# Statistical project

Brenda Téllez, Abishek Varma, Huzaifa Fazal

2022-07-07

## Setting up the working directory

```
#setwd("~/UNIPD/statistic 2/st")
setwd("/Users/huzaifa/Desktop/Unipd/Semester 2/Statistical Learning/project/Bank/bankproject")
```

Importing the libraries

```
library(dlookr)
```

```
##
## Attaching package: 'dlookr'

## The following object is masked from 'package:base':
##
##     transform
```

```
library(readr)
library(lattice)
library(modelr)
library(MASS)
#library(rgl)
library(fastDummies)
library(recipes)
```

```
## Loading required package: dplyr

##
## Attaching package: 'dplyr'

## The following object is masked from 'package:MASS':
##
##     select

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

##
## Attaching package: 'recipes'

## The following object is masked from 'package:stats':
##
##     step
```

```r
library(dummy)
```

```
## dummy 0.1.3
```

```
## dummyNews()
```

```r
library(zoo)
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```r
library(ggstatsplot)
```

```
## You can cite this package as:
##      Patil, I. (2021). Visualizations with statistical details: The 'ggstatsplot' approach.
##      Journal of Open Source Software, 6(61), 3167, doi:10.21105/joss.03167
```

```r
library(inspectdf)
library(ggplot2)
library(ggthemes)
library(vcd)
```

```
## Loading required package: grid
```

```r
library(ggmosaic)
```

```
##
## Attaching package: 'ggmosaic'
```

```
## The following objects are masked from 'package:vcd':
##
##     mosaic, spine
```

```r
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
##
## Attaching package: 'GGally'
```

```
## The following object is masked from 'package:ggmosaic':
##
##     happy
```

```r
library(caTools)
library(plyr)
```

```
## ------------------------------------------------------------------------------
```

```
## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)
```

```
## ------------------------------------------------------------------------------
```

```
##
```

```
## Attaching package: 'plyr'

## The following objects are masked from 'package:dplyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
```

```
library(stringr)
```

```
##
## Attaching package: 'stringr'

## The following object is masked from 'package:recipes':
##
##     fixed
```

```
library(scales)
```

```
##
## Attaching package: 'scales'

## The following object is masked from 'package:readr':
##
##     col_factor
```

```
library(dplyr)
library(VIM)
```

```
## Loading required package: colorspace

## VIM is ready to use.

## Suggestions and bug-reports can be submitted at: https://github.com/statistikat/VIM/issues

##
## Attaching package: 'VIM'

## The following object is masked from 'package:recipes':
##
##     prepare

## The following object is masked from 'package:datasets':
##
##     sleep
```

```
library(naniar)
library(egg)
```

```
## Loading required package: gridExtra

##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##     combine
```

#Introduction • Introduction A common practice to enhance and stimulate business growth is using marketing campaigns. Marketing campaigns come in many different forms, ranging from acquisition marketing campaigns to social media marketing campaigns. A common and widely used marketing campaign is the telemarketing strategy that banks oftentimes utilize due to the complex nature of financial products that require more nuanced explanations. However, telemarketing campaigns are demanding in terms of time, effort and resources needed. Therefore, it is of big significance to determine what factors associated with telemarketing, and/or otherwise, affect whether a client purchases financial products or not.

The aim of this project is to analyze the dataset, identify trends and build models that can determine whether a client purchases a long-term deposit based on factors such as gender, age, occupation, previous loans, previous campaign interactions, etc. For example, we are interested in identifying the duration of telemarketing calls that yield the most positive results. Which day of the week and which month should be focused on for a higher chance of success? Does it make a difference whether clients are called on their cellphone or on their telephone? Does the job or education of a client significantly affect their decision?

The dataset contains data from a Portuguese commercial bank that details various bank-client relationship information. Using this information, we generated models to predict the outcome of purchase decisions of clients.

# Data Collection

This project uses a dataset that is originally sourced from a Portuguese retail bank and was used by [S. Moro, P.cortez and P. Rita}. The dataset contains features related to direct marketing campaigns for the purpose of selling bank long-term deposits. We obtained the dataset from the UC Irvine Machine Learning Repository.

o Dataset Description • The dataset is multivariate that contains 45211 instances (rows) with 21 features (columns). Out of the 21 features, 20 are used as potential predicting factors that might affect whether a direct marketing campaign that involved telemarketing calls is successful in selling a long-term deposit or not. The feature column titled "y" is used as the target column that details whether a client subscribed to a long-term deposit or no.

The feature columns can be divided into 3-4 subgroups: personal bank details, previous contacts for current campaign, contacts for previous campaigns, and social and economic attributes. The details of each subgroup and its constituents' features can be found in the appendix xxx. The dataset contains columns that are numerical such as age, duration, pdays, etc and columns that are categorical such as job, education, loan, etc. Within the numerical features, there are continuous variables such as cons.price.idx and discrete variables such as Employment.number. Similarly, within the categorical columns, there are variables such as education are ordinal and variables such as job are nominal.

#Data Manipulation ## Importing the dataset The dataset didnt contain "NA" values but rather had "Unknown" values in multiple columns. On initial inspection of the dataset, it became apparent that the "unknown" values were missing values in most columns. However, this was not the case for all columns. Explained further in xxx. Therefore, we imported the dataset with specifying the unknown values in the dataset as NA values.

```
bank <- read.csv("bank-additional-full.csv", sep=";", na="unknown")
```

```
summary(bank)
```

```
##       age            job              marital           education
##  Min.   :17.00   Length:41188       Length:41188       Length:41188
##  1st Qu.:32.00   Class :character   Class :character   Class :character
##  Median :38.00   Mode  :character   Mode  :character   Mode  :character
##  Mean   :40.02
##  3rd Qu.:47.00
##  Max.   :98.00
##    default            housing             loan              contact
##  Length:41188       Length:41188       Length:41188       Length:41188
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
```

```
##
##
##     month            day_of_week          duration         campaign
##  Length:41188       Length:41188       Min.   :   0.0   Min.   : 1.000
##  Class :character   Class :character   1st Qu.: 102.0   1st Qu.: 1.000
##  Mode  :character   Mode  :character   Median : 180.0   Median : 2.000
##                                        Mean   : 258.3   Mean   : 2.568
##                                        3rd Qu.: 319.0   3rd Qu.: 3.000
##                                        Max.   :4918.0   Max.   :56.000
##      pdays          previous       poutcome          emp.var.rate
##  Min.   :  0.0   Min.   :0.000   Length:41188       Min.   :-3.40000
##  1st Qu.:999.0   1st Qu.:0.000   Class :character   1st Qu.:-1.80000
##  Median :999.0   Median :0.000   Mode  :character   Median : 1.10000
##  Mean   :962.5   Mean   :0.173                      Mean   : 0.08189
##  3rd Qu.:999.0   3rd Qu.:0.000                      3rd Qu.: 1.40000
##  Max.   :999.0   Max.   :7.000                      Max.   : 1.40000
##  cons.price.idx  cons.conf.idx     euribor3m      nr.employed
##  Min.   :92.20   Min.   :-50.8   Min.   :0.634   Min.   :4964
##  1st Qu.:93.08   1st Qu.:-42.7   1st Qu.:1.344   1st Qu.:5099
##  Median :93.75   Median :-41.8   Median :4.857   Median :5191
##  Mean   :93.58   Mean   :-40.5   Mean   :3.621   Mean   :5167
##  3rd Qu.:93.99   3rd Qu.:-36.4   3rd Qu.:4.961   3rd Qu.:5228
##  Max.   :94.77   Max.   :-26.9   Max.   :5.045   Max.   :5228
##       y
##  Length:41188
##  Class :character
##  Mode  :character
##
##
##
```

```
str(bank)
```

```
## 'data.frame':    41188 obs. of  21 variables:
##  $ age           : int  56 57 37 40 56 45 59 41 24 25 ...
##  $ job           : chr  "housemaid" "services" "services" "admin." ...
##  $ marital       : chr  "married" "married" "married" "married" ...
##  $ education     : chr  "basic.4y" "high.school" "high.school" "basic.6y" ...
##  $ default       : chr  "no" NA "no" "no" ...
##  $ housing       : chr  "no" "no" "yes" "no" ...
##  $ loan          : chr  "no" "no" "no" "no" ...
##  $ contact       : chr  "telephone" "telephone" "telephone" "telephone" ...
##  $ month         : chr  "may" "may" "may" "may" ...
##  $ day_of_week   : chr  "mon" "mon" "mon" "mon" ...
##  $ duration      : int  261 149 226 151 307 198 139 217 380 50 ...
##  $ campaign      : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ pdays         : int  999 999 999 999 999 999 999 999 999 999 ...
##  $ previous      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ poutcome      : chr  "nonexistent" "nonexistent" "nonexistent" "nonexistent" ...
##  $ emp.var.rate  : num  1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 ...
##  $ cons.price.idx: num  94 94 94 94 94 ...
##  $ cons.conf.idx : num  -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 ...
##  $ euribor3m     : num  4.86 4.86 4.86 4.86 4.86 ...
##  $ nr.employed   : num  5191 5191 5191 5191 5191 ...
##  $ y             : chr  "no" "no" "no" "no" ...
```

Renaming columns For ease of coding and a proper standard among the columns names we remaned the columns with slight modifications as shown below.

```
colnames(bank) <- c("Age", "Job", "Marital", "Education","Default","Housing" ,"Loan","Contact","Month",
"Euribor3m", "Employment.number","y")
columns <- colnames(bank)
Factorcols <- c("Job", "Marital", "Default","Education","Housing" ,"Loan","Contact","Month","Last.Conta
```

Removing the dot in the "admin." value in the job column

```
bank$Job = str_replace(bank$Job,"[.]","istration")
```

data type of columns

```
sapply(bank, class)
```

```
##               Age               Job           Marital         Education
##         "integer"       "character"       "character"       "character"
##           Default           Housing              Loan           Contact
##       "character"       "character"       "character"       "character"
##             Month  Last.Contact.Day          Duration          Campaign
##       "character"       "character"         "integer"         "integer"
##             Pdays Previous.Contacts          Poutcome      Emp.var.rate
##         "integer"         "integer"       "character"         "numeric"
##    Cons.price.idx     Cons.conf.idx         Euribor3m Employment.number
##         "numeric"         "numeric"         "numeric"         "numeric"
##                 y
##       "character"
```

##keeping a copy of original dataframe

```
data <- data.frame(bank)
```

#Data cleaning

##Handling null values

Many columns such as Default, Education, Loan, Housing, Job etc. contained a considerable amount of NA values, especially Default. Default had 20.87% of NA values. On further inspection, it was also noticed that the default column was highly imbalanced with only 3 "yes" values and 32469 "no" values, ontop of 8518 NA values. Therefore, we decided to drop the default column. The plot below gives a good idea of how many NA values are present in each column.

- percentage of null values (can we add these percentages to graph below?)

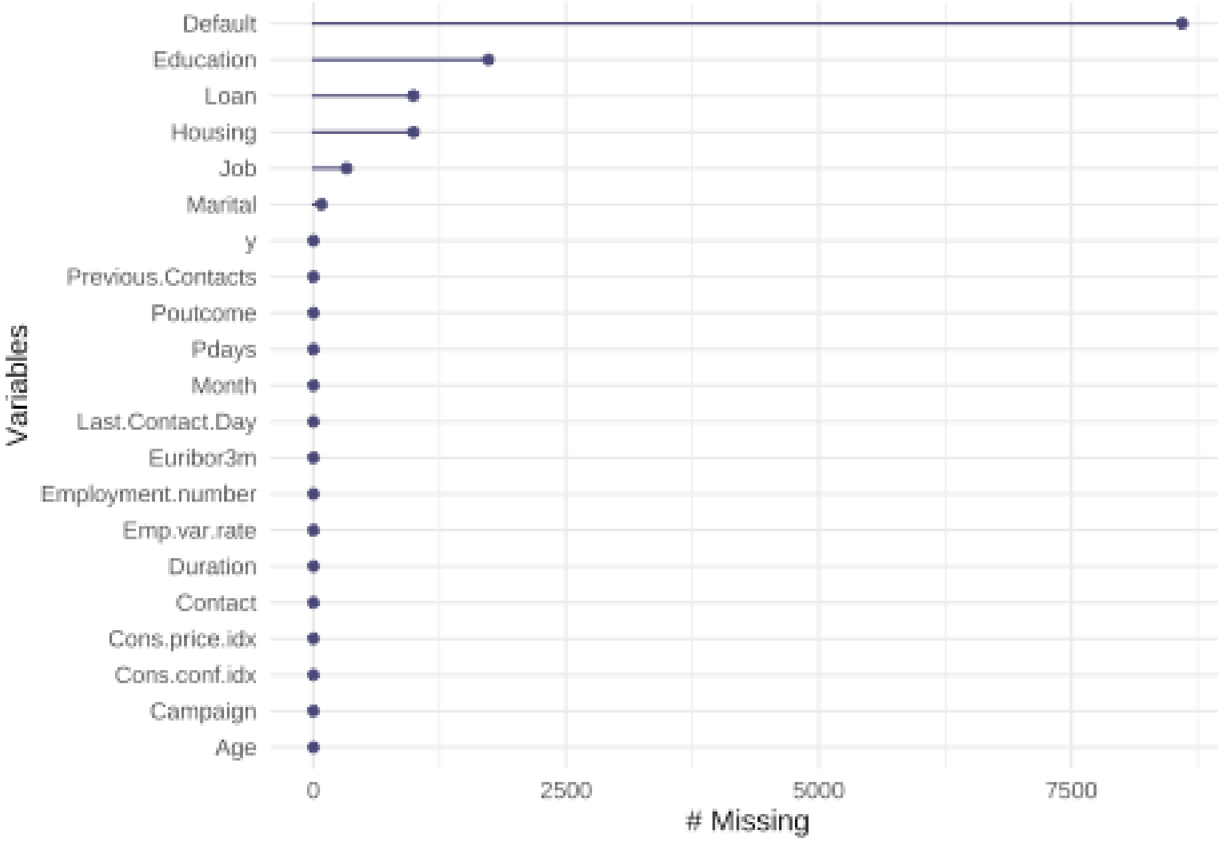
```
sapply(bank, function(x) round((sum(is.na(x))/length(x))*100,2))
```

```
##               Age               Job           Marital         Education
##              0.00              0.80              0.19              4.20
##           Default           Housing              Loan           Contact
##             20.87              2.40              2.40              0.00
##             Month  Last.Contact.Day          Duration          Campaign
##              0.00              0.00              0.00              0.00
##             Pdays Previous.Contacts          Poutcome      Emp.var.rate
##              0.00              0.00              0.00              0.00
##    Cons.price.idx     Cons.conf.idx         Euribor3m Employment.number
##              0.00              0.00              0.00              0.00
##                 y
##              0.00
```

Visualizing missing data

```
gg_miss_var(bank)
```

```
## Warning: It is deprecated to specify `guide = FALSE` to remove a guide. Please
## use `guide = "none"` instead.
```




For handling the rest of the NA values, we took 2 different approaches. First, where NA values are considered as NA values and therefore dealt with either deletion or imputations. Second, where NA values, that were originally labelled as "unknown" in the dataset, are considered as a separate category in their respective column. For example, the default column details whether a client has credit in default or not - which translates to "yes" or "no". However, in real-world scenarios it is very plausible for their to exist a third category where a client may choose to not answer questions regarding their credit in default status. Questions such as credit in default, loans, etc can be a sensitive topic and therefore clients may choose not to comment on these questions. Therefore, we decided that "unknown" entries in the default column should be considered as one of the options for a response. Therefore, the default column has 3 possible responses/categories: yes, no, or unknown.

For the Education column, NA values were dealt with using imputations. Using a contingency table between Education and job, simple logical inferences were made between the a client's job and their education. For example, most clients that have a management job are most likely to have a university degree. Most clients that have a services job are most likely to have a high school education. Therefore, using these inferences we imputed the NA values of the Education column. Similarly, imputations were made for the Job column. If client's age is greater or equal to 66 and Job column is equal to NA then we imputted the missing value to retired.

For Housing, Loan, and Default columns the NA values were replaced back to "unknown" values as they will be considered as a category within their respective columns, as discussed before.

No logical inference could be made for the marital column and therefore rows containing NA values were removed entirely. It was also noticed there were 990 rows where the Education and Job column both had NA values. This suggests that the NA values are not a result of some random event but rather are related. Furthermore, since our method of imputation used earlier would not be possible as both values are missing, we decided to remove all these rows. It can now be seen that the cleaned dataset contains no NA values.

##contingency table to infer job from education and viz

```
JobvsEd <- table(bank$Job,bank$Education,useNA = "always")
JobvsEd
```

```
##
##                  basic.4y basic.6y basic.9y high.school illiterate
##   administration       77      151      499        3329          1
##   blue-collar        2318     1426     3623         878          8
##   entrepreneur        137       71      210         234          2
##   housemaid           474       77       94         174          1
##   management          100       85      166         298          0
##   retired             597       75      145         276          3
##   self-employed        93       25      220         118          3
##   services            132      226      388        2682          0
##   student              26       13       99         357          0
##   technician           58       87      384         873          0
##   unemployed          112       34      186         259          0
##   <NA>                 52       22       31          37          0
##
##                  professional.course university.degree <NA>
##   administration                 363              5753  249
##   blue-collar                    453                94  454
##   entrepreneur                   135               610   57
##   housemaid                       59               139   42
##   management                      89              2063  123
##   retired                        241               285   98
##   self-employed                  168               765   29
##   services                       218               173  150
##   student                         43               170  167
##   technician                    3320              1809  212
##   unemployed                     142               262   19
##   <NA>                            12                45  131
```

filling null values of job based on Age

```
bank$Job[bank$Age >= 66 & is.na(bank$Job)] <- "retired"
```

filling the null values for education

```
remove_null_Ed <- function(bank, tab){
  for(column in unique(bank$Job[!is.na(bank$Job)])){
    bank$Education[bank$Job==column & is.na(bank$Education)] <- names(which.max(tab[column,]))
  }
  return(bank)
}
```

```
bank <- remove_null_Ed(bank, JobvsEd)
```

filling the null values for Job

```
remove_null_Job <- function(bank, tab){
  for(column in unique(bank$Education[!is.na(bank$Education)])){
```

```r
    bank$Job[bank$Education==column & is.na(bank$Job)] <- names(which.max(tab[,column]))
  }
  return(bank)
}
```

```r
bank <- remove_null_Job(bank, JobvsEd)
```

contingency for personal and housing loan

```r
table(bank$Housing,bank$Loan,useNA = "always")
```

```
##
##          no   yes  <NA>
##   no   16065  2557     0
##   yes  17885  3691     0
##   <NA>     0     0   990
```

replacing with unknowns

```r
bank$Housing[is.na(bank$Housing)] <- "unknown"
bank$Loan[is.na(bank$Loan)] <- "unknown"
bank$Default[is.na(bank$Default)] <- "unknown"
```

null values in marital

```r
sum(is.na(bank$Marital))
```

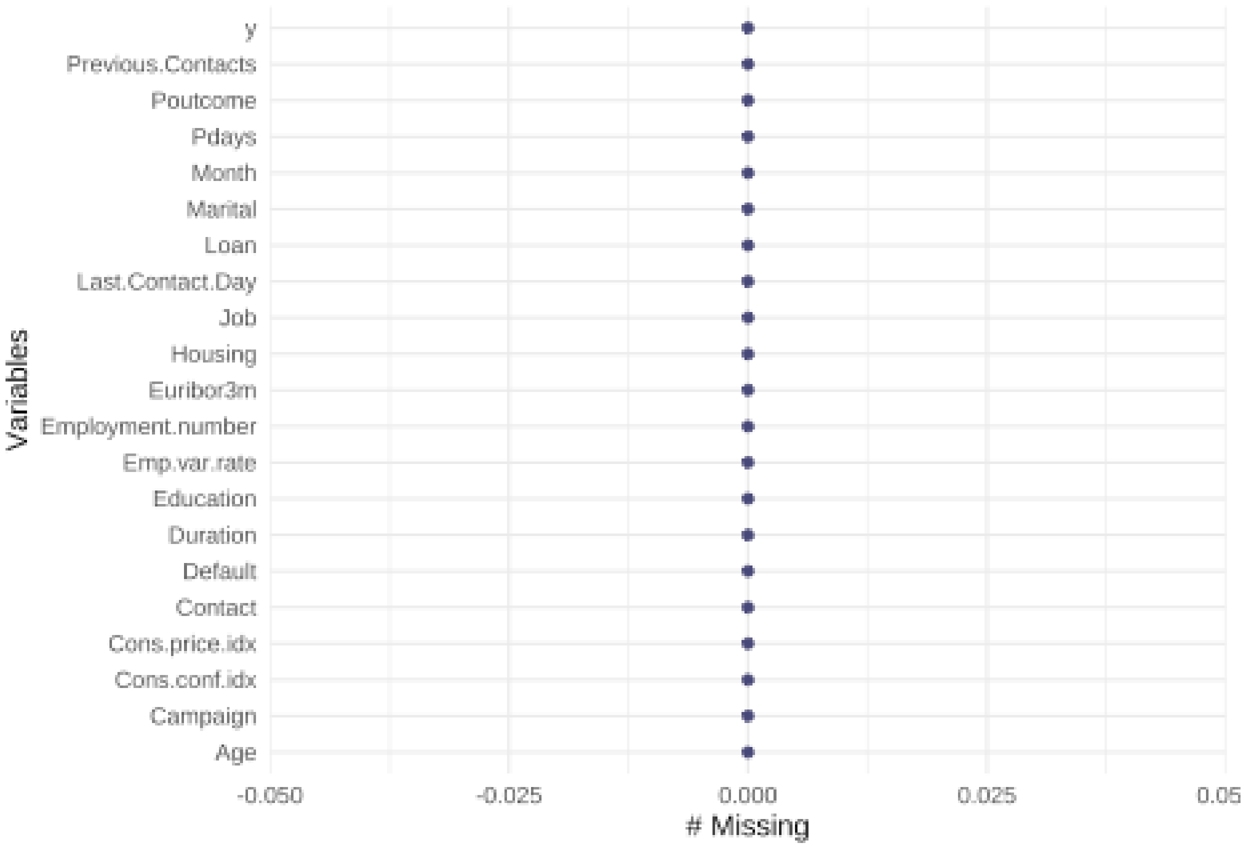
```
## [1] 80
```

removing rows with marital, Job or education as null

```r
bank <- na.omit(bank)
```

Again visualize missing

```r
gg_miss_var(bank)
```




**End of handling Missing Values**

##data transformations

All categorical columns were assigned to the factors datatype for the ease of using them in various prediction models. Some of the categorical columns were ordinal while others were nominal. Education, Month, and day_of_week are ordinal columns and thus they were assigned as an ordered factor. Job, marital, housing, loan, etc were assigned as unordered factors. However, after using ordered factor columns in our prediction models, we realised there was not much benefit in using them while it caused minor complications in some models. Therefore, we instead chnaged all categorical columns to unordered factor columns.

