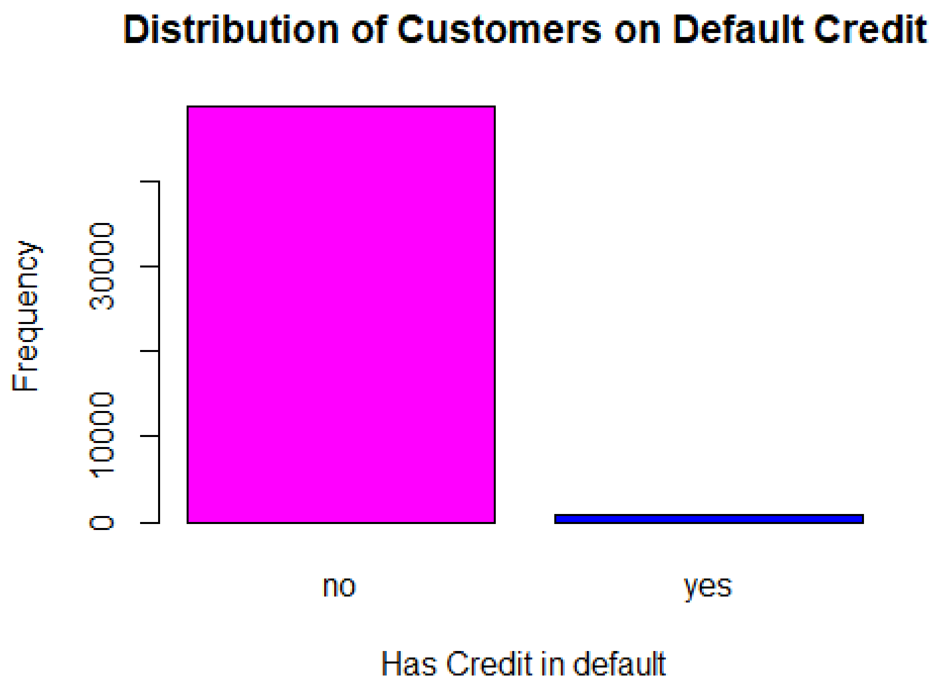
```
marital <- (df$marital)
marital.frequency <- table(marital)
marital.frequency

## marital
## divorced  married   single
##     5735    30011    13986

barplot(marital.frequency,
  main="Customers Marital Status",
  xlab="Marital Status",
  ylab = "Frequency",
  col=c("magenta","blue", "red"),
  )
```



Most of the customers participating in the campaigns were married, followed by single people and finally divorced.

Credit status

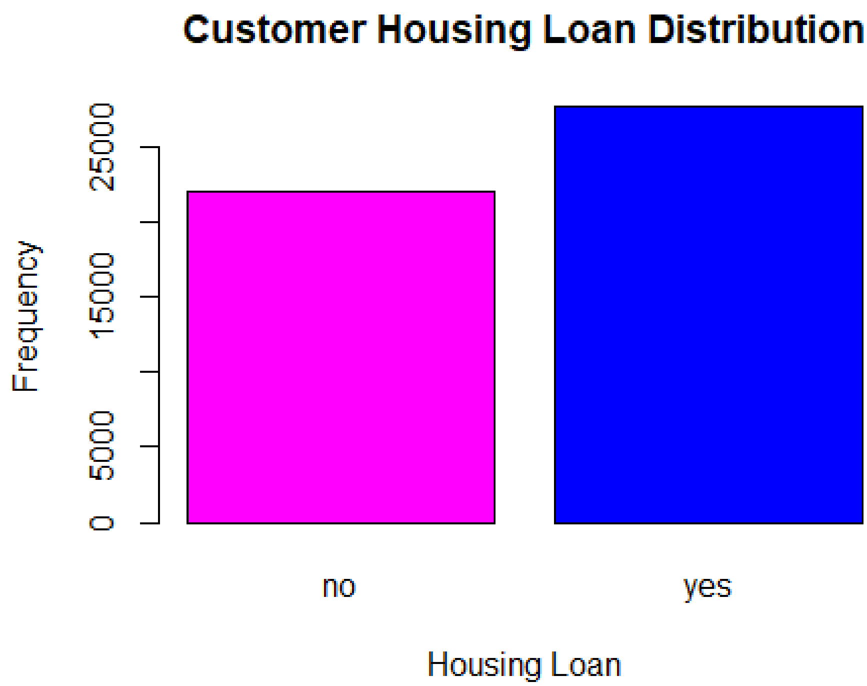
```
default <- (df$default)
default.frequency <- table(default)
default.frequency

## default
##    no   yes
## 48841   891

barplot(default.frequency,
  main="Distribution of Customers on Default Credit",
  xlab="Has Credit in default",
  ylab = "Frequency",
  col=c("magenta","blue"),
  )
```



The graph above shows that most customers don't have credit on default.

Housing Loan

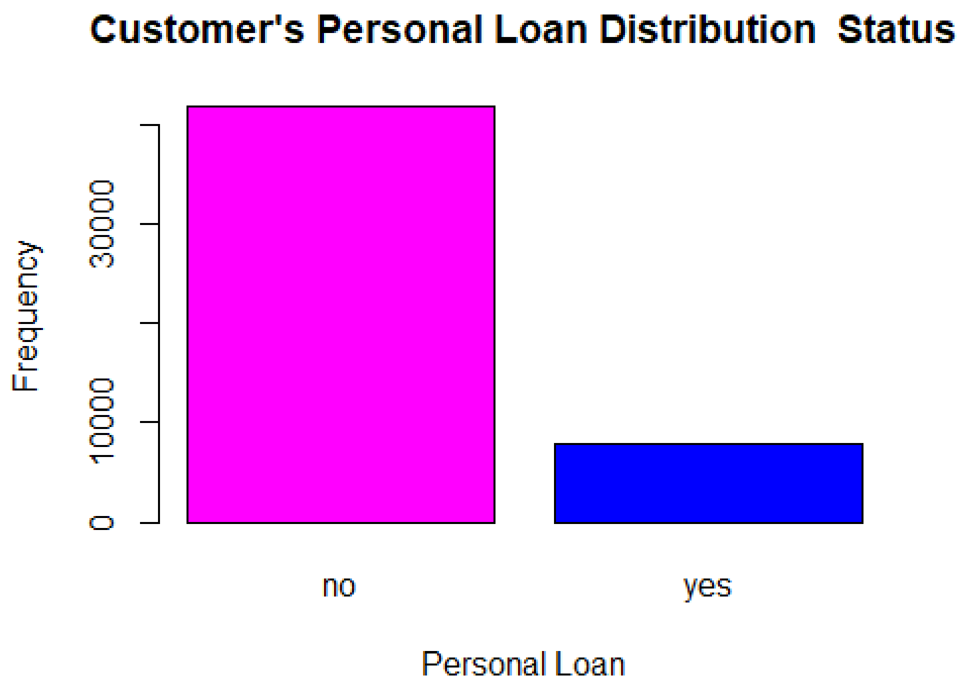
```
housing <- (df$housing)
housing.frequency <- table(housing)
housing.frequency

## housing
##    no   yes
## 22043 27689

barplot(housing.frequency,
  main="Customer Housing Loan Distribution",
  xlab="Housing Loan",
  ylab = "Frequency",
  col=c("magenta","blue"),
  )
```



Most of the customers have a housing loan.