Feature scaling was used for all the numerical variables using MinMax scaling to normalize the range of the variables and ensure the prediction models work properly. reorder the row indices.

```
rownames(bank) <- 1:nrow(bank)
```

##Converting character types to factors

```
bank[Factorcols] <- lapply(bank[Factorcols], as.factor)
```

```
str(bank)
```

```
## 'data.frame':    40990 obs. of  21 variables:
##  $ Age                : int  56 57 37 40 56 45 59 41 24 25 ...
##  $ Job                : Factor w/ 11 levels "administration",..: 4 8 8 1 8 8 1 2 10 8 ...
##  $ Marital            : Factor w/ 3 levels "divorced","married",..: 2 2 2 2 2 2 2 2 3 3 ...
##  $ Education          : Factor w/ 7 levels "basic.4y","basic.6y",..: 1 4 4 2 4 3 6 3 6 4 ...
##  $ Default            : Factor w/ 3 levels "no","unknown",..: 1 2 1 1 1 2 1 2 1 1 ...
##  $ Housing            : Factor w/ 3 levels "no","unknown",..: 1 1 3 1 1 1 1 1 3 3 ...
##  $ Loan               : Factor w/ 3 levels "no","unknown",..: 1 1 1 1 3 1 1 1 1 1 ...
##  $ Contact            : Factor w/ 2 levels "cellular","telephone": 2 2 2 2 2 2 2 2 2 2 ...
```

```
##  $ Month           : Factor w/ 10 levels "apr","aug","dec",..: 7 7 7 7 7 7 7 7 7 7 ...
##  $ Last.Contact.Day : Factor w/ 5 levels "fri","mon","thu",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ Duration        : int  261 149 226 151 307 198 139 217 380 50 ...
##  $ Campaign        : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Pdays           : int  999 999 999 999 999 999 999 999 999 999 ...
##  $ Previous.Contacts: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Poutcome        : Factor w/ 3 levels "failure","nonexistent",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ Emp.var.rate    : num  1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 ...
##  $ Cons.price.idx  : num  94 94 94 94 94 ...
##  $ Cons.conf.idx   : num  -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 ...
##  $ Euribor3m       : num  4.86 4.86 4.86 4.86 4.86 ...
##  $ Employment.number: num  5191 5191 5191 5191 5191 ...
##  $ y               : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
##  - attr(*, "na.action")= 'omit' Named int [1:198] 41 74 92 300 304 344 389 391 414 429 ...
##   ..- attr(*, "names")= chr [1:198] "41" "74" "92" "300" ...
```

#Exploratory Data Analysis

dimension of the dataset

```r
dim(bank)
```

```
## [1] 40990    21
```

There are 40990 rows and 21 columns

Taking a glance at first and last five rows

```r
head(bank)
```

```
##   Age           Job Marital   Education Default Housing Loan   Contact Month
## 1  56     housemaid married    basic.4y      no      no   no telephone   may
## 2  57      services married high.school unknown      no   no telephone   may
## 3  37      services married high.school      no     yes   no telephone   may
## 4  40 administration married    basic.6y      no      no   no telephone   may
## 5  56      services married high.school      no      no  yes telephone   may
## 6  45      services married    basic.9y unknown      no   no telephone   may
##   Last.Contact.Day Duration Campaign Pdays Previous.Contacts    Poutcome
## 1              mon      261        1   999                 0 nonexistent
## 2              mon      149        1   999                 0 nonexistent
## 3              mon      226        1   999                 0 nonexistent
## 4              mon      151        1   999                 0 nonexistent
## 5              mon      307        1   999                 0 nonexistent
## 6              mon      198        1   999                 0 nonexistent
##   Emp.var.rate Cons.price.idx Cons.conf.idx Euribor3m Employment.number  y
## 1          1.1         93.994         -36.4     4.857              5191 no
## 2          1.1         93.994         -36.4     4.857              5191 no
## 3          1.1         93.994         -36.4     4.857              5191 no
## 4          1.1         93.994         -36.4     4.857              5191 no
## 5          1.1         93.994         -36.4     4.857              5191 no
## 6          1.1         93.994         -36.4     4.857              5191 no
```

```r
tail(bank)
```

```
##       Age         Job Marital           Education Default Housing Loan  Contact
## 40985  29  unemployed  single            basic.4y      no     yes   no cellular
## 40986  73     retired married professional.course      no     yes   no cellular
## 40987  46 blue-collar married professional.course      no      no   no cellular
```

```
## 40988  56     retired married   university.degree      no     yes   no cellular
## 40989  44  technician married professional.course      no      no   no cellular
## 40990  74     retired married professional.course      no     yes   no cellular
##       Month Last.Contact.Day Duration Campaign Pdays Previous.Contacts
## 40985   nov              fri      112        1     9                 1
## 40986   nov              fri      334        1   999                 0
## 40987   nov              fri      383        1   999                 0
## 40988   nov              fri      189        2   999                 0
## 40989   nov              fri      442        1   999                 0
## 40990   nov              fri      239        3   999                 1
##          Poutcome Emp.var.rate Cons.price.idx Cons.conf.idx Euribor3m
## 40985     success         -1.1         94.767         -50.8     1.028
## 40986 nonexistent         -1.1         94.767         -50.8     1.028
## 40987 nonexistent         -1.1         94.767         -50.8     1.028
## 40988 nonexistent         -1.1         94.767         -50.8     1.028
## 40989 nonexistent         -1.1         94.767         -50.8     1.028
## 40990     failure         -1.1         94.767         -50.8     1.028
##       Employment.number   y
## 40985            4963.6  no
## 40986            4963.6 yes
## 40987            4963.6  no
## 40988            4963.6  no
## 40989            4963.6 yes
## 40990            4963.6  no
```

detailed statistics about the numerical features

```
describe(bank)
```

```
## # A tibble: 10 x 26
##    described_variables     n    na      mean      sd se_mean     IQR skewness
##    <chr>               <int> <int>     <dbl>   <dbl>   <dbl>   <dbl>    <dbl>
##  1 Age                 40990     0   40.0     10.4   0.0515   15        0.790
##  2 Duration            40990     0  258.     259.    1.28    217        3.27
##  3 Campaign            40990     0    2.57     2.77  0.0137    2        4.78
##  4 Pdays               40990     0  963.     187.    0.922     0       -4.93
##  5 Previous.Contacts   40990     0    0.173    0.495 0.00244   0        3.83
##  6 Emp.var.rate        40990     0    0.0805   1.57  0.00776   3.2     -0.722
##  7 Cons.price.idx      40990     0   93.6      0.579 0.00286   0.919   -0.229
##  8 Cons.conf.idx       40990     0  -40.5      4.63  0.0229    6.30     0.305
##  9 Euribor3m           40990     0    3.62     1.73  0.00857   3.62    -0.707
## 10 Employment.number   40990     0 5167.      72.3   0.357   129       -1.04
## # ... with 18 more variables: kurtosis <dbl>, p00 <dbl>, p01 <dbl>, p05 <dbl>,
## #   p10 <dbl>, p20 <dbl>, p25 <dbl>, p30 <dbl>, p40 <dbl>, p50 <dbl>,
## #   p60 <dbl>, p70 <dbl>, p75 <dbl>, p80 <dbl>, p90 <dbl>, p95 <dbl>,
## #   p99 <dbl>, p100 <dbl>
```

From the basic statistics and summary of the numerical columns, we can observe a few intersting things about the mean, median, min, max, etc. First, regarding the Pdays columns, it can be observed the mean is 962.6. The median and the max equals to 999 of the Pdays columns. At first glance this seems strange and is misleading, but it is due to the fact that the way the data was recorded. In the Pdays column, if a client was not contacted after a previous campaign then it is recorded as 999. It could be questioned why not record that as "0". This is because "0" values in the pdays column signifies that at the time of recording/collecting this data, there have been 0 days since the previous contact. i.e the client was contacted on the same day as the data was collected. What the pdays mean of 962.6 and median of 999 tells us is that the vast majority of clients were not contacted since the previous campaign.

The table shows the mean, median and max of Pdays if we remove all the 999 entries.

```
summary(bank$Pdays[bank$Pdays!=999])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   3.000   6.000   6.008   7.000  27.000
```

We considered changing the 999 values to something else as it might skew the prediction models to larger values. However, we realised it is unnecessary as results from the prediction can be interpreted using some threshold instead. If pdays values is considerably high, then that can be simply interpreted as 'not previously contacted'.

Age and Campaign columns have reasonable mean and median values, which are close to each other. The 'Previous' column has a mean of 0.1729 and median of 0. As the previous column details the number of contacts performed before the current campaign, the mean and median suggest that the majority of clients were not contacted previously. This indicates that the bank is mostly focused on targeting new customers with their campaigns (as no previous contacts have been made) or the bank has only recently started contacting customers for telemarketing purposes.

Nothing noteworthy is dsiplayed about the mean and medians of the economical, social data columns.

```
bank %>% inspect_types()
```

```
## # A tibble: 3 x 4
##   type      cnt  pcnt col_name
##   <chr>   <int> <dbl> <named list>
## 1 factor     11  52.4 <chr [11]>
## 2 integer     5  23.8 <chr [5]>
## 3 numeric     5  23.8 <chr [5]>
```

**Insights**: - There are 11 factor columns - There are 5 integer columns - There are 5 numeric columns

```
bank%>%inspect_cat()
```

```
## # A tibble: 11 x 5
##    col_name           cnt common            common_pcnt levels
##    <chr>            <int> <chr>                   <dbl> <named list>
##  1 Contact              2 cellular                 63.5 <tibble [2 x 3]>
##  2 Default              3 no                       79.2 <tibble [3 x 3]>
##  3 Education            7 university.degree        30.8 <tibble [7 x 3]>
##  4 Housing              3 yes                      52.4 <tibble [3 x 3]>
##  5 Job                 11 administration           25.6 <tibble [11 x 3]>
##  6 Last.Contact.Day     5 thu                      20.9 <tibble [5 x 3]>
##  7 Loan                 3 no                       82.4 <tibble [3 x 3]>
##  8 Marital              3 married                  60.6 <tibble [3 x 3]>
##  9 Month               10 may                      33.4 <tibble [10 x 3]>
## 10 Poutcome             3 nonexistent              86.3 <tibble [3 x 3]>
## 11 y                    2 no                       88.7 <tibble [2 x 3]>
```

The table above shows the most common category in each column with their respective percentages. In general columns that have multiple categories have lower percentages of the most common category. Columns such as default, loan, poutcome that have 2 or 3 categories are most imbalanced than the rest. It also interesting to see the most common education level and jobs of the clients that were contacted. 82.4% of the contacted clients dont have personal loans.

## Univariate Analysis

collecting columns of factor type

```r
factors <- subset(bank,select = names(Filter(is.factor,bank)))
```

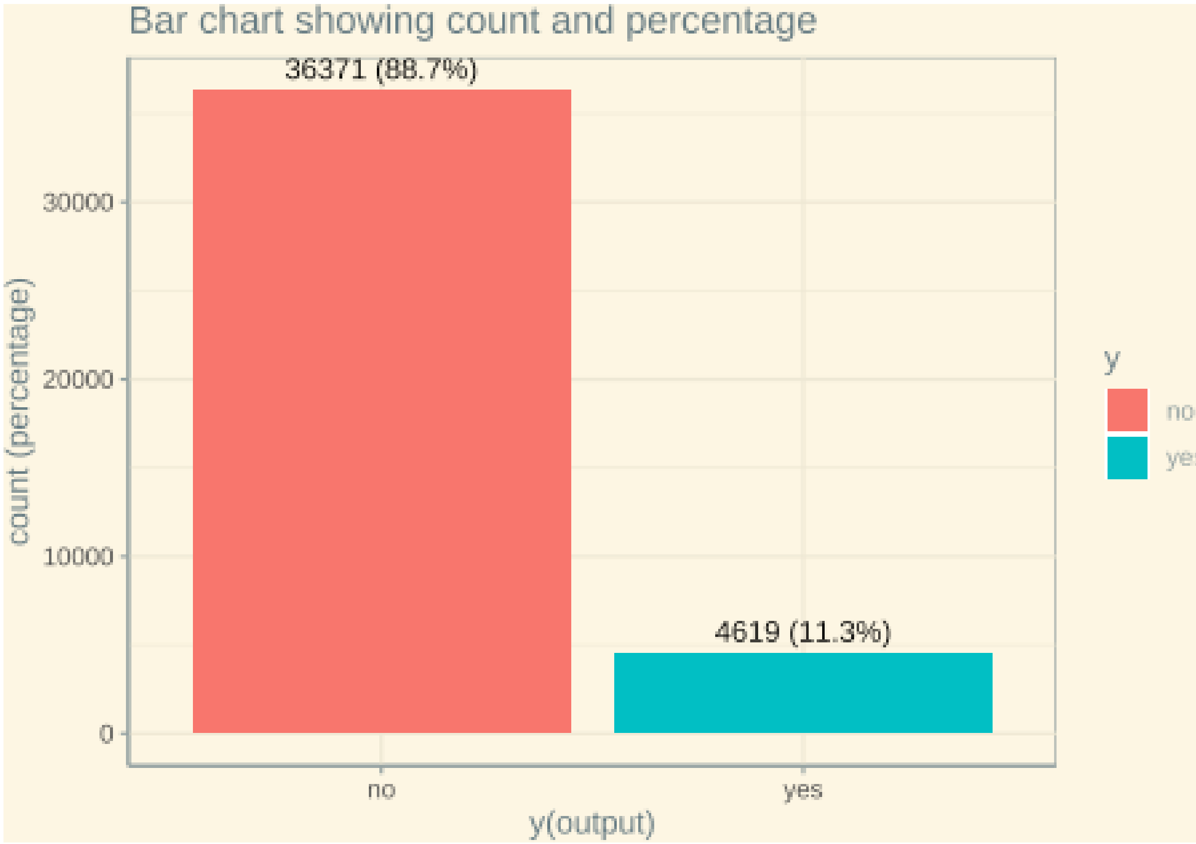
```r
summary(factors)
```

```
##              Job           Marital                     Education
##  administration:10485   divorced: 4611   basic.4y           : 4318
##  blue-collar   : 9344   married :24824   basic.6y           : 2286
##  technician    : 6743   single  :11555   basic.9y           : 6489
##  services      : 3963                    high.school        : 9817
##  management    : 2921                    illiterate         :   18
##  retired       : 1725                    professional.course: 5448
##  (Other)       : 5809                    university.degree  :12614
##     Default         Housing           Loan            Contact
##  no     :32469   no     :18526   no     :33782   cellular :26032
##  unknown: 8518   unknown:  987   unknown:  987   telephone:14958
##  yes    :    3   yes    :21477   yes    : 6221
##
##
##
##
##      Month       Last.Contact.Day        Poutcome       y
##  may    :13699   fri:7796         failure    : 4236   no :36371
##  jul    : 7148   mon:8467         nonexistent:35391   yes: 4619
##  aug    : 6135   thu:8570         success    : 1363
##  jun    : 5284   tue:8056
##  nov    : 4092   wed:8101
##  apr    : 2627
##  (Other): 2005
```

##Barcharts

The barcharts below shows that our dataset is highly imbalanced. The y-output has 36371 'No' values and 4619 'Yes' values. That is almost an imbalance ratio of 8:1 with 88.7% of the responses refusing the long-term deposit.

```r
OutputCount <- factors%>%
            dplyr::count(y)%>%
            dplyr::mutate(perc = n/sum(n) * 100)

pl <- ggplot (data = OutputCount, aes(x = y, y = n, fill = y))
pl <- pl + geom_col()
pl <- pl + geom_text(aes(x = y, y = n
                         , label = paste0(n, " (", round(perc,1),"%)")
                         , vjust = -0.5
                         ))
pl <- pl + theme_solarized() + scale_colour_solarized("red")
pl <- pl + labs(title ="Bar chart showing count and percentage", x="y(output)", y="count (percentage)")
pl
```




collecting columns of numeric type

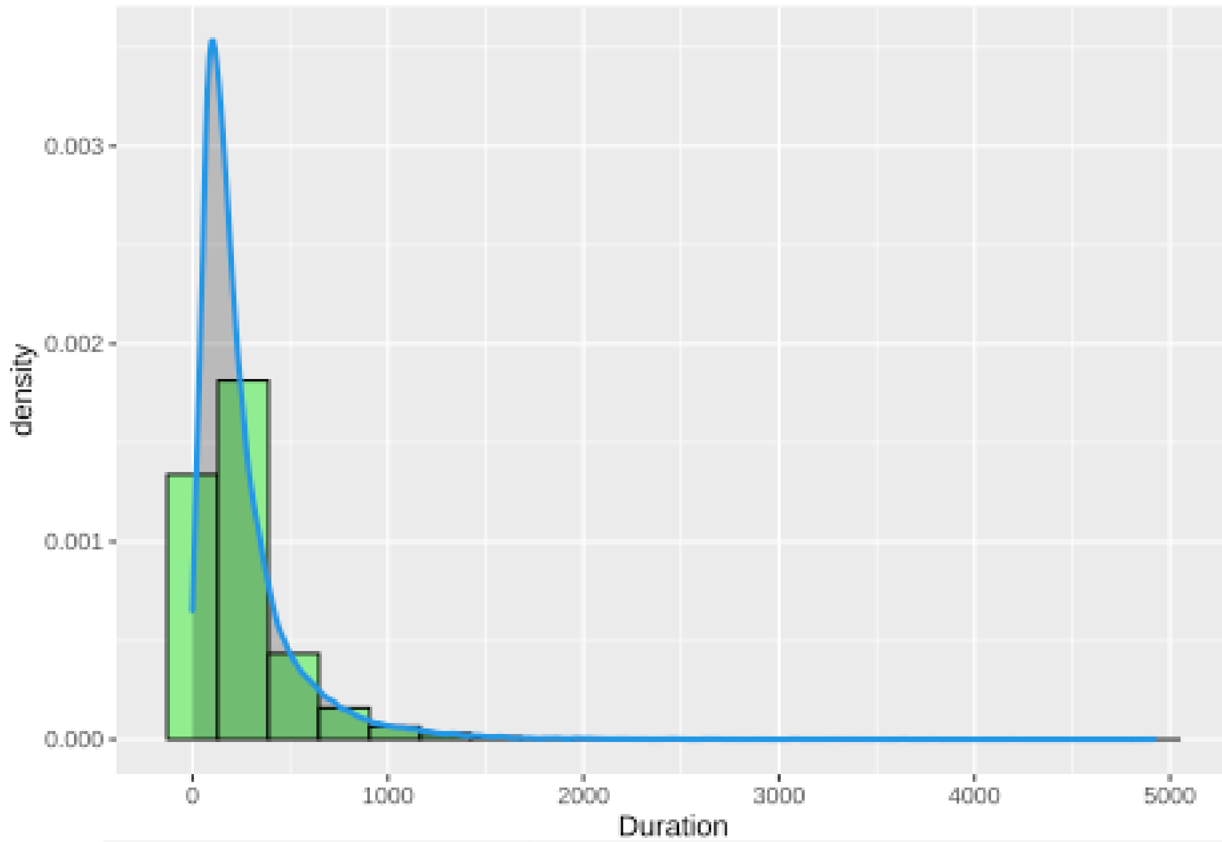
```
numerics <- subset(bank,select = names(Filter(is.numeric,bank)))
```