Personal loan

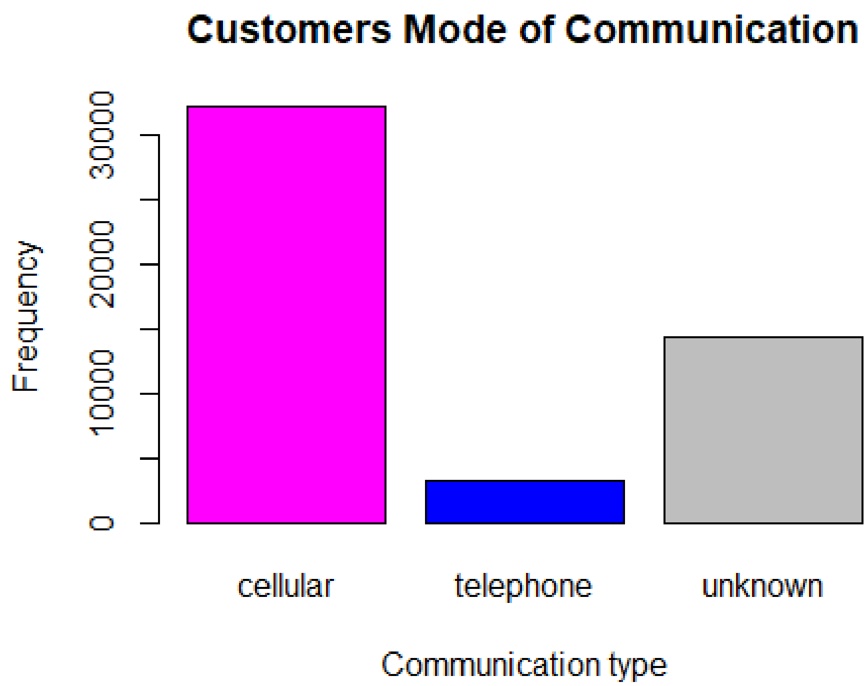
```
loan <- (df$loan)
loan.frequency <- table(loan)
loan.frequency

## loan
##    no   yes
## 41797  7935

barplot(loan.frequency,
  main="Customer's Personal Loan Distribution  Status",
  xlab="Personal Loan",
  ylab = "Frequency",
  col=c("magenta","blue"),
  )
```



Most customers don't have a personal loan.

Communication type

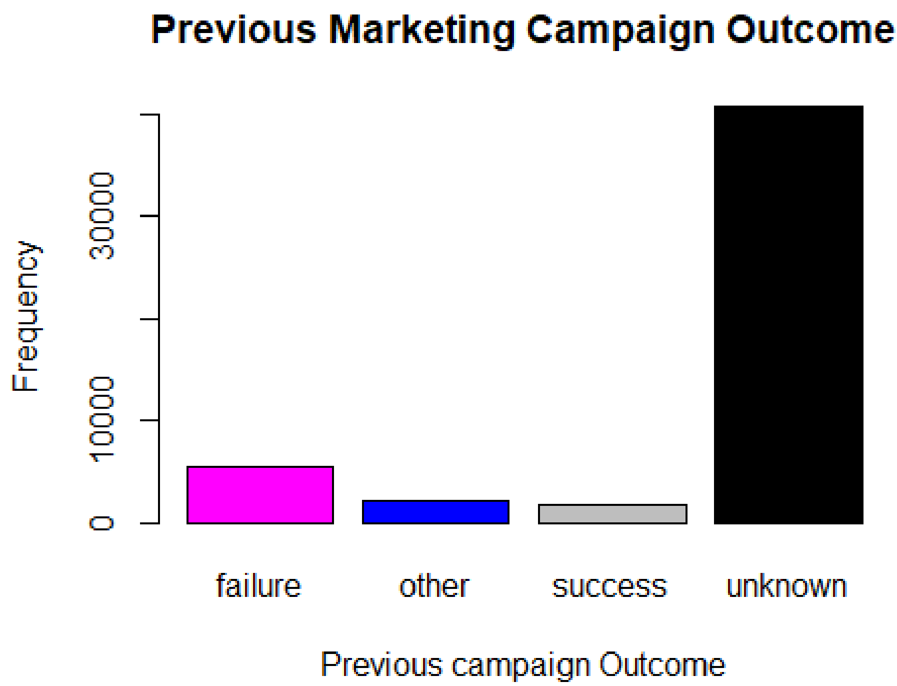
```
contact <- (df$contact)
contact.frequency <- table(contact)
contact.frequency

## contact
##  cellular telephone   unknown
##     32181      3207     14344

barplot(contact.frequency,
  main="Customers Mode of Communication",
  xlab="Communication type",
  ylab = "Frequency",
  col=c("magenta","blue", "grey"),
  )
```



The marketing team contacted most of the customers via cellphone.

Outcome of the previous marketing campaign

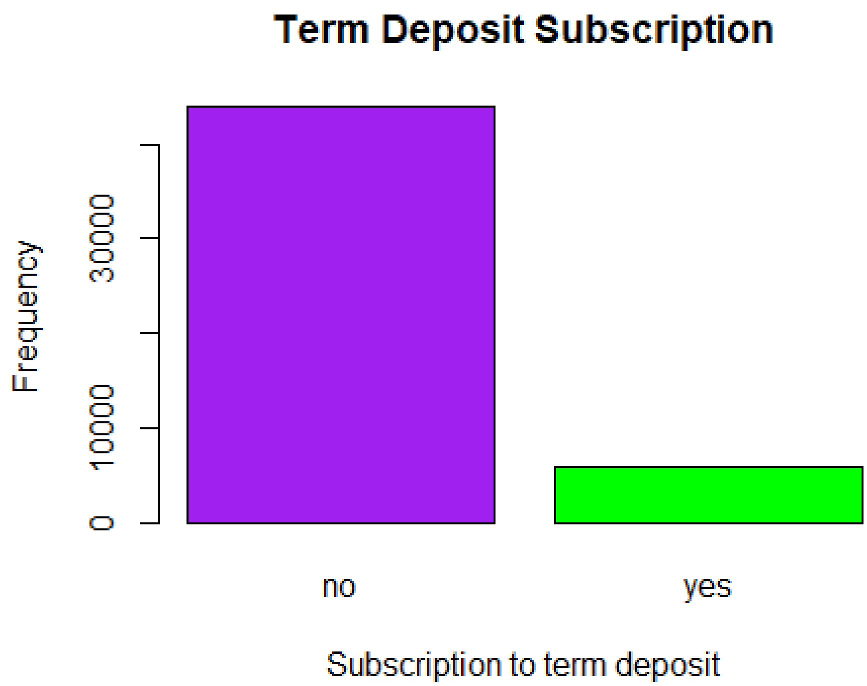
```
outcome <- (df$poutcome)
outcome.frequency <- table(outcome)
outcome.frequency

## outcome
## failure   other success unknown
##    5391    2037    1640   40664

barplot(outcome.frequency,
  main="Previous Marketing Campaign Outcome",
  xlab="Previous campaign Outcome",
  ylab = "Frequency",
  col=c("magenta","blue", "grey", "black"),
  )
```



The graph shows most customers outcome of the previous marketing campaign to be unknown, with the least of the current focus group ending in success

Subscription to term deposit

```
sb <- (df$y)
sb.frequency <- table(sb)
sb.frequency

## sb
##    no   yes
## 43922  5810

barplot(sb.frequency,
  main="Term Deposit Subscription",
  xlab="Subscription to term deposit",
  ylab = "Frequency",
  col=c("Purple","green"),
  )
```



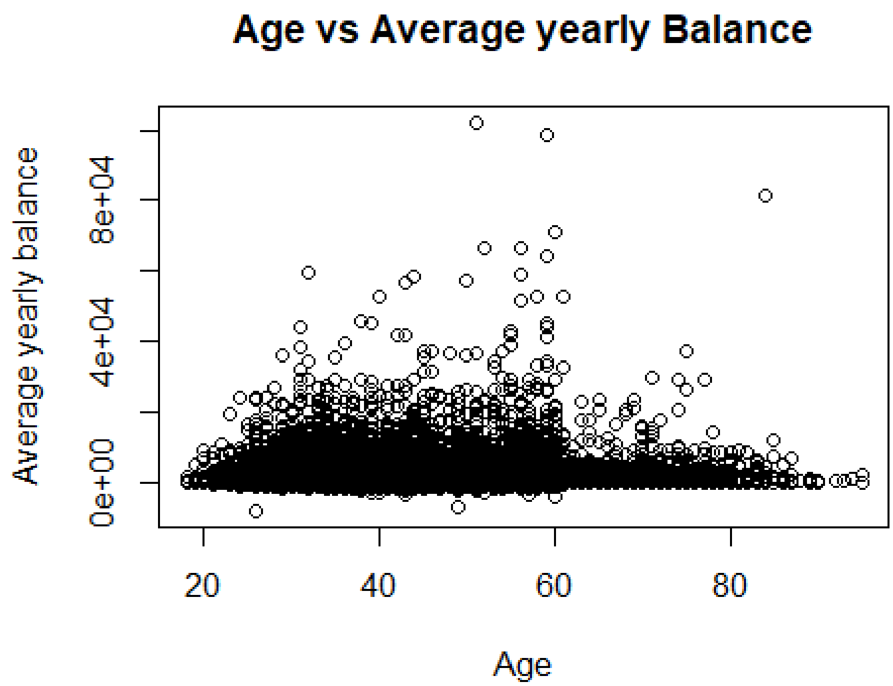
The graph shows the outcome towards term deposit subscription where most customers did not subscribe.

2. Bivariate Analysis

```
library(reshape2)
```

Comparing age vs average yearly balance

```
plot((df$age), (df$balance),
     main = "Age vs Average yearly Balance",
     xlab = 'Age',
     ylab = 'Average yearly balance')
```



There is high concentration of average yearly balance of most customers despite age to be on the lower limit, however, around age 40 to 60 years we have outliers on the upper limit.

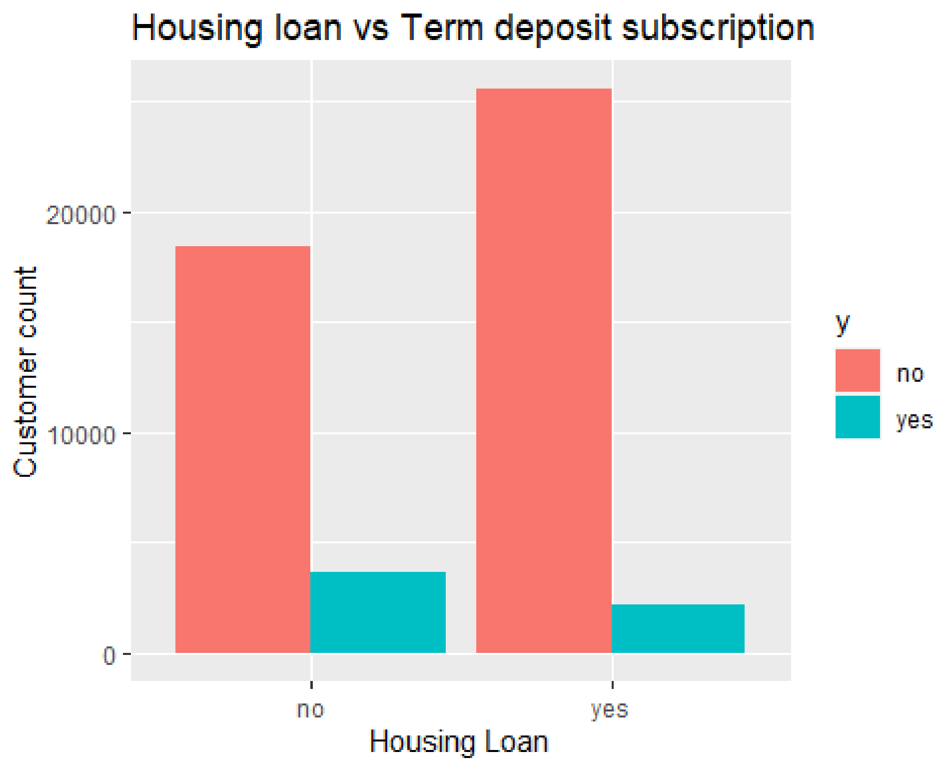
Does having a housing loan affect whether a client subscribed to a term deposit or not?

```
library(plyr)
counts <- ddply(df, .(df$y, df$housing), nrow)
names(counts) <- c("term deposit", "housing loan", "Freq")
counts

##   term deposit housing loan  Freq
## 1           no           no 18388
## 2           no          yes 25534
## 3          yes           no  3655
## 4          yes          yes  2155
```

The table shows that most people with housing loan didn't no subscribe to a term deposit.

We can see this visually

```
ggplot(df, aes(fill=y, x=housing)) + geom_bar(position = "dodge" ) +
labs(title = 'Housing loan vs Term deposit subscription',
    x = 'Housing Loan', y = 'Customer count')
```



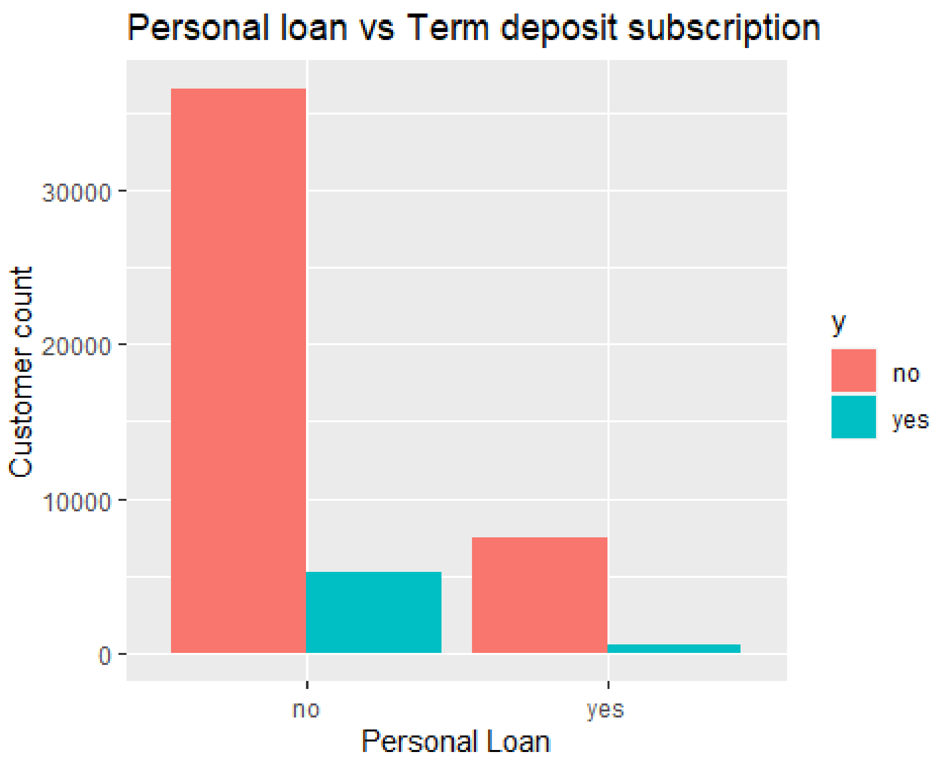
We can therefore answer our objective that indeed having a housing loan affects if someone subscribes to a term deposit or not. We can clearly see most of the people who subscribed to a term deposit did not have a housing loan.