##Histogram with Density plot

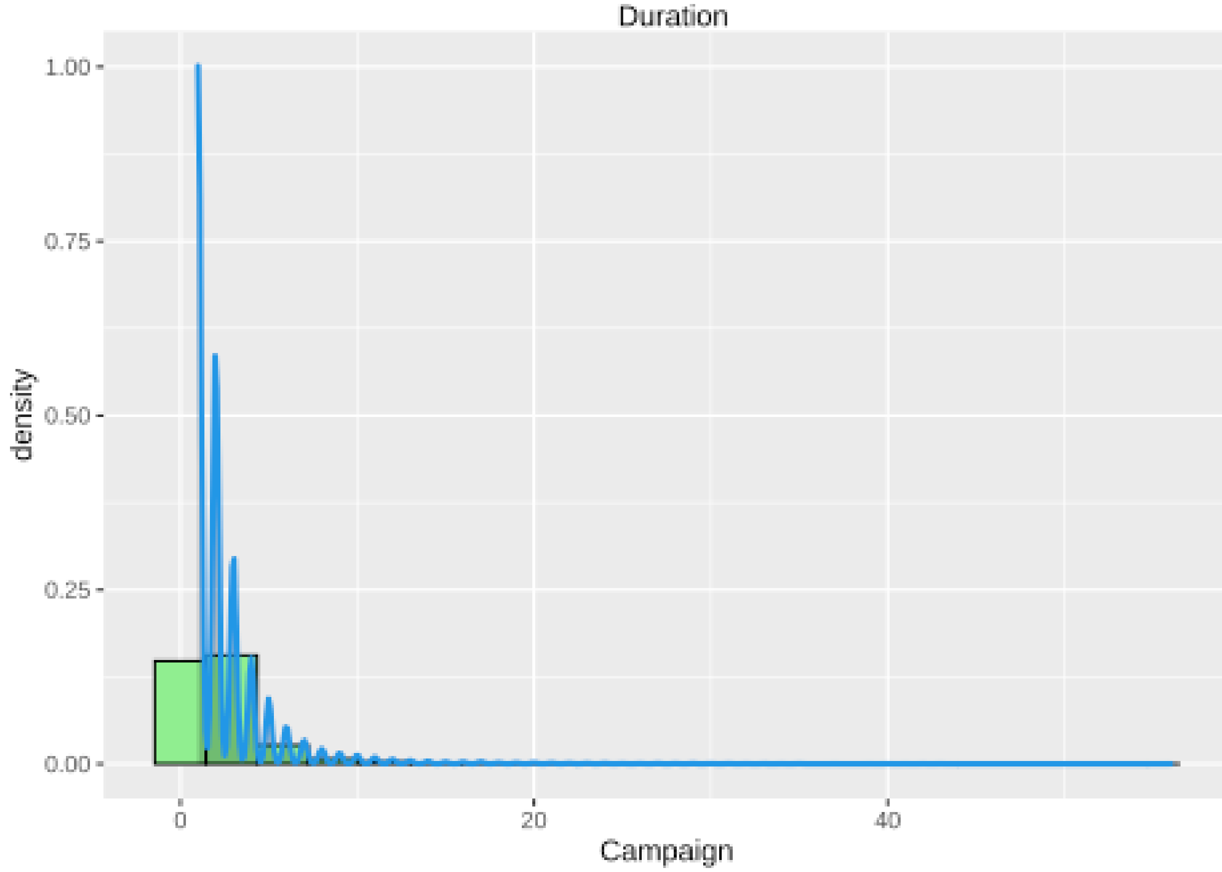
We used histograms with density plots to analyse the distribution of the various various that we will be using as predicting factors.

```
distplotsnumerics <- function(bank,numerics){
  for(col in names(numerics)){
    distplot <- ggplot(numerics, aes(x = bank[[col]])) +
    geom_histogram(aes(y = ..density..),
                   colour = 1, fill = "lightgreen", bins = 20) +
    geom_density(lwd = 1, colour = 4,
                 fill = 1, alpha = 0.25) + labs(x = col)
    print(distplot)
  }
}
```

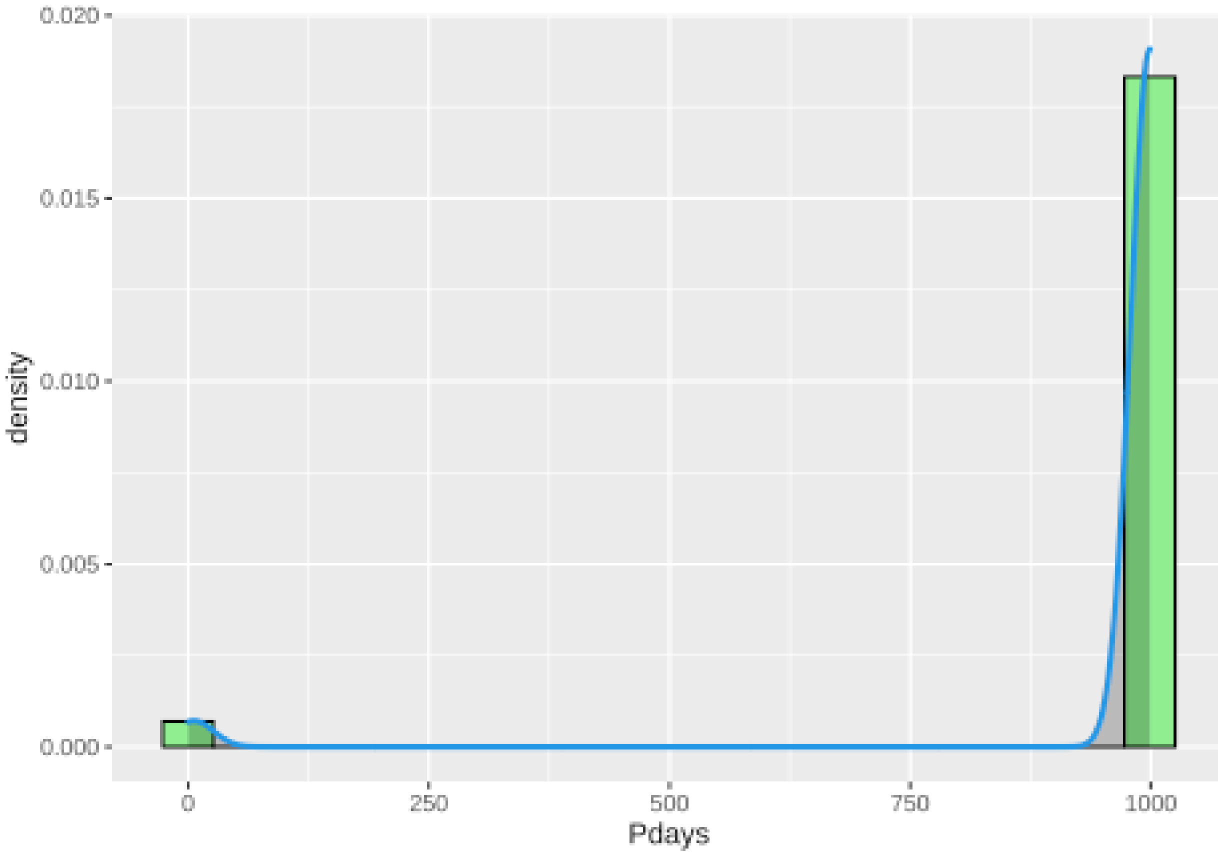
```
distplotsnumerics(bank,numerics[,2:4])
```



0.003
0.002
0.001
0.000
density
0
1000
2000
3000
4000
5000
Duration

1.00
0.75
0.50
0.25
0.00
density
0
20
40
Campaign




Last 50 columns in descending order of Duration

```
head(bank[order(bank$Duration, decreasing= T),], n = 50)
```

```
##       Age           Job  Marital           Education Default Housing    Loan
## 23950  33    technician   single professional.course      no     yes      no
## 22071  52   blue-collar  married            basic.4y      no      no      no
## 40342  27 administration  single         high.school      no      no      no
## 13742  31    technician  married professional.course      no      no      no
## 7680   37    unemployed  married professional.course      no     yes      no
## 35868  28   blue-collar  married            basic.9y      no     yes      no
## 19518  47    management  married         high.school      no      no      no
## 2294   39  self-employed married            basic.4y unknown     yes      no
## 20876  47 administration married         high.school      no     yes     yes
## 23903  27   blue-collar   single professional.course      no     yes      no
## 23864  46 administration divorced        high.school      no     yes      no
## 11951  58       retired  married         high.school      no     yes      no
## 6246   30  self-employed married            basic.9y      no      no      no
## 4183   42    management  married            basic.6y unknown     yes      no
## 27685  28  self-employed  single   university.degree      no     yes     yes
## 29135  40     housemaid  married            basic.6y unknown unknown unknown
## 27861  54   blue-collar  married professional.course      no      no      no
## 10394  47   blue-collar  married            basic.4y      no     yes      no
## 18182  32 administration married   university.degree      no     yes      no
## 3746   35       student   single         high.school      no      no     yes
## 9221   47 administration divorced  university.degree      no     yes      no
## 11291  26    technician   single   university.degree      no      no      no
## 10643  30  self-employed  single   university.degree      no     yes      no
```

```
## 17709  33 administration  married   university.degree      no      no      no
## 38981  53 administration divorced   university.degree      no      no      no
## 1672   26 administration  married         high.school      no     yes     yes
## 24259  34   entrepreneur  married   university.degree unknown     yes      no
## 8312   40    blue-collar  married            basic.9y      no      no     yes
## 26851  30    blue-collar  married         high.school      no     yes     yes
## 23868  35   entrepreneur  married   university.degree      no     yes     yes
## 26792  33     management   single   university.degree      no      no      no
## 23484  42      housemaid  married   university.degree      no     yes     yes
## 28016  57    blue-collar  married         high.school      no     yes     yes
## 35386  33    blue-collar   single         high.school      no      no      no
## 28054  34 administration   single         high.school      no     yes     yes
## 7258   53       services  married         high.school unknown      no      no
## 2311   38    blue-collar   single            basic.9y      no     yes      no
## 36332  39 administration  married   university.degree      no      no      no
## 8592   25 administration   single   university.degree      no      no      no
## 19200  39 administration   single   university.degree      no     yes      no
## 38201  59      housemaid  married            basic.4y      no      no      no
## 38618  32 administration   single   university.degree      no     yes      no
## 29219  48 administration divorced         high.school      no      no     yes
## 28137  29    blue-collar   single         high.school      no     yes     yes
## 13059  35  self-employed   single   university.degree      no     yes      no
## 10946  59        retired  married            basic.9y      no     yes      no
## 23101  29       services  married professional.course unknown     yes      no
## 1381   31       services  married            basic.6y      no      no      no
## 16551  21    blue-collar  married            basic.9y      no     yes      no
## 38885  43     management  married   university.degree      no     yes      no
##         Contact Month Last.Contact.Day Duration Campaign Pdays
## 23950 telephone   nov              mon     4918        1   999
## 22071 telephone   aug              thu     4199        3   999
## 40342 telephone   aug              fri     3785        1   999
## 13742  cellular   jul              thu     3643        1   999
## 7680  telephone   may              fri     3631        2   999
## 35868  cellular   may              tue     3509        2     3
## 19518  cellular   aug              thu     3422        1   999
## 2294  telephone   may              tue     3366        3   999
## 20876  cellular   aug              thu     3322        1   999
## 23903 telephone   oct              mon     3284        1   999
## 23864 telephone   oct              fri     3253        1   999
## 11951 telephone   jun              thu     3183        2   999
## 6246  telephone   may              tue     3094        2   999
## 4183  telephone   may              mon     3078        4   999
## 27685  cellular   mar              fri     3076        1   999
## 29135  cellular   apr              fri     2926        2   999
## 27861  cellular   apr              thu     2870        2   999
## 10394 telephone   jun              mon     2769        4   999
## 18182 telephone   jul              wed     2692        8   999
## 3746  telephone   may              fri     2680        1   999
## 9221  telephone   jun              fri     2653        3   999
## 11291 telephone   jun              thu     2635        3   999
## 10643 telephone   jun              tue     2621        3   999
## 17709  cellular   jul              tue     2516        1   999
## 38981  cellular   mar              thu     2486        1   999
## 1672  telephone   may              fri     2462        1   999
```

```
## 24259  cellular   nov             mon     2462        2   999
## 8312  telephone   jun             tue     2456        2   999
## 26851  cellular   nov             thu     2453        2   999
## 23868 telephone   oct             mon     2429        1   999
## 26792  cellular   nov             thu     2420        3   999
## 23484  cellular   aug             thu     2372        3   999
## 28016  cellular   apr             mon     2316        1   999
## 35386  cellular   may             mon     2301        1   999
## 28054  cellular   apr             tue     2299        2   999
## 7258  telephone   may             thu     2260        2   999
## 2311  telephone   may             tue     2231        1   999
## 36332  cellular   jun             wed     2219        1   999
## 8592  telephone   jun             wed     2203        2   999
## 19200  cellular   aug             wed     2191        1   999
## 38201 telephone   oct             tue     2187        1   999
## 38618 telephone   nov             fri     2184        2   999
## 29219  cellular   apr             fri     2139        3   999
## 28137  cellular   apr             wed     2129        1   999
## 13059  cellular   jul             wed     2122        1   999
## 10946 telephone   jun             wed     2093        1   999
## 23101  cellular   aug             tue     2089        6   999
## 1381  telephone   may             thu     2087        2   999
## 16551  cellular   jul             wed     2078        6   999
## 38885 telephone   dec             mon     2062        2     8
##       Previous.Contacts    Poutcome Emp.var.rate Cons.price.idx Cons.conf.idx
## 23950                 0 nonexistent         -0.1         93.200         -42.0
## 22071                 0 nonexistent          1.4         93.444         -36.1
## 40342                 0 nonexistent         -1.7         94.027         -38.3
## 13742                 0 nonexistent          1.4         93.918         -42.7
## 7680                  0 nonexistent          1.1         93.994         -36.4
## 35868                 2     success         -1.8         92.893         -46.2
## 19518                 0 nonexistent          1.4         93.444         -36.1
## 2294                  0 nonexistent          1.1         93.994         -36.4
## 20876                 0 nonexistent          1.4         93.444         -36.1
## 23903                 0 nonexistent         -0.1         93.798         -40.4
## 23864                 0 nonexistent         -0.1         93.798         -40.4
## 11951                 0 nonexistent          1.4         94.465         -41.8
## 6246                  0 nonexistent          1.1         93.994         -36.4
## 4183                  0 nonexistent          1.1         93.994         -36.4
## 27685                 0 nonexistent         -1.8         92.843         -50.0
## 29135                 0 nonexistent         -1.8         93.075         -47.1
## 27861                 0 nonexistent         -1.8         93.075         -47.1
## 10394                 0 nonexistent          1.4         94.465         -41.8
## 18182                 0 nonexistent          1.4         93.918         -42.7
## 3746                  0 nonexistent          1.1         93.994         -36.4
## 9221                  0 nonexistent          1.4         94.465         -41.8
## 11291                 0 nonexistent          1.4         94.465         -41.8
## 10643                 0 nonexistent          1.4         94.465         -41.8
## 17709                 0 nonexistent          1.4         93.918         -42.7
## 38981                 0 nonexistent         -1.8         93.369         -34.8
## 1672                  0 nonexistent          1.1         93.994         -36.4
## 24259                 0 nonexistent         -0.1         93.200         -42.0
## 8312                  0 nonexistent          1.4         94.465         -41.8
## 26851                 0 nonexistent         -0.1         93.200         -42.0
```

```
## 23868                    0 nonexistent          -0.1           93.798           -40.4
## 26792                    0 nonexistent          -0.1           93.200           -42.0
## 23484                    0 nonexistent           1.4           93.444           -36.1
## 28016                    0 nonexistent          -1.8           93.075           -47.1
## 35386                    0 nonexistent          -1.8           92.893           -46.2
## 28054                    0 nonexistent          -1.8           93.075           -47.1
## 7258                     0 nonexistent           1.1           93.994           -36.4
## 2311                     0 nonexistent           1.1           93.994           -36.4
## 36332                    1     failure          -2.9           92.963           -40.8
## 8592                     0 nonexistent           1.4           94.465           -41.8
## 19200                    0 nonexistent           1.4           93.444           -36.1
## 38201                    0 nonexistent          -3.4           92.431           -26.9
## 38618                    1     failure          -3.4           92.649           -30.1
## 29219                    0 nonexistent          -1.8           93.075           -47.1
## 28137                    1     failure          -1.8           93.075           -47.1
## 13059                    0 nonexistent           1.4           93.918           -42.7
## 10946                    0 nonexistent           1.4           94.465           -41.8
## 23101                    0 nonexistent           1.4           93.444           -36.1
## 1381                     0 nonexistent           1.1           93.994           -36.4
## 16551                    0 nonexistent           1.4           93.918           -42.7
## 38885                    1     success          -3.0           92.713           -33.0
##       Euribor3m Employment.number   y
## 23950     4.406            5195.8  no
## 22071     4.963            5228.1 yes
## 40342     0.888            4991.6  no
## 13742     4.963            5228.1 yes
## 7680      4.864            5191.0 yes
## 35868     1.266            5099.1  no
## 19518     4.968            5228.1  no
## 2294      4.856            5191.0  no
## 20876     4.964            5228.1  no
## 23903     4.912            5195.8  no
## 23864     5.045            5195.8  no
## 11951     4.955            5228.1 yes
## 6246      4.857            5191.0 yes
## 4183      4.858            5191.0  no
## 27685     1.640            5099.1 yes
## 29135     1.405            5099.1 yes
## 27861     1.483            5099.1  no
## 10394     4.960            5228.1 yes
## 18182     4.963            5228.1 yes
## 3746      4.859            5191.0 yes
## 9221      4.967            5228.1 yes
## 11291     4.961            5228.1  no
## 10643     4.961            5228.1 yes
## 17709     4.961            5228.1 yes
## 38981     0.654            5008.7 yes
## 1672      4.855            5191.0  no
## 24259     4.191            5195.8 yes
## 8312      4.864            5228.1 yes
## 26851     4.076            5195.8 yes
## 23868     5.000            5195.8  no
## 26792     4.076            5195.8 yes
## 23484     4.962            5228.1 yes
```

```
## 28016      1.466            5099.1  no
## 35386      1.244            5099.1 yes
## 28054      1.453            5099.1 yes
## 7258       4.860            5191.0  no
## 2311       4.856            5191.0 yes
## 36332      1.260            5076.2  no
## 8592       4.864            5228.1 yes
## 19200      4.967            5228.1  no
## 38201      0.737            5017.5  no
## 38618      0.714            5017.5 yes
## 29219      1.405            5099.1 yes
## 28137      1.445            5099.1  no
## 13059      4.962            5228.1 yes
## 10946      4.962            5228.1 yes
## 23101      4.965            5228.1 yes
## 1381       4.855            5191.0 yes
## 16551      4.963            5228.1 yes
## 38885      0.709            5023.5 yes
```

#insights - The Duration and Campaign variables are strongly positively skewed, while pdays is strongly negatively skewed. Interesting to note that whenever duration is on the longer end (higher value), then the client has not been contacted prior to this campaign as well (pdays=999) and thus poutcome is "nonexistent". This is observable from the table above that shows the dataset in descending order by Duration. This observation can be understood in the context that when clients have not been contacted before then more time would be required to introduce the purpose of the current call, build trust, relay the required information regarding the current campaign, etc and thus the duration of the call will be longer. On the contrary, when clients have been contacted previously then relatively less time will be required to explain the purpose of the call as the customer will be familiar with such calls from previous experience.

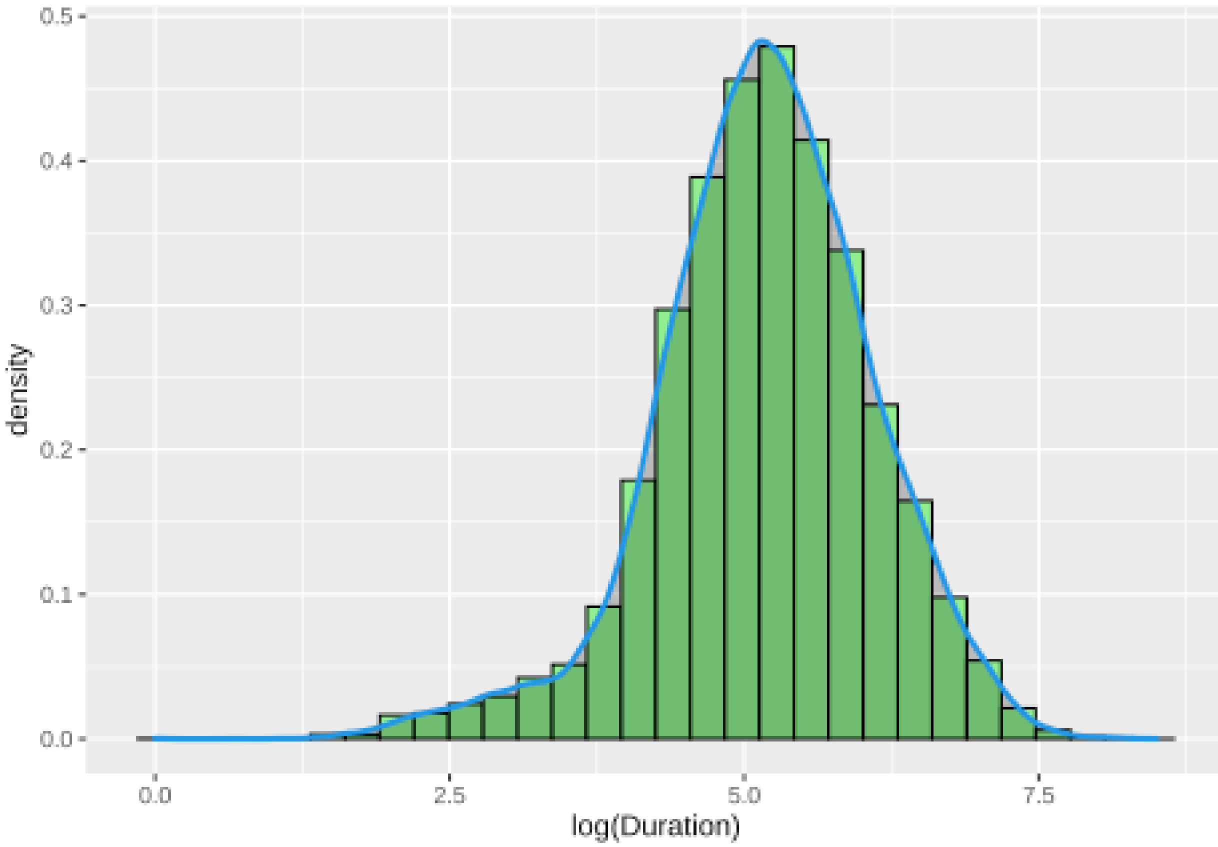
## Log transformation for right skewed features

```r
logpn1 <-ggplot(numerics, aes(x = log(Duration))) +
  geom_histogram(aes(y = ..density..),
                 colour = 1, fill = "lightgreen") +
  geom_density(lwd = 1, colour = 4,
               fill = 1, alpha = 0.25, bins = 20)
```

```
## Warning: Ignoring unknown parameters: bins
```

```r
logpn1
```

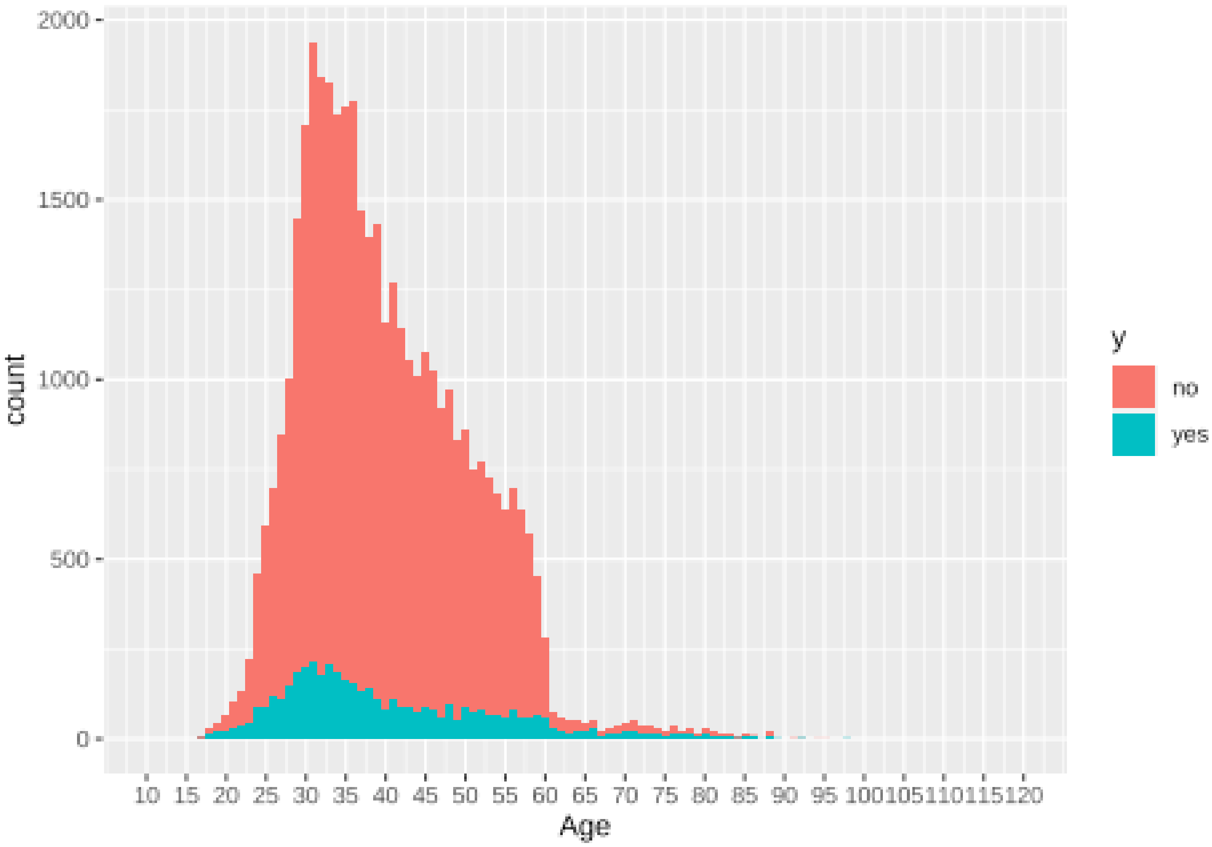
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 4 rows containing non-finite values (stat_bin).

## Warning: Removed 4 rows containing non-finite values (stat_density).
```




A common strategy to fix skewness is use log tranformation. The graph above shows the duration column after being log transformed. Eventhough, a log transform makes the data distribution more normally distributed it did provide any significant improvement over the non-log transformed data in our testing. This is most likely due to the fact that in regression models there are no assumptions made about the distribution shape of the independent variable. Especially in logistic regression, a log transformation of the independent variable would make it make it difficult to interpret the odds ratio of the dependent variable as it is a per-unit change of the independent variable. For example, for each additional log-unit of x (duration), the output of y increases by xx amount. Therefore, we did not use log transformations and instead used the original data.

### Overlay densityplots for age

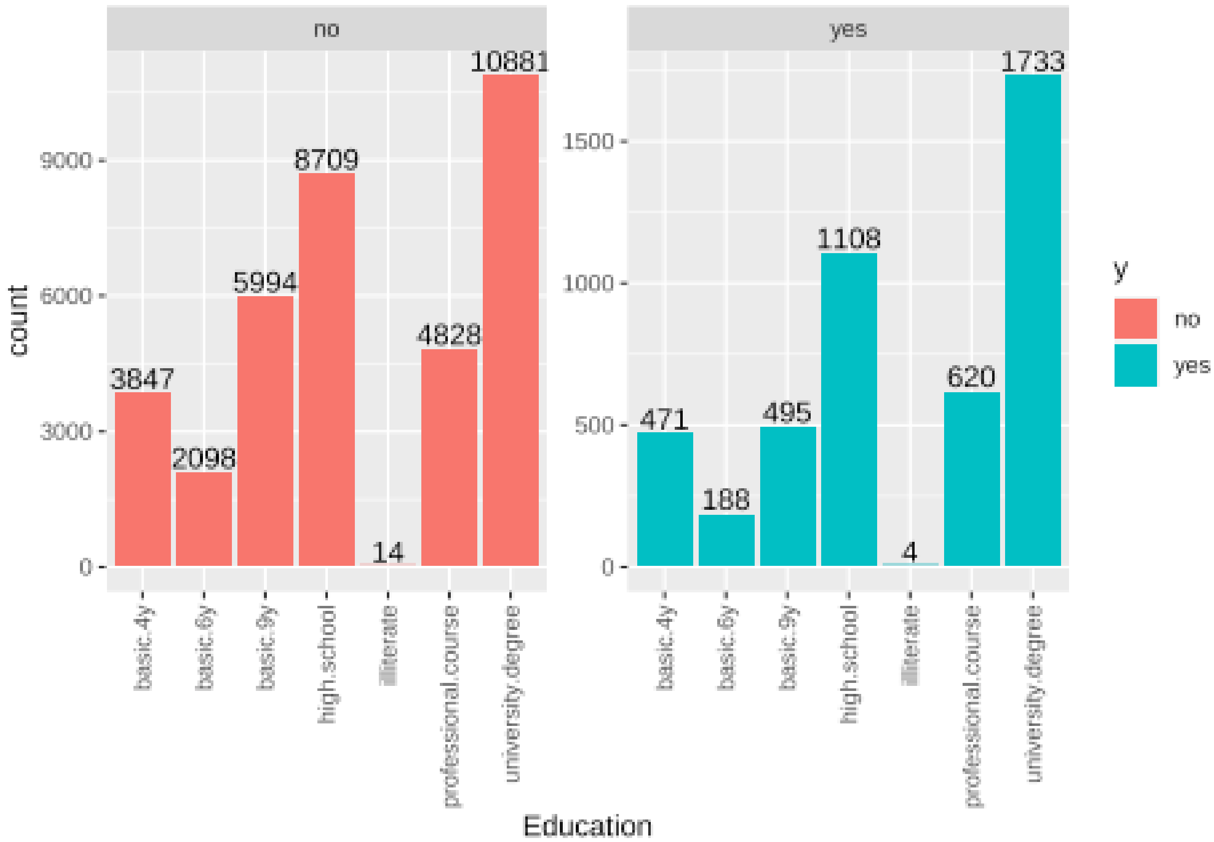
```
ggplot(bank,aes(x=Age, fill=y)) + geom_histogram(binwidth = 1) + scale_x_continuous(breaks = seq(10, 12(
```



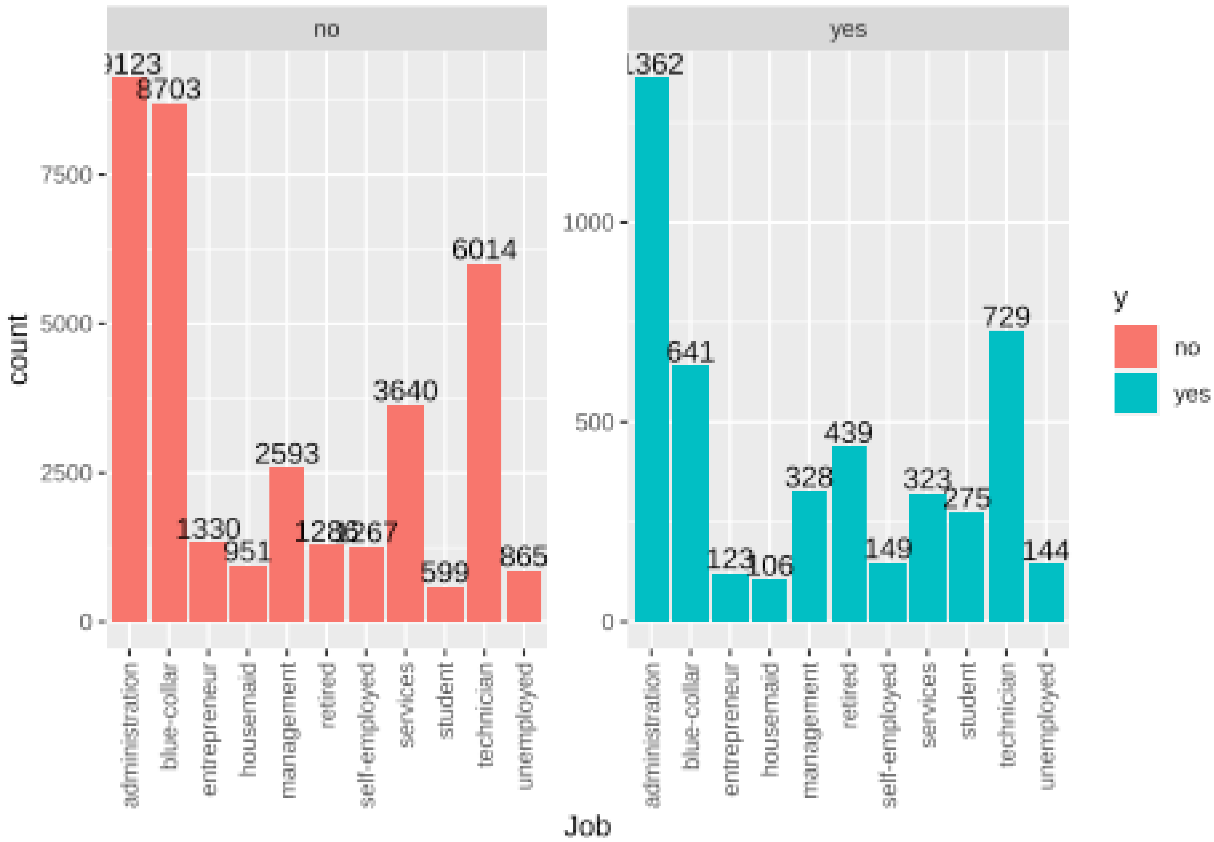

The overlay of the two histograms of both the subscribed (yes) and unsubscribed (no) clients against the age variable illustrates a big gap in the count of each age group among 'yes' and 'no'. There are considerably less clients who subscribed to the term deposit than those that declined in each age category. It can be observed that the age group that was cotacted the most frequently, in general, also responded positively to subscribing to a term deposit - the highest bin for the 'yes' respondents corresponds with the bin (age) that was contacted the most.

FOR THIS: MONTH AND DAY AS WELL Job vs y and Education vs Y

```
ggplot(bank,aes(x=Education, fill=y)) + geom_bar(position = "dodge") + geom_text(aes(label=..count..),s
```



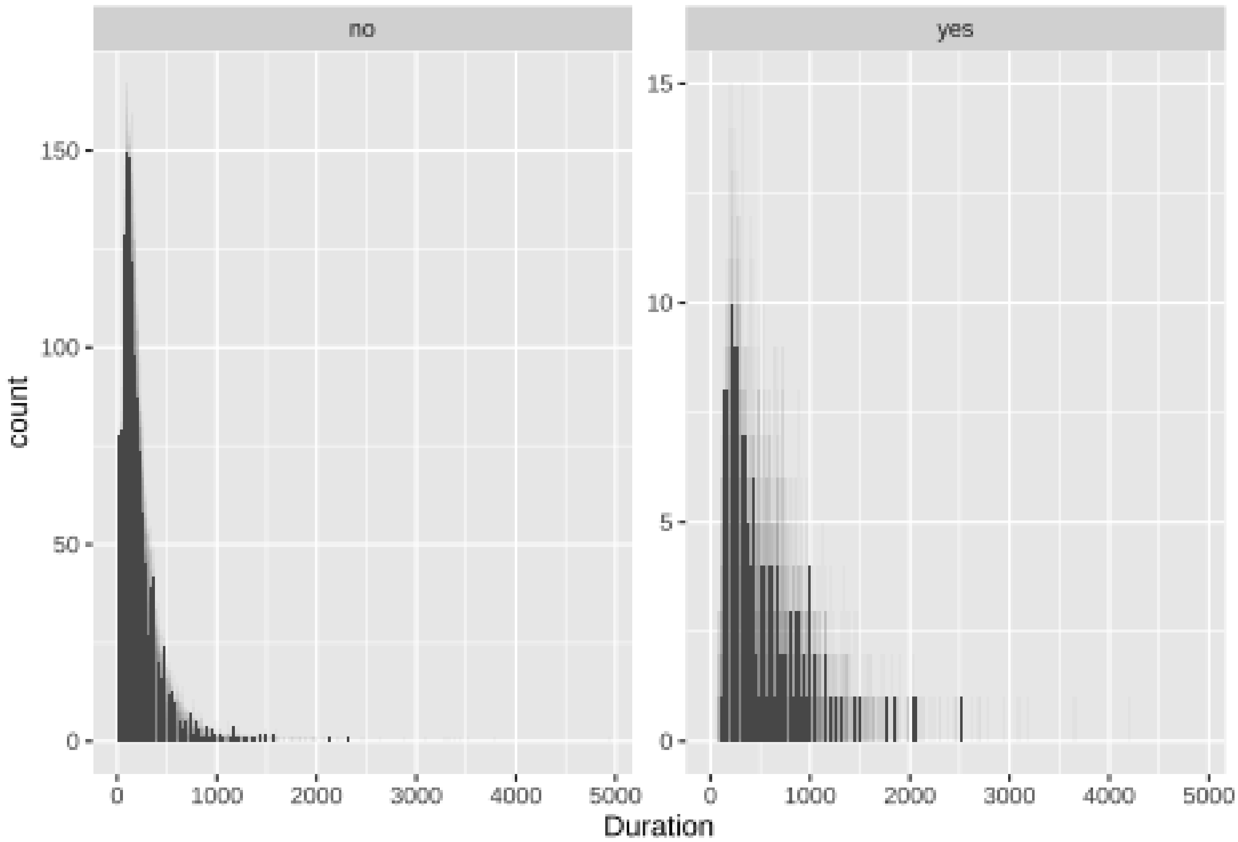