Does having a Personal loan affect whether a client subscribed to a term deposit or not?

```
loan_counts <- ddply(df, .(df$y, df$loan), nrow)
names(loan_counts) <- c("Term deposit", "Personal loan", "Freq")
loan_counts

##   Term deposit Personal loan  Freq
## 1           no            no 36514
## 2           no           yes  7408
## 3          yes            no  5283
## 4          yes           yes   527
```

The table shows that most people with personal loan did not subscribe to a term deposit.

```
ggplot(df, aes(fill=y, x=loan)) + geom_bar(position = "dodge" ) + labs(title
= 'Personal loan vs Term deposit subscription',
    x = 'Personal Loan', y = 'Customer count')
```



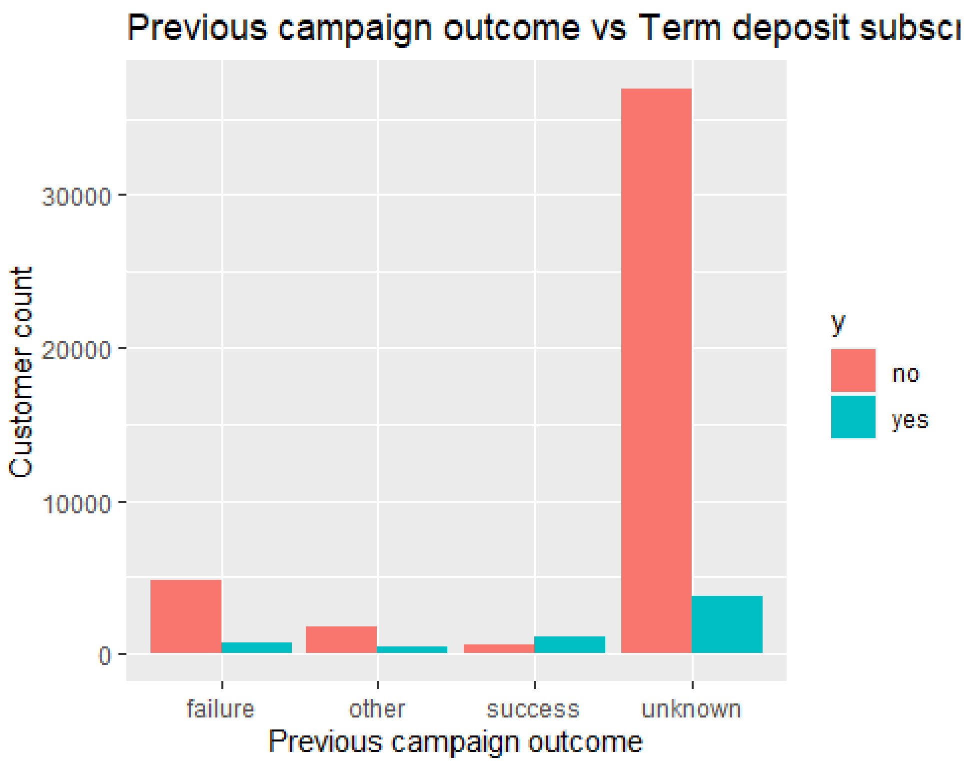
We can therefore answer our objective that indeed having a personal loan affects if someone subscribes to a term deposit or not. We can clearly see most of the people who subscribed to a term deposit did not have a personal loan.

Does previous campaign success lead to current campaign success to term deposit subscription?

```
previous_outcome <- ddply(df, .(df$y, df$poutcome), nrow)
names(previous_outcome) <- c("Term deposit", "Previous outcome", "Freq")
previous_outcome

##   Term deposit Previous outcome  Freq
## 1           no          failure  4710
## 2           no            other  1692
## 3           no          success   579
## 4           no          unknown 36941
## 5          yes          failure   681
## 6          yes            other   345
## 7          yes          success  1061
## 8          yes          unknown  3723
```

From this table we can see previous success indeed lead to current success.

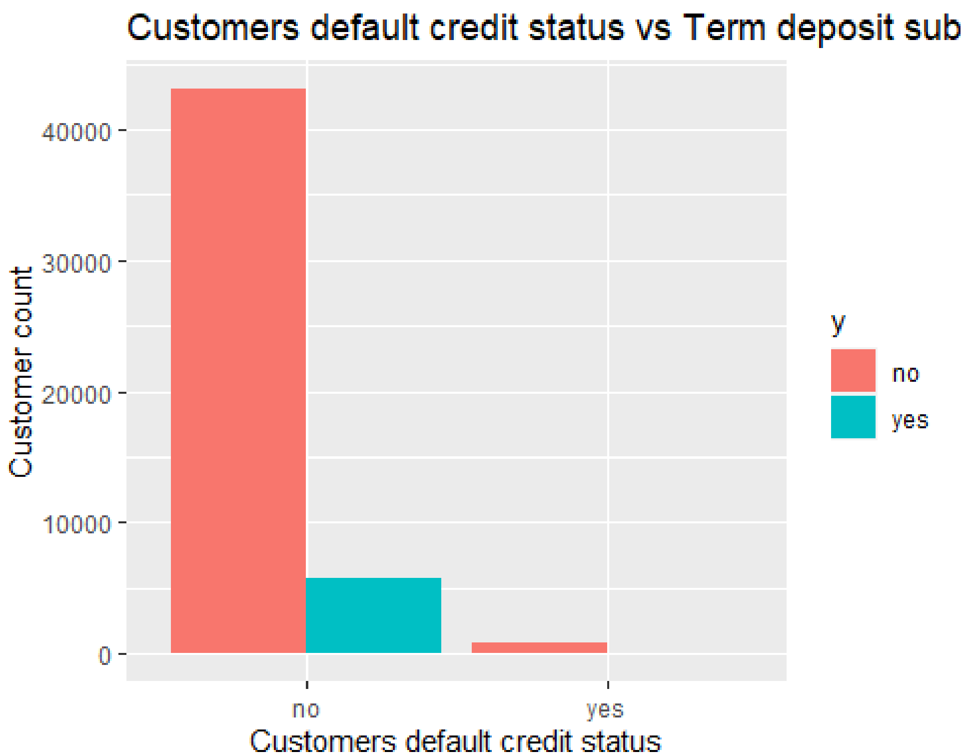
```
ggplot(df, aes(fill=y, x=poutcome)) + geom_bar(position = "dodge" ) +
labs(title = 'Previous campaign outcome vs Term deposit subscription',
    x = 'Previous campaign outcome', y = 'Customer count')
```



The success of previous campaign had a higher chance of success to the current campaign.

Does having credit on default affect term deposit subscription?

```
default_count <- ddply(df, .(df$y, df$default), nrow)
names(default_count) <- c("term deposit", "Credit by Default", "Freq")
default_count

##   term deposit Credit by Default  Freq
## 1           no                no 43092
## 2           no               yes   830
## 3          yes                no  5749
## 4          yes               yes    61

ggplot(df, aes(fill=y, x=default)) + geom_bar(position = "dodge" ) +
labs(title = 'Customers default credit status vs Term deposit subscription',
    x = 'Customers default credit status', y = 'Customer count')
```



The graph and table above shows having a credit on default doesn't lead to term deposit subscription.

Job type vs Term deposit subscription

```
job_count <- ddply(df, .(df$job, df$y), nrow)
names(job_count) <- c("Job type", "term deposit", "Freq")
job_count

##          Job type term deposit Freq
## 1          admin.           no 4960
## 2          admin.          yes  689
## 3     blue-collar           no 9901
## 4     blue-collar          yes  777
## 5    entrepreneur           no 1517
## 6    entrepreneur          yes  138
## 7       housemaid           no 1229
## 8       housemaid          yes  123
## 9      management           no 8995
## 10     management          yes 1432
## 11        retired           no 1924
## 12        retired          yes  570
## 13  self-employed           no 1555
## 14  self-employed          yes  207
## 15       services           no 4164
## 16       services          yes  407
## 17        student           no  734
```

```
## 18       student          yes  288
## 19    technician           no 7442
## 20    technician          yes  923
## 21    unemployed           no 1216
## 22    unemployed          yes  215
## 23       unknown           no  285
## 24       unknown          yes   41
```