```
ggplot(bank,aes(x=Job, fill=y)) + geom_bar() + geom_bar(position = "dodge") + geom_text(aes(label=..cou
```




The bar plots of Jobs and Education vs y-output shows that the categories are relatively balanced in these 2 columns. For the education plot, the 'illiterate' category is only 14 for for the 'no' respondents and 4 for the 'yes' respondents. Since it is significantly lower than the other categories, 'illiterate' rows were removed from the dataset. A general trend can be seen that the higher the education level, the more likely they are to be contacted by the bank and are also more likely to respond in a positive manner. This trend is seen in both education and job columns. Except for the retired category and services category in the job column. Compared to the other categories, the retired category was contacted less but a higher proportion of them responded yes. The contrary is true for the services category.

Duration has lot of outliers

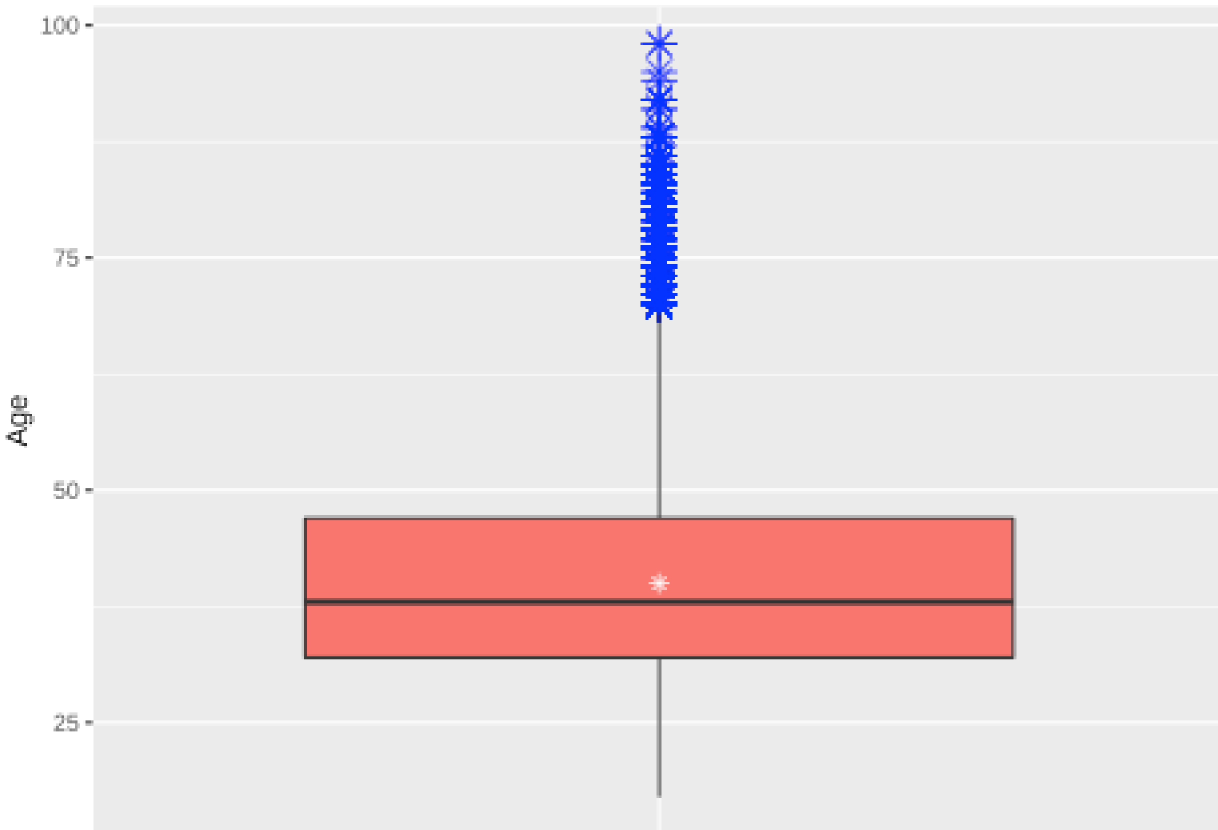
```
ggplot(bank,aes(x=Duration)) + geom_bar() + facet_wrap(~y, scales = "free_y") + scale_color_brewer(pale
```



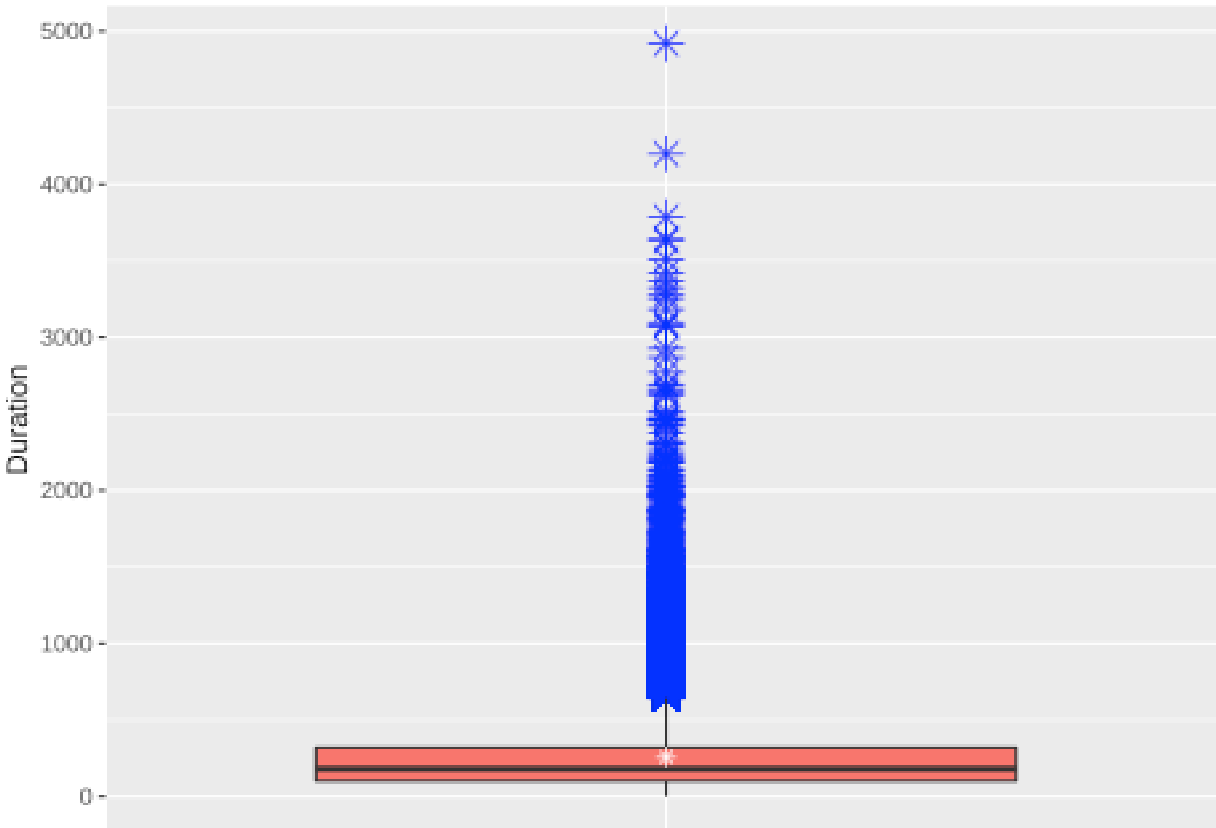

In the bar chart of the duration column, it can be noted that when clients refuse to subscribe to a term deposit then the call duration are mostly towards the low end. This can be explained that when clients are certain they don't want to subscribe to the term deposit then refuse early on in the call and the call ends. However, when clients ended-up successfully subscribing to term deposits then it can be seen that call durations are longer on average - observed from the slightly less positively skewed chart of the 'yes' respondents when compared to the chart of the 'no' respondent. This is mostly due to the extra time needed to either clarify or convince a client regarding the details of the term deposit.

#Boxplots

```
bp1 <- ggplot(numerics, aes(x = factor(0), y = Age, fill = factor(0))) +
  geom_boxplot(outlier.colour="blue", outlier.shape=8, outlier.size=4) +
  theme(axis.title.x=element_blank(),axis.text.x=element_blank(),axis.ticks.x=element_blank(),legend.po
  stat_summary(fun = "mean", geom = "point", shape = 8,
               size = 2, color = "white")
bp1
```




```
bp2 <- ggplot(numerics, aes(x = factor(0), y = Duration, fill = factor(0))) +
  geom_boxplot(outlier.colour="blue", outlier.shape=8, outlier.size=4) +
  theme(axis.title.x=element_blank(),axis.text.x=element_blank(),axis.ticks.x=element_blank(),legend.pos
  stat_summary(fun = "mean", geom = "point", shape = 8,
               size = 2, color = "white")
bp2
```

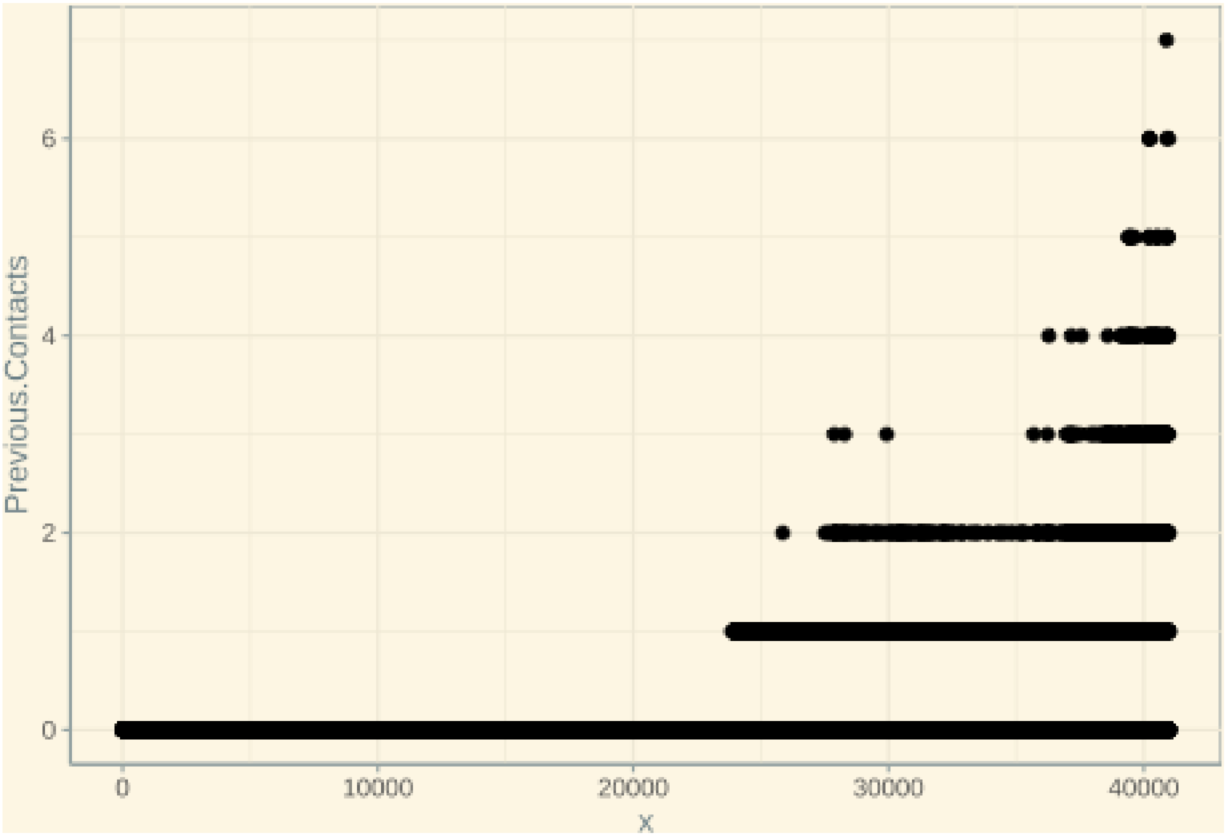



The boxplots Age and Duration show that both columns have a lot of outliers present. It is especially exhibited in Duration where the interquartile range is very small with a range of about 200. While the outliers can go all the way to around 5000s. We experimented with removing the outliers and running the models but they yielded no significant results. Thus, it was decided to keep the outliers as most of them were in columns that were related directly to the telemarketing campaign such as pdays, previous, campaign, and duration and most probably hold important information.

collecting integer columns

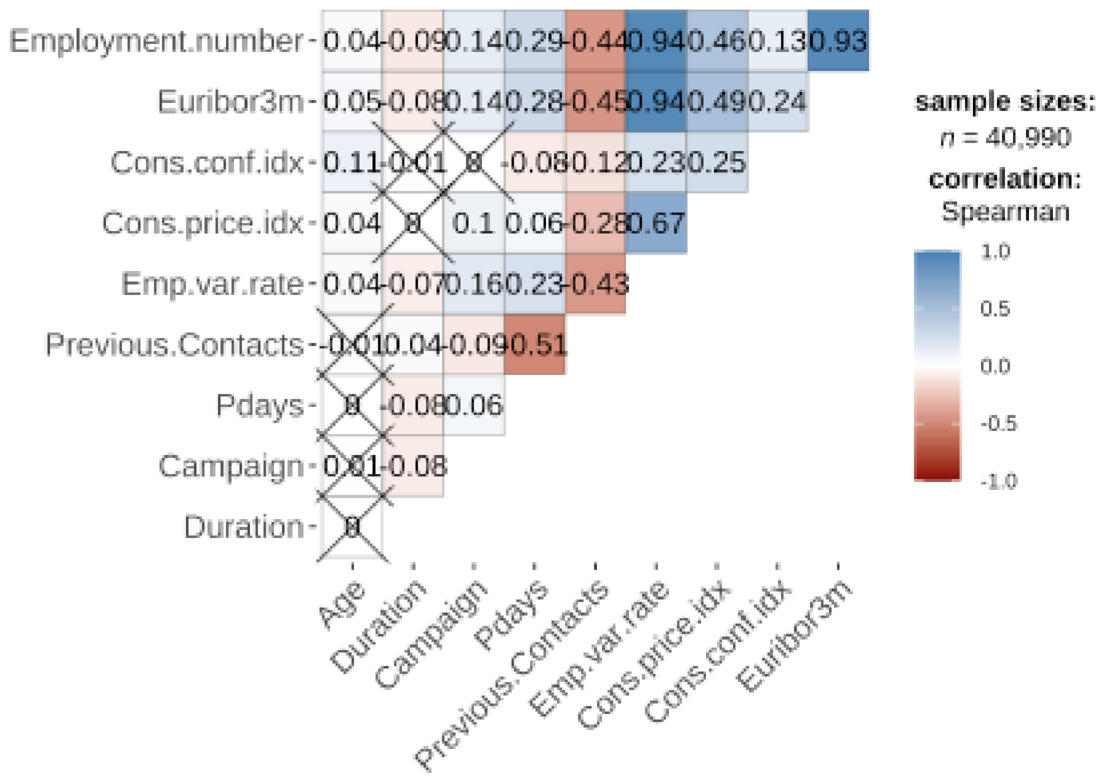
```
integers <- subset(bank,select = names(Filter(is.integer,bank)))
```

```
x <- c(1:nrow(bank))
ggplot(integers, aes(x=x, y=Previous.Contacts)) +
    geom_point(size=2) +
    theme_solarized()
```




**Correlation numerical columns:** Parametric for Pearson, nonparametric for Spearman's correlation

```
ggstatsplot::ggcorrmat(data = bank, type = "nonparametric", colors = c("darkred", "white", "steelblue"))
```




X = non-significant at $p < 0.05$ (Adjustment: Holm)

In the correleation matrix above, the non-significant correlations (by default at the 5% significance level with the Holm adjustment method) are shown by a cross on the correlation coefficients. Looking at the crosses, its clear there is no significant correlation between Age and Duration, Campaign, Pdays, and Previous.contacts. Interestingly there are very strong correlations between the social and economic data. The Strongest positive correlation of 0.93 is between Euribor3m and Employment.number. A strong correlation between independent variables may cause a multicollinearity problem which might make our models sensitive to small changes. It makes is difficult for the model to estimate the relationship between the dependent variable and the independent variables in an independent manner because the correlated independent variables will tend to change in unison. However, it is not always a problem and is discussed further in the models section.

#Statistical Analysis ##chi-square statistical test for correlation of categorical features

```
summary(bank[Factorcols])
```

```
##               Job            Marital         Default
##  administration:10485   divorced: 4611   no     :32469
##  blue-collar   : 9344   married :24824   unknown: 8518
##  technician    : 6743   single  :11555   yes    :    3
##  services      : 3963
##  management    : 2921
##  retired       : 1725
##  (Other)       : 5809
##                 Education         Housing           Loan             Contact
##  basic.4y           : 4318   no     :18526   no     :33782   cellular :26032
##  basic.6y           : 2286   unknown:   987   unknown:   987   telephone:14958
##  basic.9y           : 6489   yes    :21477   yes    : 6221
##  high.school        : 9817
##  illiterate         :   18
##  professional.course: 5448
```

```
##  university.degree  :12614
##      Month       Last.Contact.Day        Poutcome       y
##  may    :13699   fri:7796         failure    : 4236   no :36371
##  jul    : 7148   mon:8467         nonexistent:35391   yes: 4619
##  aug    : 6135   thu:8570         success    : 1363
##  jun    : 5284   tue:8056
##  nov    : 4092   wed:8101
##  apr    : 2627
##  (Other): 2005
```

### testing relationship between Factor columns and y(subscribed or unsubscribed)

measuring the effect size

```
cramersv <- function(x,n,d){
  v <- sqrt(x/(n*d))
  return(v)
}
```

Effect size interpretation for cramersv for df=1: small(.1), medium(.3), large(.5) H0: There is no relationship(Independent) H1: There is a relationship(dependent)

```
chisquare_test <- function(bank,Factorcols)
{  n <- nrow(bank)
  for(col in Factorcols[-length(Factorcols)]){
    if(col!="Default")
    {
      print(paste("*",col,"*"))
      print(table(bank[[col]],bank$y))
      chi <- chisq.test(bank[[col]],bank$y)
      print(chi)
      print(paste("effect size:",cramersv(chi$statistic,n,chi$parameter)))
    }
  }
}
```

```
chisquare_test(bank,Factorcols)
```

```
## [1] "* Job *"
##
##                   no  yes
##   administration 9123 1362
##   blue-collar    8703  641
##   entrepreneur   1330  123
##   housemaid       951  106
##   management     2593  328
##   retired        1286  439
##   self-employed  1267  149
##   services       3640  323
##   student         599  275
##   technician     6014  729
##   unemployed      865  144
##
##  Pearson's Chi-squared test
##
## data:  bank[[col]] and bank$y
```

```
## X-squared = 979.04, df = 10, p-value < 2.2e-16
##
## [1] "effect size: 0.0488721275382095"
## [1] "* Marital *"
##
##               no   yes
##   divorced  4135   476
##   married  22299  2525
##   single    9937  1618
##
##  Pearson's Chi-squared test
##
## data:  bank[[col]] and bank$y
## X-squared = 120.38, df = 2, p-value < 2.2e-16
##
## [1] "effect size: 0.0383197800632542"
## [1] "* Education *"
##
##                          no   yes
##   basic.4y             3847   471
##   basic.6y             2098   188
##   basic.9y             5994   495
##   high.school          8709  1108
##   illiterate             14     4
##   professional.course  4828   620
##   university.degree   10881  1733

## Warning in chisq.test(bank[[col]], bank$y): Chi-squared approximation may be
## incorrect

##
##  Pearson's Chi-squared test
##
## data:  bank[[col]] and bank$y
## X-squared = 186.96, df = 6, p-value < 2.2e-16
##
## [1] "effect size: 0.0275714940810703"
## [1] "* Housing *"
##
##              no   yes
##   no      16513  2013
##   unknown   880   107
##   yes     18978  2499
##
##  Pearson's Chi-squared test
##
## data:  bank[[col]] and bank$y
## X-squared = 6.0813, df = 2, p-value = 0.0478
##
## [1] "effect size: 0.00861276577315766"
## [1] "* Loan *"
##
##              no   yes
##   no      29951  3831
##   unknown   880   107
```

```
##   yes      5540   681
##
##  Pearson's Chi-squared test
##
## data:  bank[[col]] and bank$y
## X-squared = 0.99884, df = 2, p-value = 0.6069
##
## [1] "effect size: 0.00349055503649505"
## [1] "* Contact *"
##
##                no   yes
##   cellular  22194  3838
##   telephone 14177   781
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  bank[[col]] and bank$y
## X-squared = 860.48, df = 1, p-value < 2.2e-16
##
## [1] "effect size: 0.144887806747072"
## [1] "* Month *"
##
##          no   yes
##   apr  2088   539
##   aug  5486   649
##   dec    92    89
##   jul  6501   647
##   jun  4728   556
##   mar   269   274
##   may 12816   883
##   nov  3677   415
##   oct   401   312
##   sep   313   255
##
##  Pearson's Chi-squared test
##
## data:  bank[[col]] and bank$y
## X-squared = 3078.9, df = 9, p-value < 2.2e-16
##
## [1] "effect size: 0.0913560419150834"
## [1] "* Last.Contact.Day *"
##
##         no  yes
##   fri 6954  842
##   mon 7623  844
##   thu 7531 1039
##   tue 7104  952
##   wed 7159  942
##
##  Pearson's Chi-squared test
##
## data:  bank[[col]] and bank$y
## X-squared = 25.771, df = 4, p-value = 3.519e-05
##
```

```
## [1] "effect size: 0.0125370732017601"
## [1] "* Poutcome *"
##
##                 no   yes
##   failure      3634   602
##   nonexistent 32263  3128
##   success       474   889
##
##  Pearson's Chi-squared test
##
## data:  bank[[col]] and bank$y
## X-squared = 4214.1, df = 2, p-value < 2.2e-16
##
## [1] "effect size: 0.226725131514327"
```

**chi-squared test results** - The test gives the categorical columns are dependent on output-y but, the effect size is not so significant for all the cases except Poutcome

### Education category as illiterate are very few therefore dropping rows with illiterate

1) illiterates are more likely to unsubscribe beacause of the lack of financial knowledge
2) Except Loan and Housing Loan remaining all columns are highly correlated with output

```
bank <- subset(bank, Education!="illiterate")
```

dropping unused factor levels in Education

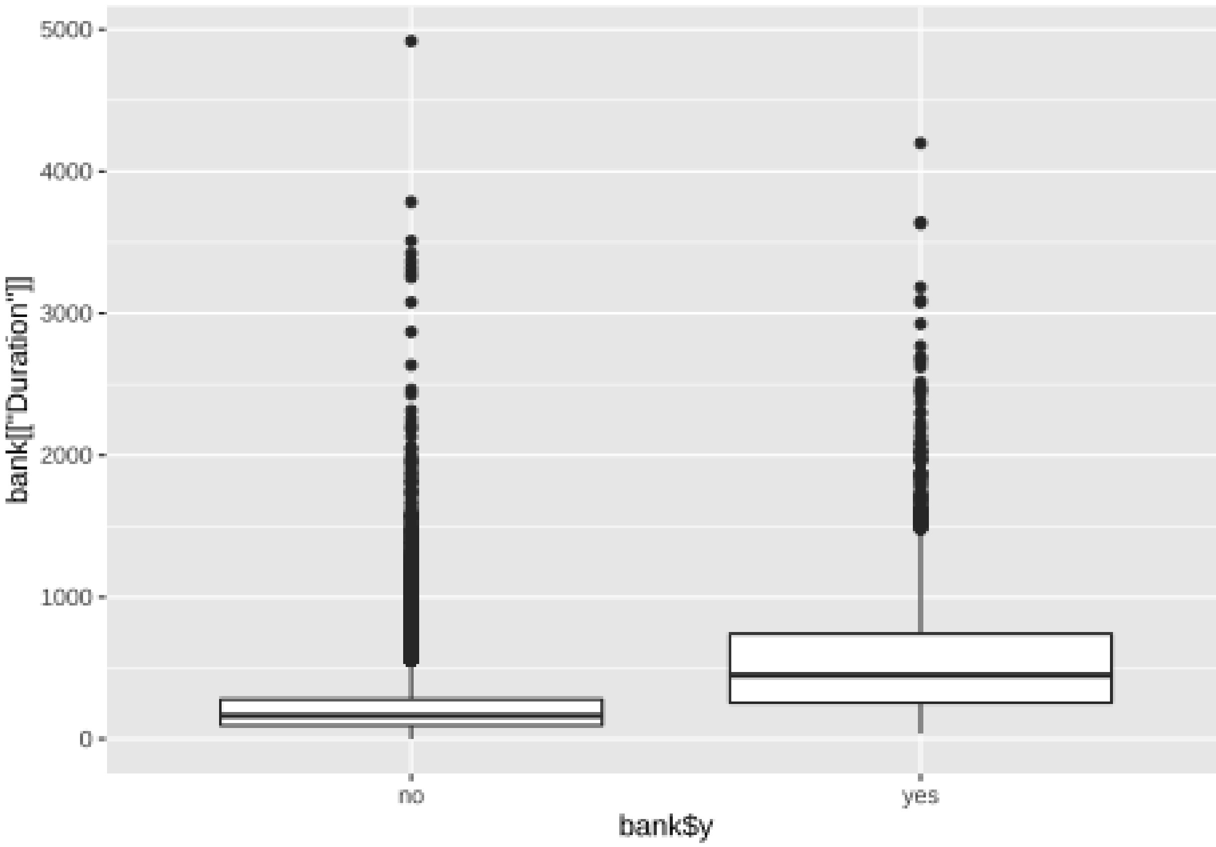
```
bank$Education <- droplevels(bank$Education)
```

```
levels(bank$Education)
```

```
## [1] "basic.4y"            "basic.6y"            "basic.9y"
## [4] "high.school"         "professional.course" "university.degree"
```

### testing correlation between numerical and output

```
graph <- ggplot() + geom_boxplot(aes(bank$y,bank[["Duration"]]))
graph
```




Since, the overlap of boxplot is lesser they are highly correleted with each other which is already given in problem description

##Anova for analysis of variances

```
aov.dur <- aov(Duration~y,data=bank)
summary(aov.dur)
```

```
##                Df    Sum Sq  Mean Sq F value Pr(>F)    
## y               1 4.530e+08 4.53e+08    8061 <2e-16 ***
## Residuals   40970 2.302e+09 5.62e+04                   
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

##Bartlett's test for homogeneity of variances

```
bartlettfornumerics <- function(bank,numerics){
  for(col in names(numerics)){
    print(paste("*",col,"*"))
    print(bartlett.test(bank[[col]],bank$y))
  }
}
```

```
bartlettfornumerics(bank,numerics)
```

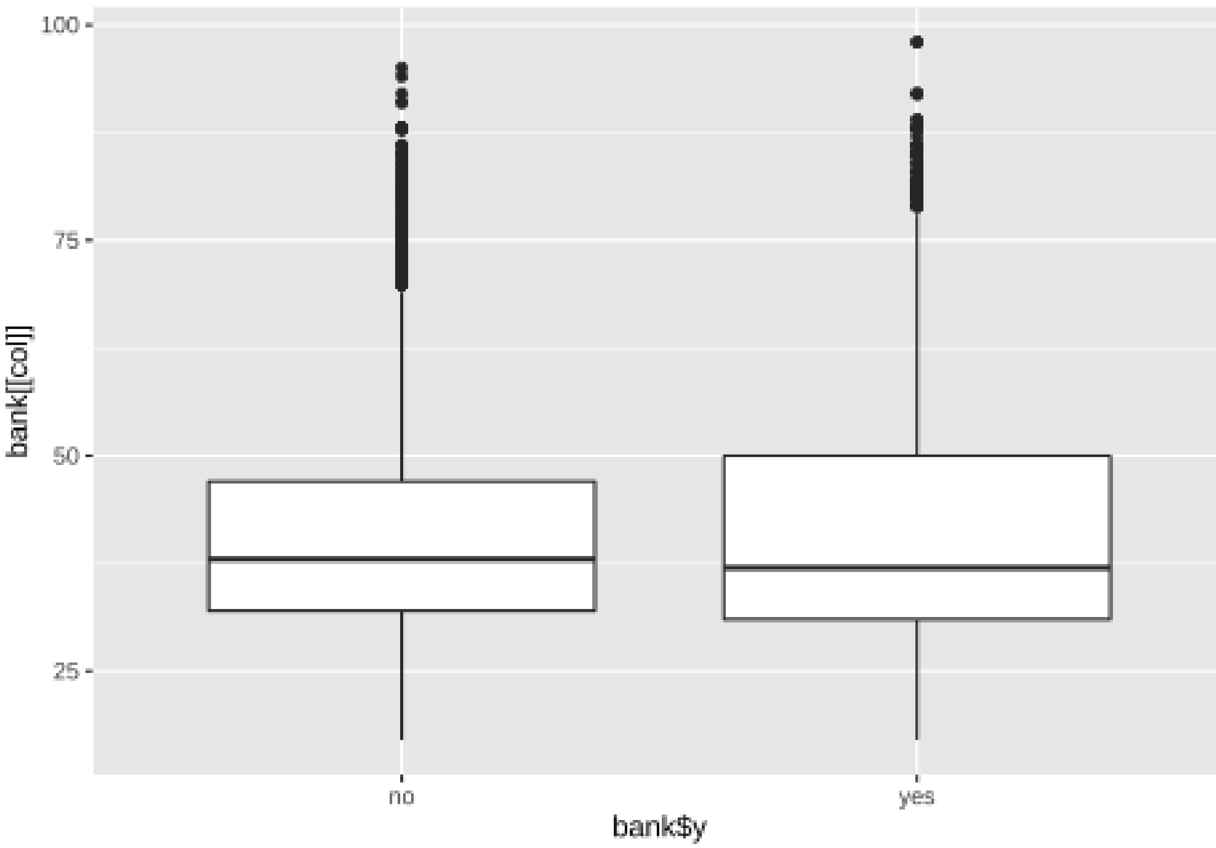
```
## [1] "* Age *"
## 
##  Bartlett test of homogeneity of variances
## 
## data:  bank[[col]] and bank$y
```

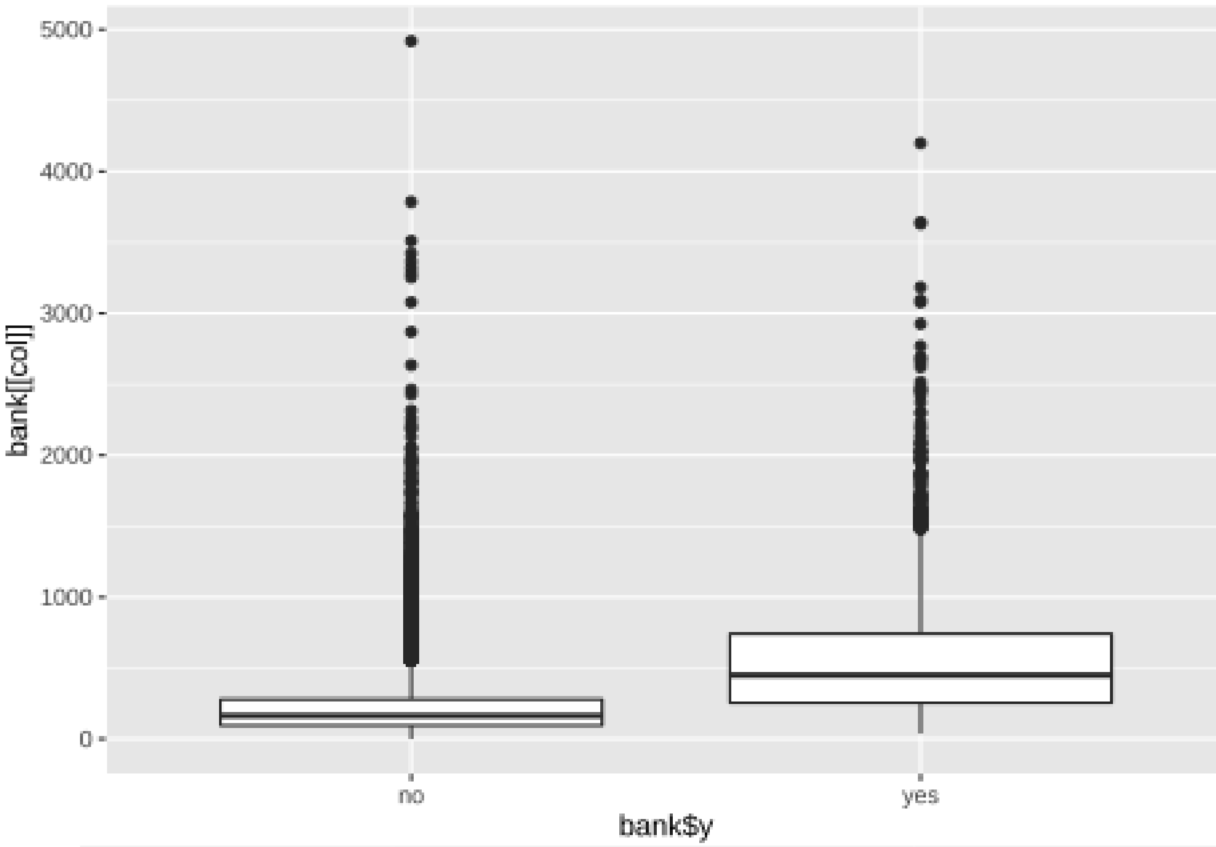
```
## Bartlett's K-squared = 1091.8, df = 1, p-value < 2.2e-16
##
## [1] "* Duration *"
##
##  Bartlett test of homogeneity of variances
##
## data:  bank[[col]] and bank$y
## Bartlett's K-squared = 4962, df = 1, p-value < 2.2e-16
##
## [1] "* Campaign *"
##
##  Bartlett test of homogeneity of variances
##
## data:  bank[[col]] and bank$y
## Bartlett's K-squared = 1828.4, df = 1, p-value < 2.2e-16
##
## [1] "* Pdays *"
##
##  Bartlett test of homogeneity of variances
##
## data:  bank[[col]] and bank$y
## Bartlett's K-squared = 20199, df = 1, p-value < 2.2e-16
##
## [1] "* Previous.Contacts *"
##
##  Bartlett test of homogeneity of variances
##
## data:  bank[[col]] and bank$y
## Bartlett's K-squared = 6479.6, df = 1, p-value < 2.2e-16
##
## [1] "* Emp.var.rate *"
##
##  Bartlett test of homogeneity of variances
##
## data:  bank[[col]] and bank$y
## Bartlett's K-squared = 69.298, df = 1, p-value < 2.2e-16
##
## [1] "* Cons.price.idx *"
##
##  Bartlett test of homogeneity of variances
##
## data:  bank[[col]] and bank$y
## Bartlett's K-squared = 329.12, df = 1, p-value < 2.2e-16
##
## [1] "* Cons.conf.idx *"
##
##  Bartlett test of homogeneity of variances
##
## data:  bank[[col]] and bank$y
## Bartlett's K-squared = 1091.9, df = 1, p-value < 2.2e-16
##
## [1] "* Euribor3m *"
##
##  Bartlett test of homogeneity of variances
```

```
##
## data:  bank[[col]] and bank$y
## Bartlett's K-squared = 31.144, df = 1, p-value = 2.396e-08
##
## [1] "* Employment.number *"
##
##  Bartlett test of homogeneity of variances
##
## data:  bank[[col]] and bank$y
## Bartlett's K-squared = 884.67, df = 1, p-value < 2.2e-16
```

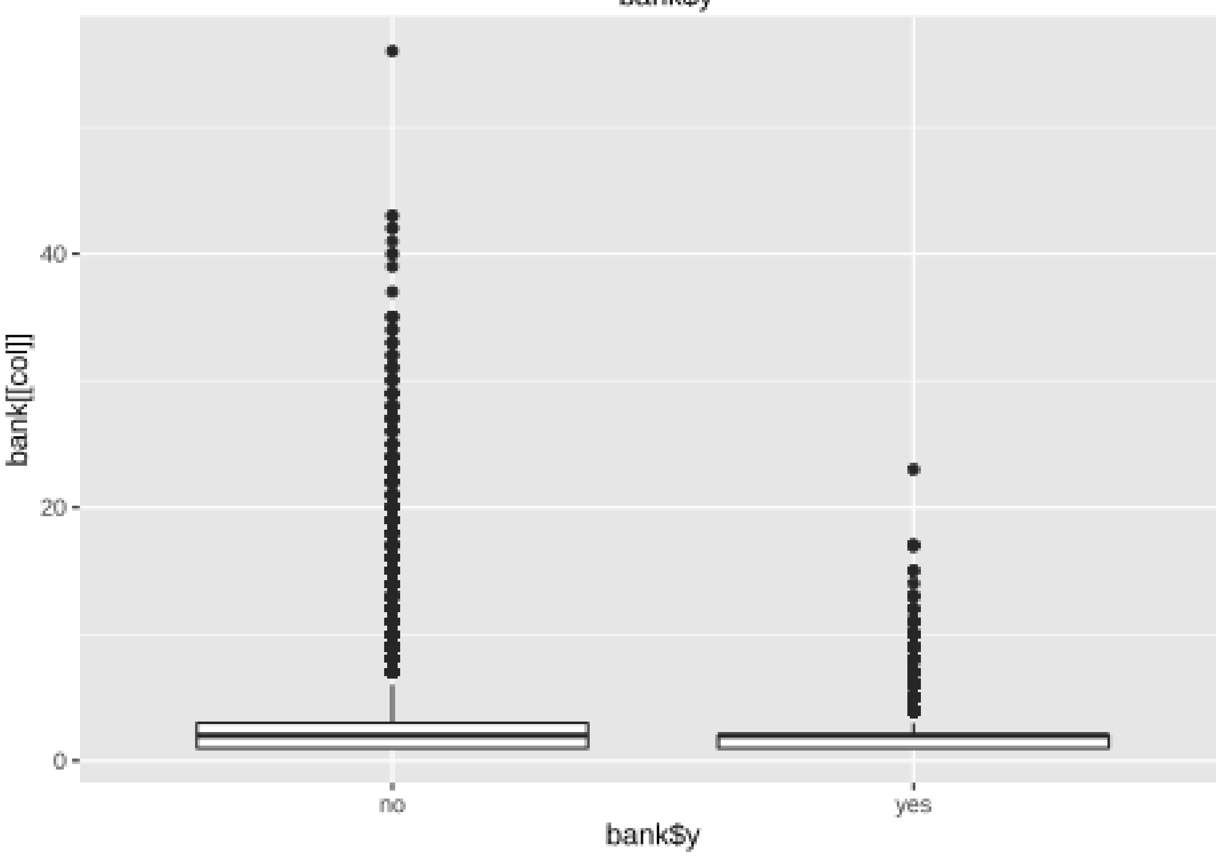
```
boxplotsnumericsvsy <- function(bank,numerics){
  for(col in names(numerics)){
    graph <- ggplot() + geom_boxplot(aes(bank$y,bank[[col]]))
    print(graph)
  }
}
```

```
boxplotsnumericsvsy(bank,numerics)
```