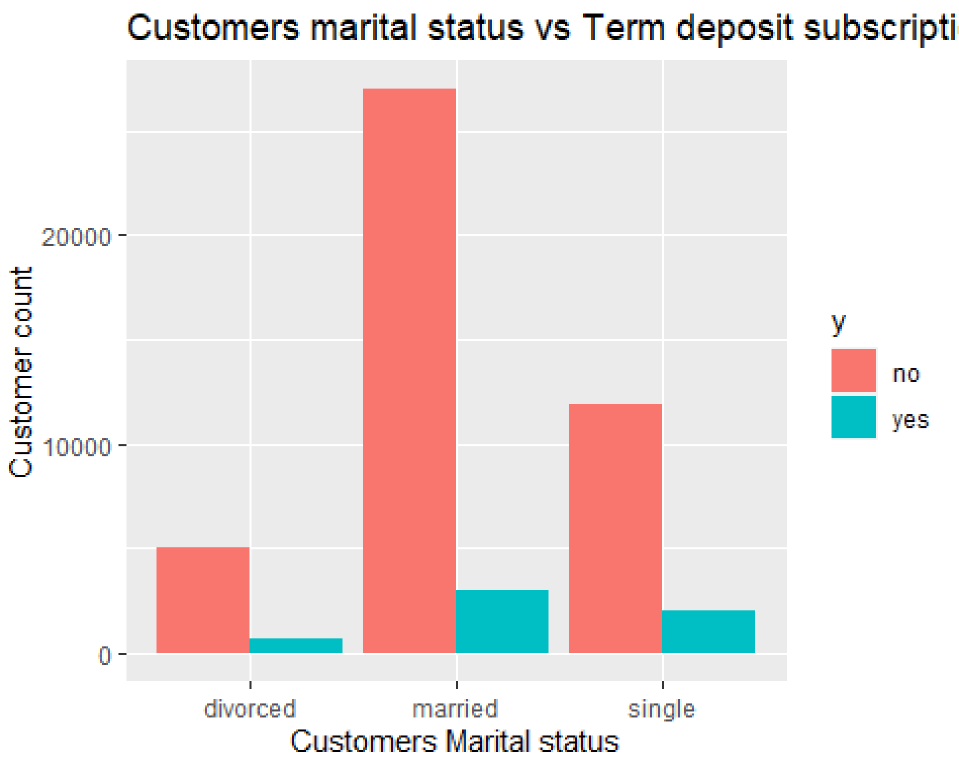
The table above shows that most people in management subscribed to a term deposit, followed by blue collar and administrative.

Marital status vs Term deposit subscription

```
maritalstatus <- ddply(df, .(df$marital, df$y), nrow)
names(maritalstatus) <- c("maritalstatus", "Term Deposit", "Freq")
maritalstatus
```

```
##   maritalstatus Term Deposit  Freq
## 1      divorced           no  5036
## 2      divorced          yes   699
## 3       married           no 26979
## 4       married          yes  3032
## 5        single           no 11907
## 6        single          yes  2079
```

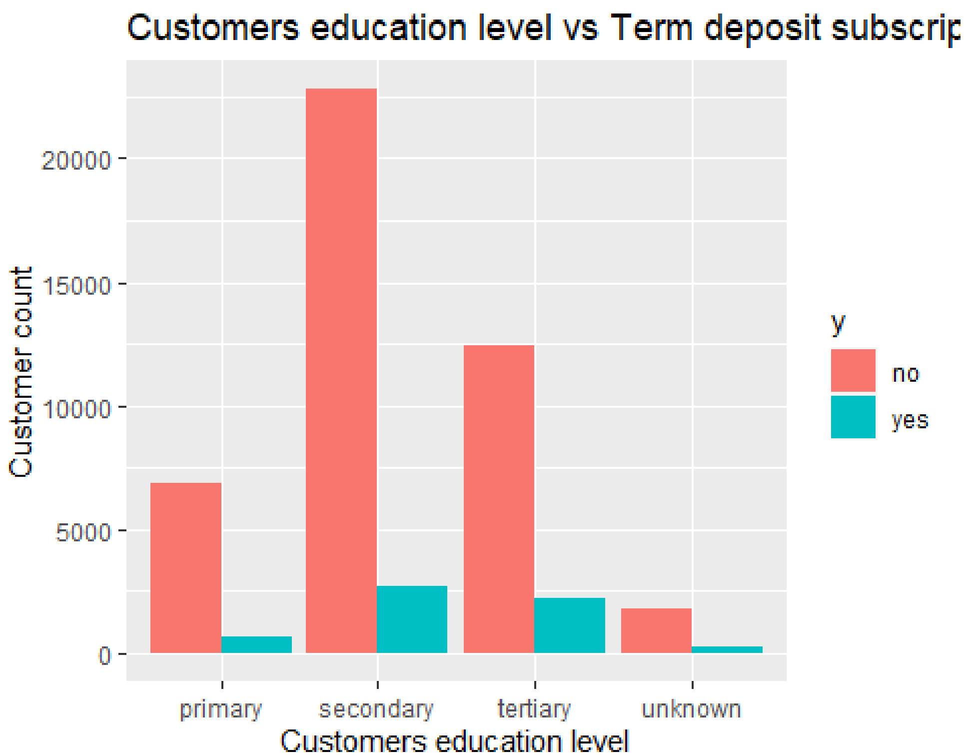
```
ggplot(df, aes(fill=y, x=marital)) + geom_bar(position = "dodge" ) +
labs(title = 'Customers marital status vs Term deposit subscription',
    x = 'Customers Marital status', y = 'Customer count')
```



Most married people subscribed to term deposit, however, they were also the majority in the campaign.

Education Level vs Term deposit subscription

```
edu_count<- ddply(df, .(df$education, df$y), nrow)
names(edu_count) <- c("Education level", "term deposit", "Freq")
edu_count

##   Education level term deposit  Freq
## 1         primary           no  6874
## 2         primary          yes   655
## 3       secondary           no 22813
## 4       secondary          yes  2695
## 5        tertiary           no 12462
## 6        tertiary          yes  2189
## 7         unknown           no  1773
## 8         unknown          yes   271

ggplot(df, aes(fill=y, x=education)) + geom_bar(position = "dodge" )
+labs(title = 'Customers education level vs Term deposit subscription',
    x = 'Customers education level', y = 'Customer count')
```



The graph above shows most customers as previously observed had some level of secondary education. However, proportionally most tertiary educational holder actually subscribed to term deposit compared to other levels of education.

Multiple calls(campaign) contact led to a term deposit or not?

```
campaign_count<- ddply(df, .(df$campaign, df$y), nrow)
names(campaign_count) <- c("Campaign", "term deposit", "Freq")
campaign_count

##    Campaign term deposit  Freq
## 1         1           no 16477
## 2         1          yes  2801
## 3         2           no 12230
## 4         2          yes  1539
## 5         3           no  5404
## 6         3          yes   675
## 7         4           no  3487
## 8         4          yes   360
## 9         5           no  1783
## 10        5          yes   148
## 11        6           no  1338
## 12        6          yes   108
## 13        7           no   757
## 14        7          yes    53
## 15        8           no   560
## 16        8          yes    36
```

```
## 17         9           no   334
## 18         9          yes    23
## 19        10           no   278
## 20        10          yes    15
## 21        11           no   207
## 22        11          yes    16
## 23        12           no   171
## 24        12          yes     5
## 25        13           no   142
## 26        13          yes     8
## 27        14           no    99
## 28        14          yes     4
## 29        15           no    89
## 30        15          yes     4
## 31        16           no    85
## 32        16          yes     2
## 33        17           no    69
## 34        17          yes     7
## 35        18           no    58
## 36        19           no    47
## 37        20           no    45
## 38        20          yes     1
## 39        21           no    36
## 40        21          yes     1
## 41        22           no    25
## 42        23           no    24
## 43        24           no    21
## 44        24          yes     2
## 45        25           no    26
## 46        26           no    13
## 47        27           no    10
## 48        28           no    19
## 49        29           no    16
## 50        29          yes     1
## 51        30           no     9
## 52        31           no    13
## 53        32           no    10
## 54        32          yes     1
## 55        33           no     6
## 56        34           no     5
## 57        35           no     4
## 58        36           no     4
## 59        37           no     2
## 60        38           no     3
## 61        39           no     1
## 62        41           no     2
## 63        43           no     3
## 64        44           no     2
## 65        46           no     1
## 66        50           no     3
```

```
## 67        51           no      1
## 68        55           no      1
## 69        58           no      1
## 70        63           no      1
```

The table above shows that multiple contact during the campaign did not result to subscription. Most the people who actually subscribed to term deposit were only contacted once.