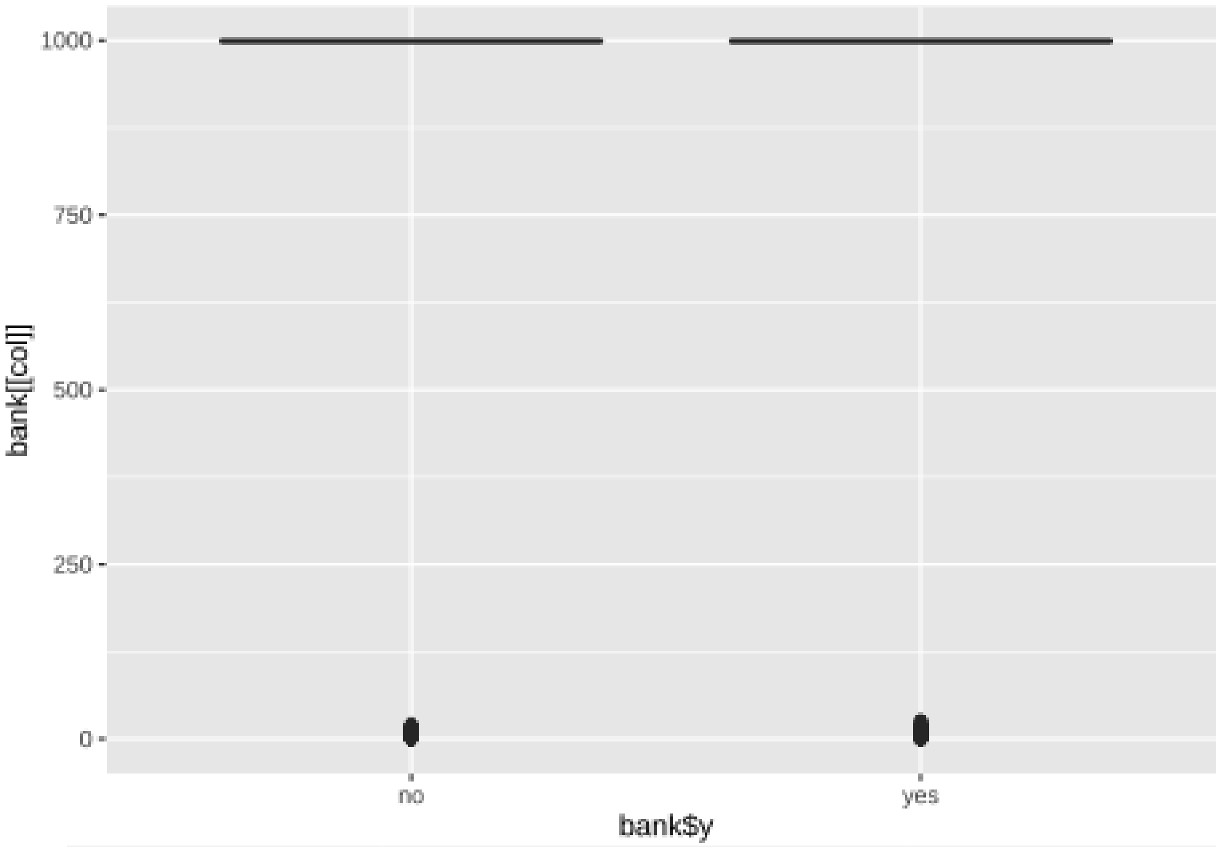




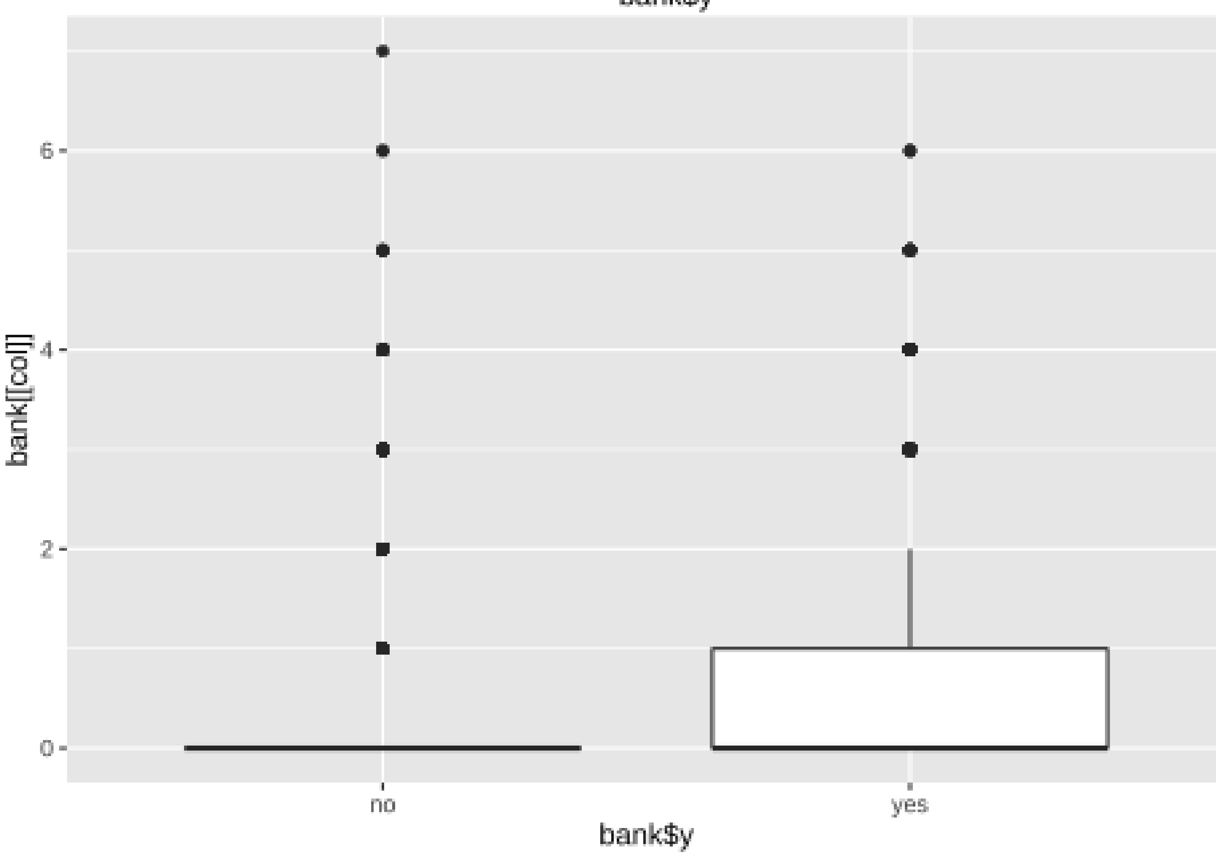
5000
4000
3000
2000
1000
0
bank[[col]]
no
yes
bank$y

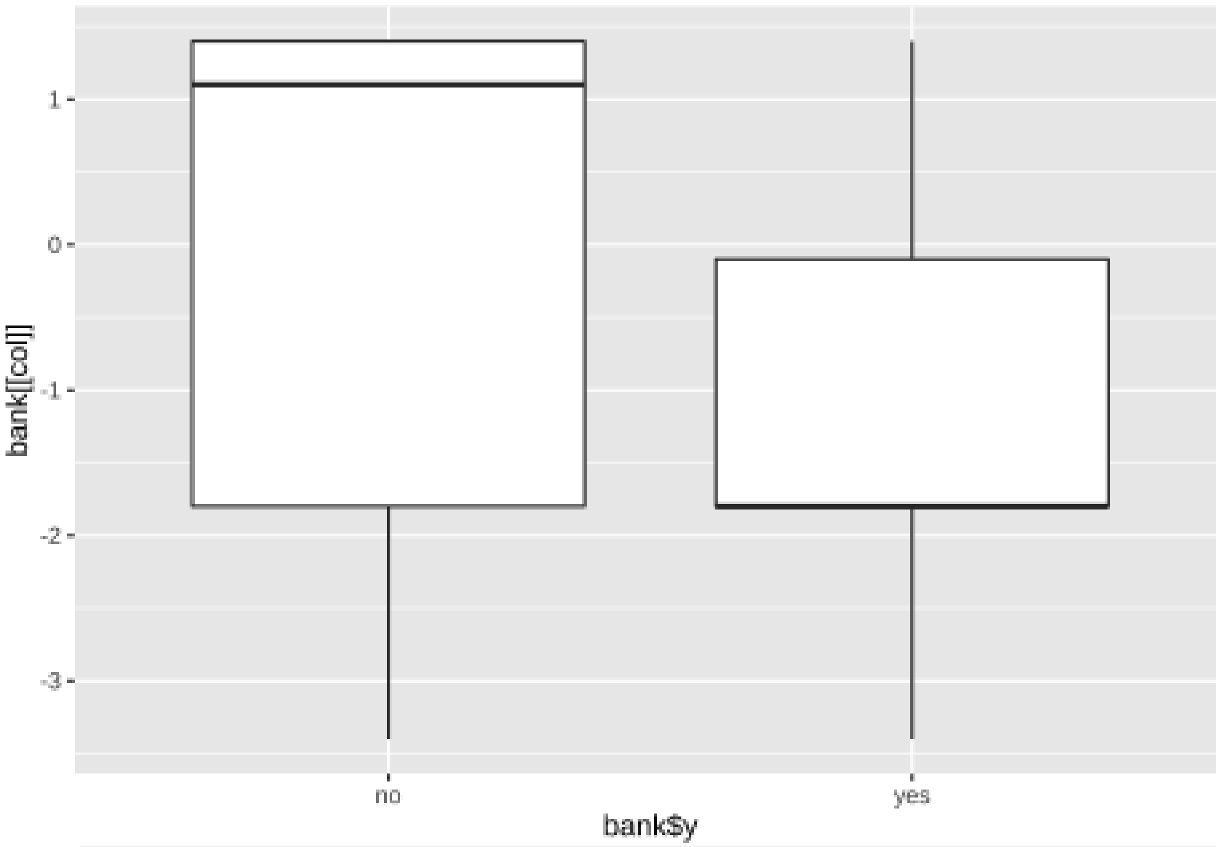
40
20
0
bank[[col]]
no
yes
bank$y

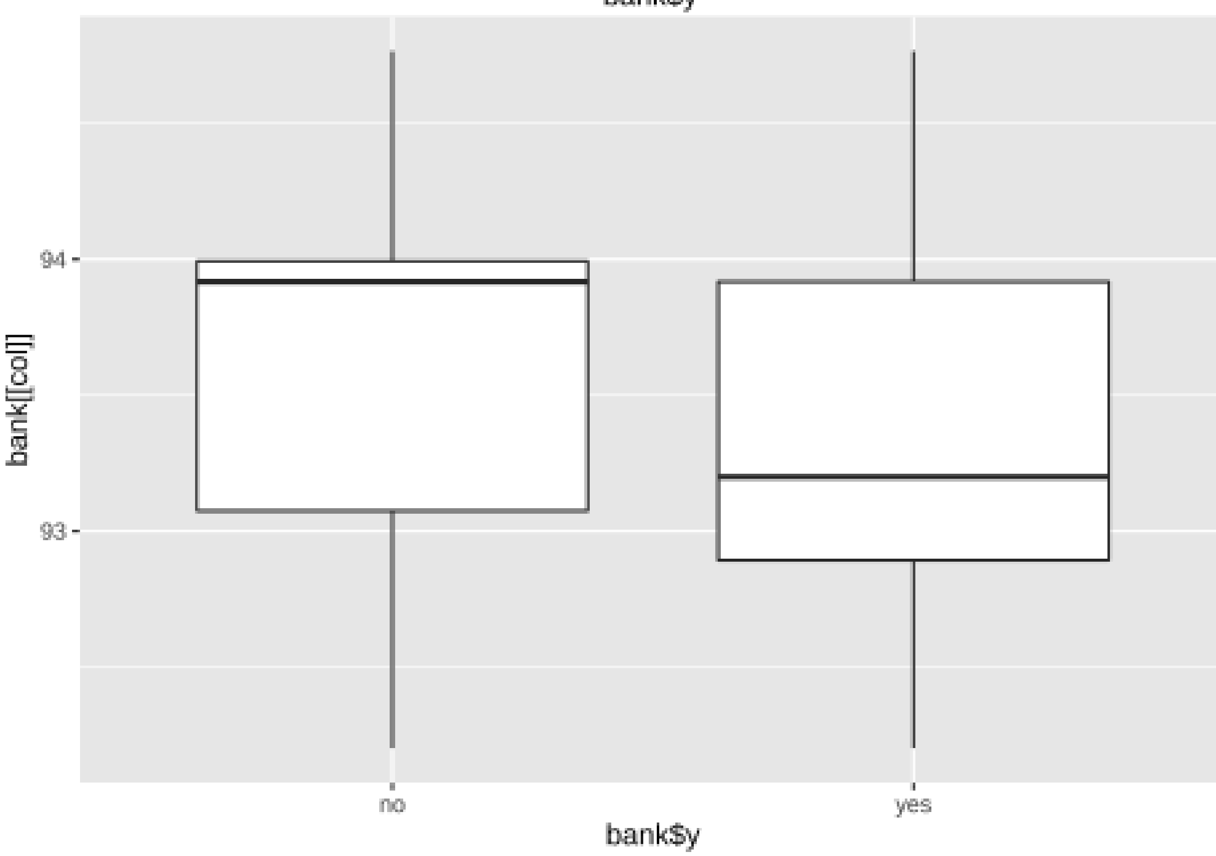


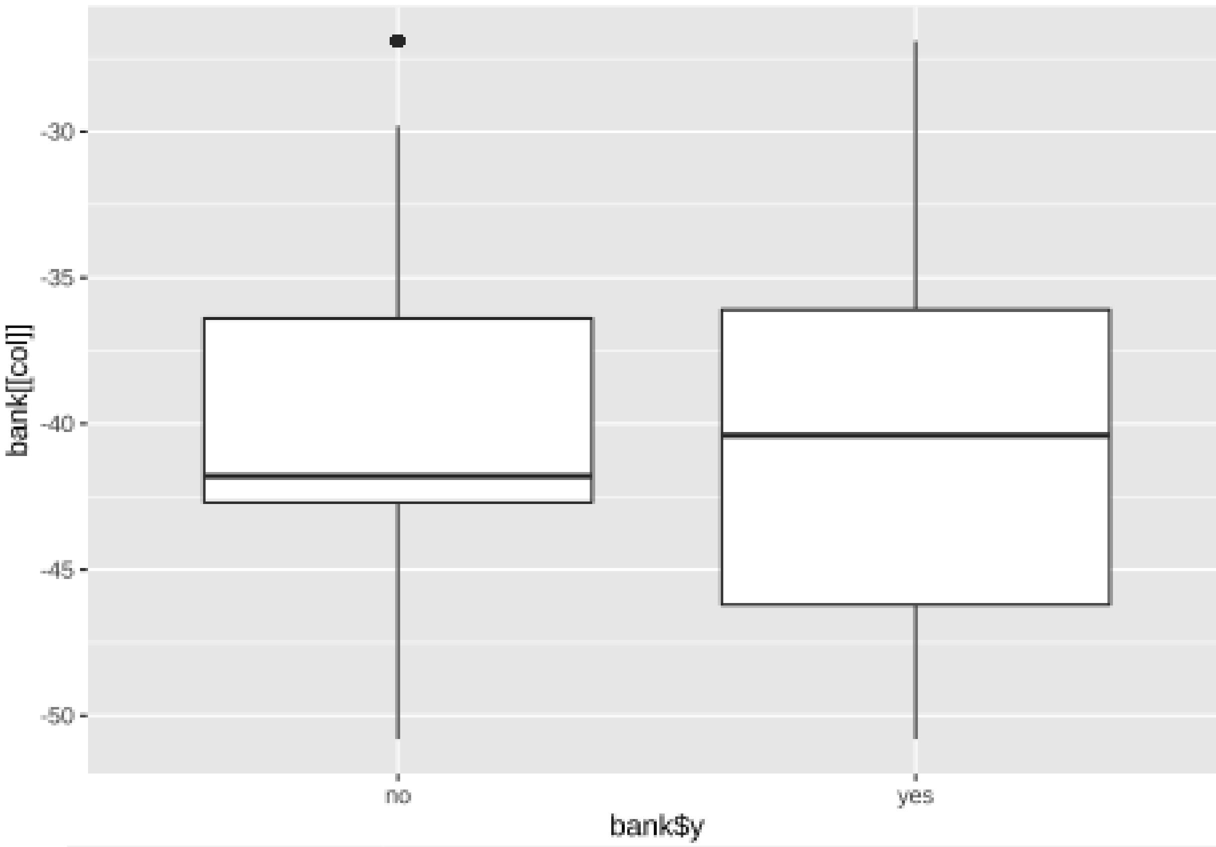
1000
750
500
250
0
bank[[col]]
no
yes
bank$y