3. Multivariate Analysis

Getting a summary of the variables

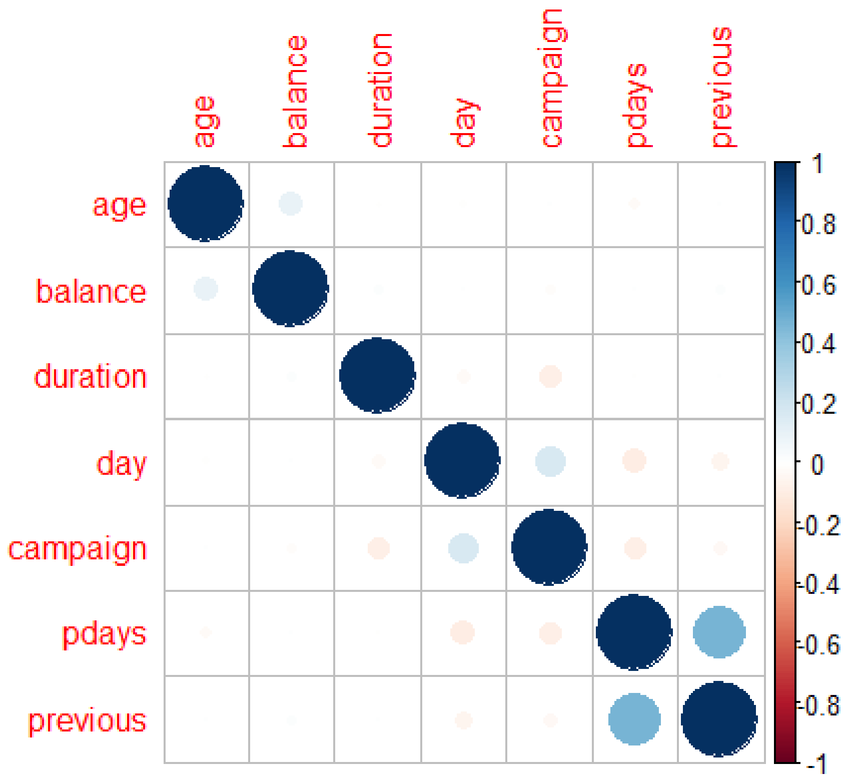
```
summary(df)

##       age             job               marital            education        
##  Min.   :18.00   Length:49732       Length:49732       Length:49732      
##  1st Qu.:33.00   Class :character   Class :character   Class :character  
##  Median :39.00   Mode  :character   Mode  :character   Mode  :character  
##  Mean   :40.96                                                            
##  3rd Qu.:48.00                                                            
##  Max.   :95.00                                                            
##    default             balance          housing              loan          
##  Length:49732       Min.   : -8019   Length:49732       Length:49732      
##  Class :character   1st Qu.:    72   Class :character   Class :character  
##  Mode  :character   Median :   448   Mode  :character   Mode  :character  
##                     Mean   :  1368                                        
##                     3rd Qu.:  1431                                        
##                     Max.   :102127                                        
##    contact               day           month              duration     
##  Length:49732       Min.   : 1.00   Length:49732       Min.   :   0.0  
##  Class :character   1st Qu.: 8.00   Class :character   1st Qu.: 103.0  
##  Mode  :character   Median :16.00   Mode  :character   Median : 180.0  
##                     Mean   :15.82                      Mean   : 258.7  
##                     3rd Qu.:21.00                      3rd Qu.: 320.0  
##                     Max.   :31.00                      Max.   :4918.0  
##     campaign          pdays           previous            poutcome        
##  Min.   : 1.000   Min.   : -1.00   Min.   :  0.0000   Length:49732      
##  1st Qu.: 1.000   1st Qu.: -1.00   1st Qu.:  0.0000   Class :character  
##  Median : 2.000   Median : -1.00   Median :  0.0000   Mode  :character  
##  Mean   : 2.767   Mean   : 40.16   Mean   :  0.5769                     
##  3rd Qu.: 3.000   3rd Qu.: -1.00   3rd Qu.:  0.0000                     
##  Max.   :63.000   Max.   :871.00   Max.   :275.0000                     
##       y            
##  Length:49732      
##  Class :character  
##  Mode  :character  
##                    
##                    
##                    
```

The summary above shows the following:

```
  * The minimum age was 18 while the maximum was 95 years while the mean was 40.

  * The minimum customer's average yearly balance was -8019, the maximum was 102127 while the mean was 1368.

  * The minimum number of days that passed by after the client was last contacted from a previous campaign was 1 day, the maximum was 31 days while the mean was 15 days.

  * The minimum number of contacts performed during this campaign and for a particular client was 1, the maximum was 63 while the mean was 2.
```

Checking for correlation

```
library(corrplot)

numeric <- df %>%
  select_if(is.numeric) %>%
  select("age", "balance", "duration", "day", "campaign", "pdays", "previous")

corrplot(cor(numeric))
```



There is no correlation among the numeric columns observed

Modeling

Will be performing our modeling using supervised method then challenge with unsupervised learning.

Loading important libraries

```
library(caTools)
library(party)
library(dplyr)
library(magrittr)
library(randomForest)
library(e1071)
library(caTools)
library(class)
library(rpart)
library(rpart.plot)
library(caret)
library(caretEnsemble)
library(psych)
library(Amelia)
library(mice)
library(GGally)
```

A. Pre-processing

Previewing our train data set.

```
head(df)

## # A tibble: 6 × 17
##     age job    marital educa…¹ default balance housing loan  contact   day
month
##   <dbl> <chr>  <chr>   <chr>   <chr>     <dbl> <chr>   <chr> <chr>   <dbl>
<chr>
## 1    58 manag… married tertia… no         2143 yes     no    unknown     5
may
## 2    44 techn… single  second… no           29 yes     no    unknown     5
may
## 3    33 entre… married second… no            2 yes     yes   unknown     5
may
## 4    47 blue-… married unknown no         1506 yes     no    unknown     5
may
## 5    33 unkno… single  unknown no            1 no      no    unknown     5
may
## 6    35 manag… married tertia… no          231 yes     no    unknown     5
may
## # … with 6 more variables: duration <dbl>, campaign <dbl>, pdays <dbl>,
## #   previous <dbl>, poutcome <chr>, y <chr>, and abbreviated variable name
## #   ¹education
## # ℹ Use `colnames()` to see all variable names
```

Selecting numeric columns

```
num <- df[, c(1,6,10,12:15)]
head(num)

## # A tibble: 6 × 7
##     age balance   day duration campaign pdays previous
##   <dbl>   <dbl> <dbl>    <dbl>    <dbl> <dbl>    <dbl>
## 1    58    2143     5      261        1    -1        0
## 2    44      29     5      151        1    -1        0
## 3    33       2     5       76        1    -1        0
## 4    47    1506     5       92        1    -1        0
## 5    33       1     5      198        1    -1        0
## 6    35     231     5      139        1    -1        0
```

Selecting categorical columns

```
cat <- df[, c(2:5,7:9,11,16,17)]
head(cat)

## # A tibble: 6 × 10
##   job          marital educa…¹ default housing loan  contact month poutc…²
y
##   <chr>        <chr>   <chr>   <chr>   <chr>   <chr> <chr>   <chr> <chr>
```

```
<chr>
## 1 management   married tertia… no      yes     no     unknown may    unknown
no
## 2 technician   single  second… no      yes     no     unknown may    unknown
no
## 3 entrepreneur married second… no      yes     yes    unknown may    unknown
no
## 4 blue-collar  married unknown no      yes     no     unknown may    unknown
no
## 5 unknown      single  unknown no      no      no     unknown may    unknown
no
## 6 management   married tertia… no      yes     no     unknown may    unknown
no
## # … with abbreviated variable names ¹education, ²poutcome
```

Label encoding our categorical columns

```
library(superml)

label <- LabelEncoder$new()
cat$job <- label$fit_transform(cat$job)
cat$marital <- label$fit_transform(cat$marital)
cat$education <- label$fit_transform(cat$education)
cat$default <- label$fit_transform(cat$default)
cat$housing <- label$fit_transform(cat$housing)
cat$loan <- label$fit_transform(cat$loan)
cat$contact <- label$fit_transform(cat$contact)
cat$month <- label$fit_transform(cat$month)
cat$poutcome <- label$fit_transform(cat$poutcome)
cat$y <- label$fit_transform(cat$y)
head(cat)

## # A tibble: 6 × 10
##     job marital education default housing  loan contact month poutcome
y
##   <dbl>   <dbl>     <dbl>   <dbl>   <dbl> <dbl>   <dbl> <dbl>    <dbl>
<dbl>
## 1     0       0         0       0       0     0       0     0        0
0
## 2     1       1         1       0       0     0       0     0        0
0
## 3     2       0         1       0       0     1       0     0        0
0
## 4     3       0         2       0       0     0       0     0        0
0
## 5     4       1         2       0       1     0       0     0        0
0
## 6     0       0         0       0       0     0       0     0        0
0
```

joining now categorical and numeric data

```r
data <-cbind(num, cat)
head(data)
```

```
##   age balance day duration campaign pdays previous job marital education
## 1  58    2143   5      261        1    -1        0   0       0         0
## 2  44      29   5      151        1    -1        0   1       1         1
## 3  33       2   5       76        1    -1        0   2       0         1
## 4  47    1506   5       92        1    -1        0   3       0         2
## 5  33       1   5      198        1    -1        0   4       1         2
## 6  35     231   5      139        1    -1        0   0       0         0
##   default housing loan contact month poutcome y
## 1       0       0    0       0     0        0 0
## 2       0       0    0       0     0        0 0
## 3       0       0    1       0     0        0 0
## 4       0       0    0       0     0        0 0
## 5       0       1    0       0     0        0 0
## 6       0       0    0       0     0        0 0
```

B. Feature selection

Will also perform feature selection to remove redundant feature in our data set.

a. Getting a correlation Matrix

```r
correlationMatrix <- cor(data)
```

b. Choosing the highly correlated features

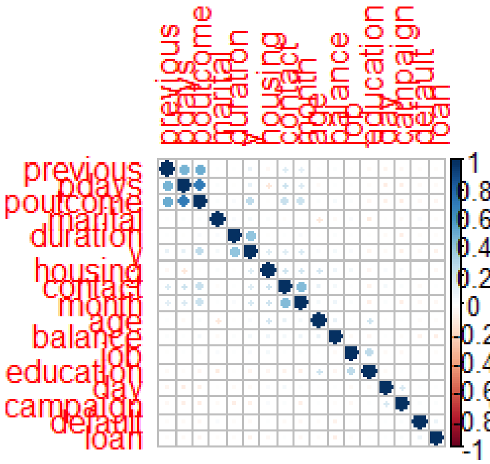
```r
highlyCorrelated <- findCorrelation(correlationMatrix, cutoff=0.70)
```

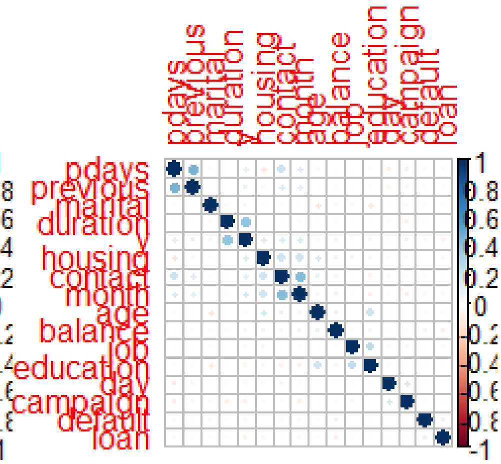
c. Removing the redundant (highly correlated) features

```r
Dataset2<-data[-highlyCorrelated]
```

d. Previews the correlation matrix

```r
par(mfrow = c(1, 2))
corrplot(correlationMatrix, order = "hclust")
corrplot(cor(Dataset2), order = "hclust")
```





We can see from the graphs above we don't have highly correlated feature so none was removed.

C. Dealing with class Imbalance

Previewing our classes

```
head(data)

##   age balance day duration campaign pdays previous job marital education
## 1  58    2143   5      261        1    -1        0   0       0         0
## 2  44      29   5      151        1    -1        0   1       1         1
## 3  33       2   5       76        1    -1        0   2       0         1
## 4  47    1506   5       92        1    -1        0   3       0         2
## 5  33       1   5      198        1    -1        0   4       1         2
## 6  35     231   5      139        1    -1        0   0       0         0
##   default housing loan contact month poutcome y
## 1       0       0    0       0     0        0 0
## 2       0       0    0       0     0        0 0
## 3       0       0    1       0     0        0 0
## 4       0       0    0       0     0        0 0
## 5       0       1    0       0     0        0 0
## 6       0       0    0       0     0        0 0

class<- (data$y)
class.frequency <- table(class)
class.frequency
```

```
## class
##     0     1
## 43922  5810
```

From this frequency table we have a huge class imbalance and will deal with them before moving forward.

```
library(imbalance)
```

Selecting the two class in the data set

```
df_p <- which(data$y == "0")
df_n <- which(data$y == "1")
```

Under sampling the majority class.

```
nsample <- 5810
pick_negative <- sample(df_p, nsample)

undersample_df1 <- data[c(df_n, pick_negative), ]

dim(undersample_df1)

## [1] 11620    17
```

The final product we have a new data set with 11620 rows and 17 columns

Previewing our response variable class

```
table(undersample_df1$y)

##
##    0    1
## 5810 5810
```

Now will go ahead and split our data into train and test data set

```
train.size = floor(0.75*nrow(undersample_df1))
train.index = sample(1:nrow(undersample_df1), train.size)
train.set = undersample_df1[train.index,]
test.set = undersample_df1[-train.index,]
x.train = train.set[,-17]
x.test = test.set[,-17]
y.train = train.set[,17]
y.test = test.set[,17]
```

KNN Classifier Model

Fitting KNN model

```
knn.3 <- knn(train = x.train, test = x.test, cl = y.train , k = 5)
```

```
def = table(predicted = knn.3, true = y.test)
def

##          true
## predicted    0    1
##         0 1087  366
##         1  349 1103

confusionMatrix(def)

## Confusion Matrix and Statistics
##
##          true
## predicted    0    1
##         0 1087  366
##         1  349 1103
##
##                Accuracy : 0.7539
##                  95% CI : (0.7378, 0.7694)
##     No Information Rate : 0.5057
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.5077
##
##  Mcnemar's Test P-Value : 0.5496
##
##             Sensitivity : 0.7570
##             Specificity : 0.7509
##          Pos Pred Value : 0.7481
##          Neg Pred Value : 0.7596
##              Prevalence : 0.4943
##          Detection Rate : 0.3742
##    Detection Prevalence : 0.5002
##       Balanced Accuracy : 0.7539
##
##        'Positive' Class : 0
##
```

The model gives us a balanced accuracy of 76.08 before any hyper parameter tuning is performed.

Parameter tuning

creating Standardization function

```
standardize = function(x){
  z <- (x - mean(x)) / sd(x)
  return( z)
}
```

applying the function to the data set

```
undersample_df2 <-
  apply(undersample_df1, 2, standardize)
head(undersample_df2)

##             age     balance       day  duration   campaign      pdays
## 84   1.48749280  0.25750091 -1.261149 1.8768338 -0.5649107 -0.4827459
## 87   1.23586231 -0.48867169 -1.261149 3.0820085 -0.5649107 -0.4827459
## 88  -0.02229016 -0.09090779 -1.261149 2.8608235 -0.5649107 -0.4827459
## 130  1.15198548  0.30068670 -1.261149 0.5639024 -0.5649107 -0.4827459
## 169  1.06810864 -0.44353766 -1.261149 0.8304587 -0.1943206 -0.4827459
## 271  0.06158667 -0.50328342 -1.261149 0.5156954 -0.1943206 -0.4827459
##       previous        job    marital   education    default    housing
## 84  -0.3744594  0.7274588 -0.7895147  0.01835929 -0.1257843 -1.0448518
## 87  -0.3744594  0.7274588 -0.7895147  0.01835929 -0.1257843  0.9569912
## 88  -0.3744594 -0.8102461 -0.7895147  0.01835929 -0.1257843 -1.0448518
## 130 -0.3744594  1.0349998 -0.7895147  0.01835929 -0.1257843 -1.0448518
## 169 -0.3744594  0.7274588 -0.7895147 -1.02740014 -0.1257843  0.9569912
## 271 -0.3744594 -1.1177871  0.6543255 -1.02740014 -0.1257843 -1.0448518
##           loan   contact     month   poutcome        y
## 84  -0.3852733 -1.698934 -1.035052 -0.5169591 0.999957
## 87  -0.3852733 -1.698934 -1.035052 -0.5169591 0.999957
## 88  -0.3852733 -1.698934 -1.035052 -0.5169591 0.999957
## 130 -0.3852733 -1.698934 -1.035052 -0.5169591 0.999957
## 169 -0.3852733 -1.698934 -1.035052 -0.5169591 0.999957
## 271  2.5953366 -1.698934 -1.035052 -0.5169591 0.999957

train1.size = floor(0.75*nrow(undersample_df2))
train1.index = sample(1:nrow(undersample_df1), train1.size)
train1.set = undersample_df2[train1.index,]
test1.set = undersample_df2[-train1.index,]
x.train1 = train1.set[,-17]
x.test1 = test1.set[,-17]
y.train1 = train1.set[,17]
y.test1 = test1.set[,17]

knn5 <- knn(train = x.train1, test = x.test1, cl = y.train1 , k = 5)

defp = table(predicted = knn5, true = y.test1)

confusionMatrix(defp)

## Confusion Matrix and Statistics
##
##                        true
## predicted                -0.999956969814305 0.999956969814305
##    -0.999956969814305                  1231               314
##    0.999956969814305                    261              1099
##
##                Accuracy : 0.8021
##                  95% CI : (0.7871, 0.8164)
##     No Information Rate : 0.5136
```

```
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.6034
##
##  Mcnemar's Test P-Value : 0.03012
##
##             Sensitivity : 0.8251
##             Specificity : 0.7778
##          Pos Pred Value : 0.7968
##          Neg Pred Value : 0.8081
##              Prevalence : 0.5136
##          Detection Rate : 0.4238
##    Detection Prevalence : 0.5318
##       Balanced Accuracy : 0.8014
##
##        'Positive' Class : -0.999956969814305
##
```

After hyper parameter tuning our model improved to 78.49% balanced accuracy.

Naive Bayes

Fitting Naive Bayes Model

```
set.seed(120)
classifier_cl <- naiveBayes(y.train ~ ., data = x.train)
```

Predicting on test data'

```
y_pred <- predict(classifier_cl, newdata = x.test)
```

Confusion Matrix

```
cm <- table(y.test, y_pred)
cm

##         y_pred
## y.test     0    1
##       0 1116  320
##       1  421 1048
```

Model Evaluation

```
confusionMatrix(cm)

## Confusion Matrix and Statistics
##
##         y_pred
## y.test     0    1
##       0 1116  320
##       1  421 1048
##
```

```
##                Accuracy : 0.7449
##                  95% CI : (0.7287, 0.7607)
##     No Information Rate : 0.5291
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.4902
##
##  Mcnemar's Test P-Value : 0.0002392
##
##             Sensitivity : 0.7261
##             Specificity : 0.7661
##          Pos Pred Value : 0.7772
##          Neg Pred Value : 0.7134
##              Prevalence : 0.5291
##          Detection Rate : 0.3842
##    Detection Prevalence : 0.4943
##       Balanced Accuracy : 0.7461
##
##        'Positive' Class : 0
##
```

The model had a balanced accuracy of 74.79% which was lower than knn and also below our metrics of success

SVM

Fitting SVM to the Training set

```
classifier = svm(formula = y.train ~ .,
                 data = x.train,
                 type = 'C-classification',
                 kernel = 'linear')
```

Predicting the Test set results

```
y_pred = predict(classifier, newdata = x.test)
```

Making the Confusion Matrix

```
cm = table(y.test, y_pred)
cm

##       y_pred
## y.test    0    1
##      0 1182  254
##      1  286 1183

confusionMatrix(cm)

## Confusion Matrix and Statistics
##
##       y_pred
```

```
## y.test    0    1
##      0 1182  254
##      1  286 1183
##
##                Accuracy : 0.8141
##                  95% CI : (0.7995, 0.8281)
##     No Information Rate : 0.5053
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.6283
##
##  Mcnemar's Test P-Value : 0.1822
##
##             Sensitivity : 0.8052
##             Specificity : 0.8232
##          Pos Pred Value : 0.8231
##          Neg Pred Value : 0.8053
##              Prevalence : 0.5053
##          Detection Rate : 0.4069
##    Detection Prevalence : 0.4943
##       Balanced Accuracy : 0.8142
##
##        'Positive' Class : 0
##
```

The SVM model had a balanced accuracy of 81.14% making the best model compared to the previous two, and also qualifies with our metric of success.

Unsupervised Learning using K-Mean Clustering Method

```
head(undersample_df1)

##     age balance day duration campaign pdays previous job marital education
## 84   59    2343   5     1042        1    -1        0   6       0         1
## 87   56      45   5     1467        1    -1        0   6       0         1
## 88   41    1270   5     1389        1    -1        0   1       0         1
## 130  55    2476   5      579        1    -1        0   7       0         1
## 169  54     184   5      673        2    -1        0   6       0         0
## 271  42       0   5      562        2    -1        0   0       1         0
##     default housing loan contact month poutcome y
## 84        0       0    0       0     0        0 1
## 87        0       1    0       0     0        0 1
## 88        0       0    0       0     0        0 1
## 130       0       0    0       0     0        0 1
## 169       0       1    0       0     0        0 1
## 271       0       0    1       0     0        0 1
```

Selecting the predictor columns

```
predictorcol <- undersample_df1[, -17]

label <- undersample_df1[, 17]
```

Fitting the K-mean Clustering model using k=2

```
kmeans.re <- kmeans(predictorcol, centers = 2, nstart = 20)
```

Confusion Matrix

```
kmeancm <- table(label, kmeans.re$cluster)
kmeancm

##
## label    1    2
##     0  256 5554
##     1  321 5489
```

The table shows despite being to make correct prediction of the two classes , there was also a case of high mis-prediction for both classes making this model unsuitable

Conclusion

- From our models above we are able to see they performed differently summarized below:
  - KNN model = 78.49% Balanced accuracy
  - Naive Bayes = 74.79% Balanced accuracy
  - SVM = 81.14% Balanced accuracy
- Overall the best model to determine is a customer subscribe to term deposit or not is SVM. It's also important to note unsupervised techniques are not suitable for this project.
- Most Customers who will subscribe to term deposit are those without loan (housing and personal Loan).
- Making multiple campaign calls to the same customer doesn't result in them subscribing to term deposit.
- Having credit on default doesn't equate term deposit subscription.

Recommendation

For effectiveness of the campaigns the marketing team would:

```
  * Don't call one customer multiple times (more than 2 times) instead spread
that time to other customers.

  * Be aware people with previous loan (any form) might not be willing to
```

```
subscribe to a term deposit.
```