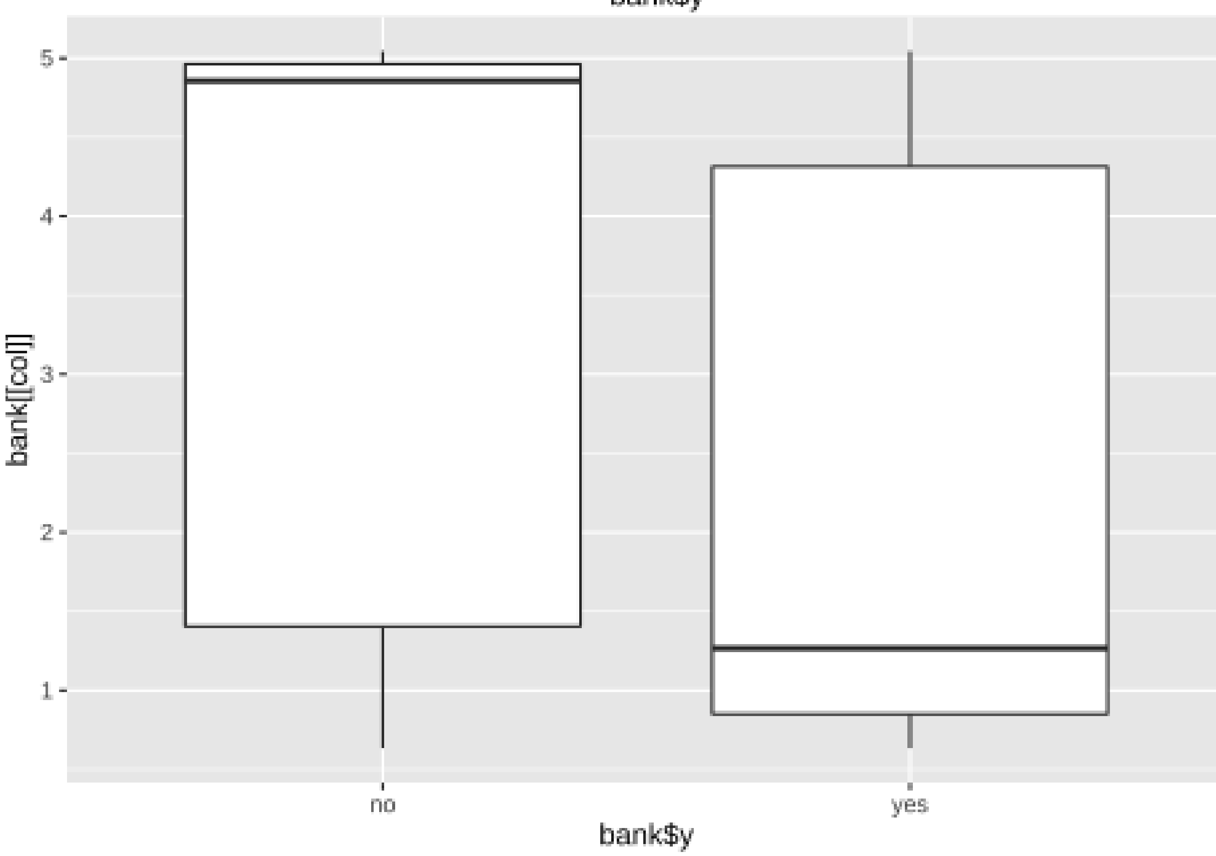



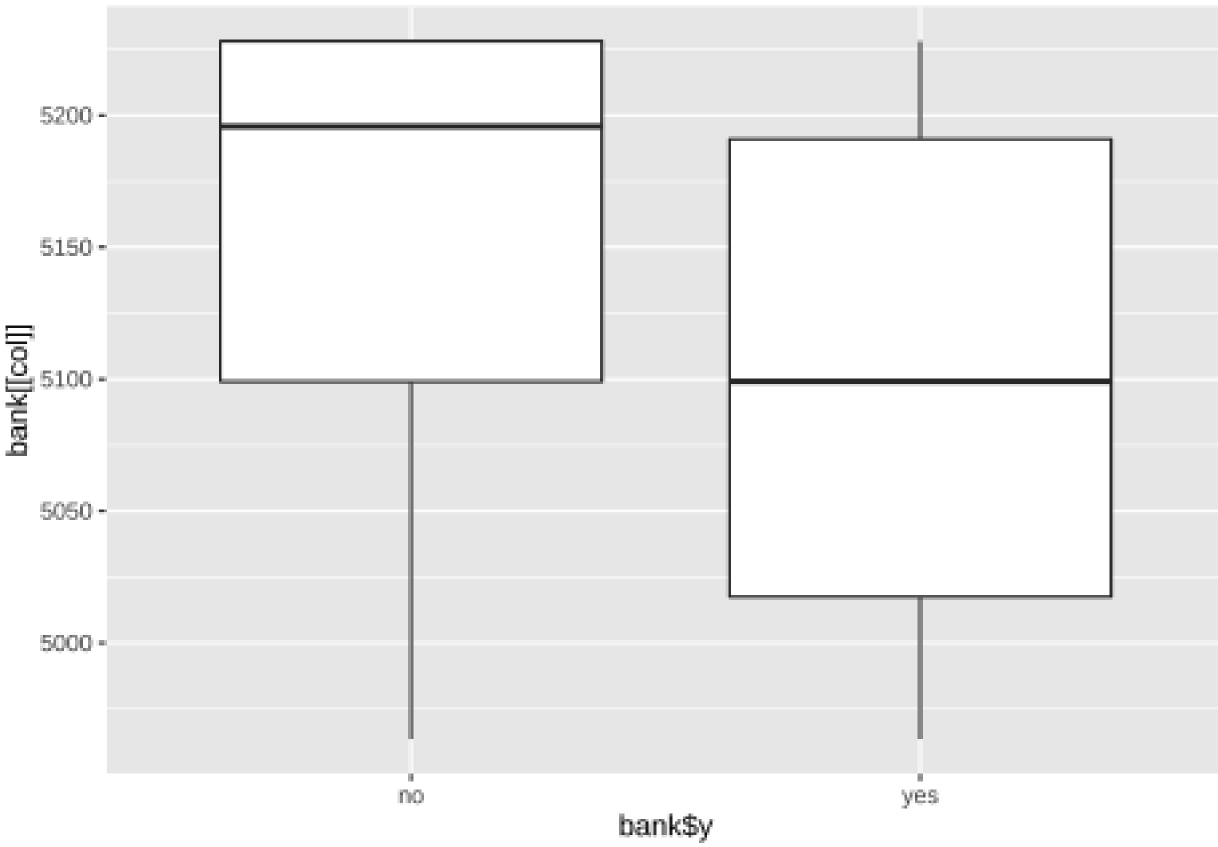
6
4
2
0
bank[[col]]
no
yes
bank$y


1
0
-1
-2
-3
bank[[col]]
no
yes
bank$y


94
93
bank[[col]]
no
yes
bank$y


-30
-35
-40
-45
-50
bank[[col]]
no
yes
bank$y


5
4
3
2
1
bank[[col]]
no
yes
bank$y




##data transformations

```
summary(bank)
```

```
##       Age                  Job            Marital     
##  Min.   :17   administration:10484   divorced: 4609  
##  1st Qu.:32   blue-collar   : 9336   married :24809  
##  Median :38   technician    : 6743   single  :11554  
##  Mean   :40   services      : 3963                   
##  3rd Qu.:47   management    : 2921                   
##  Max.   :98   retired       : 1722                   
##               (Other)       : 5803                   
##                Education        Default         Housing           Loan      
##  basic.4y           : 4318   no     :32458   no     :18518   no     :33767  
##  basic.6y           : 2286   unknown: 8511   unknown:  987   unknown:  987  
##  basic.9y           : 6489   yes    :    3   yes    :21467   yes    : 6218  
##  high.school        : 9817                                                  
##  professional.course: 5448                                                  
##  university.degree  :12614                                                  
##                                                                             
##       Contact          Month       Last.Contact.Day    Duration     
##  cellular :26017   may    :13696   fri:7792         Min.   :   0.0  
##  telephone:14955   jul    : 7143   mon:8466         1st Qu.: 102.0  
##                    aug    : 6130   thu:8564         Median : 180.0  
##                    jun    : 5284   tue:8051         Mean   : 258.3  
##                    nov    : 4089   wed:8099         3rd Qu.: 319.0  
##                    apr    : 2625                    Max.   :4918.0  
##                    (Other): 2005                                    
```

```
##     Campaign          Pdays       Previous.Contacts        Poutcome
##  Min.   : 1.000   Min.   :  0.0   Min.   :0.000     failure    : 4235
##  1st Qu.: 1.000   1st Qu.:999.0   1st Qu.:0.000     nonexistent:35375
##  Median : 2.000   Median :999.0   Median :0.000     success    : 1362
##  Mean   : 2.565   Mean   :962.6   Mean   :0.173
##  3rd Qu.: 3.000   3rd Qu.:999.0   3rd Qu.:0.000
##  Max.   :56.000   Max.   :999.0   Max.   :7.000
##
##   Emp.var.rate      Cons.price.idx  Cons.conf.idx      Euribor3m
##  Min.   :-3.40000   Min.   :92.20   Min.   :-50.80   Min.   :0.634
##  1st Qu.:-1.80000   1st Qu.:93.08   1st Qu.:-42.70   1st Qu.:1.344
##  Median : 1.10000   Median :93.75   Median :-41.80   Median :4.857
##  Mean   : 0.08055   Mean   :93.58   Mean   :-40.51   Mean   :3.620
##  3rd Qu.: 1.40000   3rd Qu.:93.99   3rd Qu.:-36.40   3rd Qu.:4.961
##  Max.   : 1.40000   Max.   :94.77   Max.   :-26.90   Max.   :5.045
##
##  Employment.number   y
##  Min.   :4964      no :36357
##  1st Qu.:5099      yes: 4615
##  Median :5191
##  Mean   :5167
##  3rd Qu.:5228
##  Max.   :5228
##
```

removing Default column because the ratio of yes to no is unbalanced

```
bank <- subset(bank,select=-c(Default))
```

removing Duration since we can know the output only after the call has been made

```
bank <- subset(bank,select=-c(Duration))
```

```
levels(bank$y) <- c(0,1)
```

##handling categories

```
bindata <- subset(dummy_cols(bank,remove_first_dummy = TRUE), select=-c(Job,Education,Month,Last.Contac
```

```
bindata <- subset(bindata,select=-c(y_1))
```

##rescaling continuous columns

```
str(bindata)
```

```
## 'data.frame':    40972 obs. of  47 variables:
##  $ Age                          : int  56 57 37 40 56 45 59 41 24 25 ...
##  $ Campaign                     : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Pdays                        : int  999 999 999 999 999 999 999 999 999 999 ...
##  $ Previous.Contacts            : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Emp.var.rate                 : num  1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 ...
##  $ Cons.price.idx               : num  94 94 94 94 94 ...
##  $ Cons.conf.idx                : num  -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 .
##  $ Euribor3m                    : num  4.86 4.86 4.86 4.86 4.86 ...
##  $ Employment.number            : num  5191 5191 5191 5191 5191 ...
##  $ y                            : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Job_blue-collar              : int  0 0 0 0 0 0 0 1 0 0 ...
##  $ Job_entrepreneur             : int  0 0 0 0 0 0 0 0 0 0 ...
```

```
##  $ Job_housemaid               : int  1 0 0 0 0 0 0 0 0 0 ...
##  $ Job_management              : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Job_retired                 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Job_self-employed           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Job_services                : int  0 1 1 0 1 1 0 0 0 1 ...
##  $ Job_student                 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Job_technician              : int  0 0 0 0 0 0 0 0 1 0 ...
##  $ Job_unemployed              : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Marital_married             : int  1 1 1 1 1 1 1 1 0 0 ...
##  $ Marital_single              : int  0 0 0 0 0 0 0 0 1 1 ...
##  $ Education_basic.6y          : int  0 0 0 1 0 0 0 0 0 0 ...
##  $ Education_basic.9y          : int  0 0 0 0 0 1 0 1 0 0 ...
##  $ Education_high.school       : int  0 1 1 0 1 0 0 0 0 1 ...
##  $ Education_professional.course: int  0 0 0 0 0 0 1 0 1 0 ...
##  $ Education_university.degree  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Housing_unknown             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Housing_yes                 : int  0 0 1 0 0 0 0 0 1 1 ...
##  $ Loan_unknown                : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Loan_yes                    : int  0 0 0 0 1 0 0 0 0 0 ...
##  $ Contact_telephone           : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Month_aug                   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Month_dec                   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Month_jul                   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Month_jun                   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Month_mar                   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Month_may                   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Month_nov                   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Month_oct                   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Month_sep                   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Last.Contact.Day_mon        : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Last.Contact.Day_thu        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Last.Contact.Day_tue        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Last.Contact.Day_wed        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Poutcome_nonexistent        : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Poutcome_success            : int  0 0 0 0 0 0 0 0 0 0 ...
```

```r
str(bank)
```

```
## 'data.frame':    40972 obs. of  19 variables:
##  $ Age              : int  56 57 37 40 56 45 59 41 24 25 ...
##  $ Job              : Factor w/ 11 levels "administration",..: 4 8 8 1 8 8 1 2 10 8 ...
##  $ Marital          : Factor w/ 3 levels "divorced","married",..: 2 2 2 2 2 2 2 2 3 3 ...
##  $ Education        : Factor w/ 6 levels "basic.4y","basic.6y",..: 1 4 4 2 4 3 5 3 5 4 ...
##  $ Housing          : Factor w/ 3 levels "no","unknown",..: 1 1 3 1 1 1 1 1 3 3 ...
##  $ Loan             : Factor w/ 3 levels "no","unknown",..: 1 1 1 1 3 1 1 1 1 1 ...
##  $ Contact          : Factor w/ 2 levels "cellular","telephone": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Month            : Factor w/ 10 levels "apr","aug","dec",..: 7 7 7 7 7 7 7 7 7 7 ...
##  $ Last.Contact.Day : Factor w/ 5 levels "fri","mon","thu",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ Campaign         : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Pdays            : int  999 999 999 999 999 999 999 999 999 999 ...
##  $ Previous.Contacts: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Poutcome         : Factor w/ 3 levels "failure","nonexistent",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ Emp.var.rate     : num  1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 ...
##  $ Cons.price.idx   : num  94 94 94 94 94 ...
##  $ Cons.conf.idx    : num  -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 ...
```

```
##  $ Euribor3m        : num  4.86 4.86 4.86 4.86 4.86 ...
##  $ Employment.number: num  5191 5191 5191 5191 5191 ...
##  $ y                : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
```

minmax scaling

```
minmax <- function(x) {
    return((x- min(x)) /(max(x)-min(x)))
}
```

```
bank <- bank%>%mutate_if(is.numeric,minmax)
```

```
bindata <- bindata%>%mutate_if(is.numeric,minmax)
```

## Split the data into training and test set

```
set.seed(115)
trainIndices = sample(1:dim(bank)[1],round(.8 * dim(bank)[1]))
```

## Build bank test/train

```
bank.train = bank[trainIndices,]
bank.test = bank[-trainIndices,]
```

```
print(table(bank.train$y)/nrow(bank.train))
```

```
##
##         0         1
## 0.8862957 0.1137043
```

```
print(table(bank.test$y)/nrow(bank.test))
```

```
##
##        0        1
## 0.891628 0.108372
```

```
bank.test.class <- bank.test$y
bank.test <- subset(bank.test,select=-c(y))
```

## Build bindata test/train

```
bin.train = bindata[trainIndices,]
bin.test = bindata[-trainIndices,]
```

```
print(table(bin.train$y)/nrow(bin.train))
```

```
##
##         0         1
## 0.8862957 0.1137043
```

```
print(table(bin.test$y)/nrow(bin.test))
```

```
##
##        0        1
## 0.891628 0.108372
```

```r
bin.test.class <- bin.test$y
bin.test <- subset(bin.test,select=-c(y))
```

#modelling

```r
library(e1071)
```

```
##
## Attaching package: 'e1071'

## The following objects are masked from 'package:dlookr':
##
##     kurtosis, skewness
```

```r
library(caTools)
library(class)
library(caret)
library(ROSE)
```

```
## Loaded ROSE 0.0-4
```

```r
library(ROCR)
```

##Logistic Regression

Starting with the full model

```r
glm.out <- glm(y~.,data=bank.train, family = "binomial")
```

```r
summary(glm.out)
```

```
##
## Call:
## glm(formula = y ~ ., family = "binomial", data = bank.train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1110  -0.3908  -0.3196  -0.2599   2.9658
##
## Coefficients: (1 not defined because of singularities)
##                                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)                    -1.2573025  0.5151832  -2.440 0.014667 *
## Age                            -0.1460792  0.1920763  -0.761 0.446940
## Jobblue-collar                 -0.1173002  0.0788588  -1.487 0.136890
## Jobentrepreneur                -0.0043824  0.1180007  -0.037 0.970374
## Jobhousemaid                   -0.0938806  0.1482344  -0.633 0.526522
## Jobmanagement                  -0.0253760  0.0834163  -0.304 0.760969
## Jobretired                      0.3401969  0.1060906   3.207 0.001343 **
## Jobself-employed               -0.0104956  0.1127120  -0.093 0.925810
## Jobservices                    -0.1042535  0.0849642  -1.227 0.219812
## Jobstudent                      0.2848847  0.1095732   2.600 0.009324 **
## Jobtechnician                   0.0002343  0.0712199   0.003 0.997375
## Jobunemployed                   0.0103556  0.1236843   0.084 0.933274
## Maritalmarried                  0.0329193  0.0674172   0.488 0.625343
## Maritalsingle                   0.0290947  0.0771153   0.377 0.705959
## Educationbasic.6y               0.1950516  0.1131341   1.724 0.084694 .
## Educationbasic.9y              -0.0366701  0.0902569  -0.406 0.684533
## Educationhigh.school            0.0460973  0.0888163   0.519 0.603748
```

```
## Educationprofessional.course  0.0956247  0.0980003   0.976 0.329184
## Educationuniversity.degree    0.1700879  0.0888256   1.915 0.055511 .
## Housingunknown               -0.0977340  0.1334515  -0.732 0.463951
## Housingyes                   -0.0213076  0.0404615  -0.527 0.598461
## Loanunknown                          NA         NA      NA       NA
## Loanyes                      -0.0773423  0.0565385  -1.368 0.171325
## Contacttelephone             -0.7887348  0.0758115 -10.404  < 2e-16 ***
## Monthaug                      0.3743492  0.1206825   3.102 0.001923 **
## Monthdec                      0.5381976  0.2118717   2.540 0.011079 *
## Monthjul                      0.0489800  0.0934453   0.524 0.600169
## Monthjun                     -0.6306972  0.1235239  -5.106 3.29e-07 ***
## Monthmar                      1.4968538  0.1463015  10.231  < 2e-16 ***
## Monthmay                     -0.4397211  0.0808314  -5.440 5.33e-08 ***
## Monthnov                     -0.4596285  0.1175851  -3.909 9.27e-05 ***
## Monthoct                     -0.0552917  0.1514693  -0.365 0.715085
## Monthsep                      0.1331698  0.1777168   0.749 0.453654
## Last.Contact.Daymon          -0.2422047  0.0644226  -3.760 0.000170 ***
## Last.Contact.Daythu           0.0287649  0.0623119   0.462 0.644349
## Last.Contact.Daytue           0.0337580  0.0640165   0.527 0.597962
## Last.Contact.Daywed           0.1205612  0.0637077   1.892 0.058436 .
## Campaign                     -2.0791475  0.5546419  -3.749 0.000178 ***
## Pdays                        -1.2254246  0.2262510  -5.416 6.09e-08 ***
## Previous.Contacts            -0.4122987  0.4357934  -0.946 0.344104
## Poutcomenonexistent           0.4729014  0.0966416   4.893 9.91e-07 ***
## Poutcomesuccess               0.6813410  0.2218922   3.071 0.002136 **
## Emp.var.rate                 -7.0228597  0.6689174 -10.499  < 2e-16 ***
## Cons.price.idx                5.3380864  0.6315784   8.452  < 2e-16 ***
## Cons.conf.idx                 0.8801959  0.1872645   4.700 2.60e-06 ***
## Euribor3m                     0.6511612  0.5666569   1.149 0.250503
## Employment.number             1.8949782  0.7984677   2.373 0.017631 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 23219  on 32777  degrees of freedom
## Residual deviance: 18080  on 32732  degrees of freedom
## AIC: 18172
##
## Number of Fisher Scoring iterations: 6
```

**Insights:** - Na's above is due to two of the independent variables being perfectly collinear( Housing_unknown and loan_unknown) - Realized Housing and Personal Loan are collinear reason being client chooses to not disclose both together - The AIC for full model is 18172 and there is a significant difference between Null and Residual deviance - At first look, Job, Contact type, Month, Campaign, Pdays, Poutcome, Emp.var.rate,Cons.price.idx and cons.conf.idx are have significant influence on output - coefficient estimates of monthmarch, Campaign, Pdays, Emp.var.rate, Cons.price.index has the most influence with unit change in them

```r
stepmodel <- stepAIC(glm.out,direction = "backward",trace = FALSE)
stepmodel
```

```
##
## Call:  glm(formula = y ~ Job + Education + Contact + Month + Last.Contact.Day +
##     Campaign + Pdays + Poutcome + Emp.var.rate + Cons.price.idx +
```

```
##     Cons.conf.idx + Employment.number, family = "binomial", data = bank.train)
##
## Coefficients:
##                   (Intercept)                 Jobblue-collar
##                     -1.837049                      -0.114682
##               Jobentrepreneur                   Jobhousemaid
##                     -0.010348                      -0.104402
##                 Jobmanagement                     Jobretired
##                     -0.031733                       0.295164
##              Jobself-employed                    Jobservices
##                     -0.007536                      -0.104016
##                    Jobstudent                  Jobtechnician
##                      0.305759                       0.004669
##                 Jobunemployed               Educationbasic.6y
##                      0.012630                       0.205582
##             Educationbasic.9y           Educationhigh.school
##                     -0.024806                       0.060860
## Educationprofessional.course    Educationuniversity.degree
##                      0.107723                       0.186258
##              Contacttelephone                       Monthaug
##                     -0.782207                       0.392166
##                      Monthdec                       Monthjul
##                      0.596197                       0.069404
##                      Monthjun                       Monthmar
##                     -0.639189                       1.544914
##                      Monthmay                       Monthnov
##                     -0.420445                      -0.380506
##                      Monthoct                       Monthsep
##                      0.045692                       0.223956
##          Last.Contact.Daymon            Last.Contact.Daythu
##                     -0.240146                       0.027603
##          Last.Contact.Daytue            Last.Contact.Daywed
##                      0.037400                       0.122790
##                      Campaign                          Pdays
##                     -2.127050                      -1.151519
##          Poutcomenonexistent                Poutcomesuccess
##                      0.550012                       0.739335
##                  Emp.var.rate                 Cons.price.idx
##                     -6.932523                       5.604686
##                 Cons.conf.idx              Employment.number
##                      1.023976                       2.575396
##
## Degrees of Freedom: 32777 Total (i.e. Null);  32740 Residual
## Null Deviance:       23220
## Residual Deviance: 18090     AIC: 18160
```

#run this part at last # `{r} # library(bootStepAIC) #` # # `{r} # bootmod <- boot.stepAIC(glm.out,bank.train,B=`
`# bootmod #`

**Insights** - We started with a full model having an AIC of 18172 then the Marital is found to be insignificant then we removed marital and this process is repeated and removed Housing, Loan, Age, Previous.Contacts and Euribor3m reducing the AIC to 18168, 18166, 18165, 18163, 18162 respectively - Campaign,Cons.conf.idx,Cons.price.idx,Contacttelephone,Emp.var.rate,Monthmar,Monthmay,Pdays Poutcomenonexistent were selected 100% of the times.

Fitting best model got from bootstrap stepwise AIC

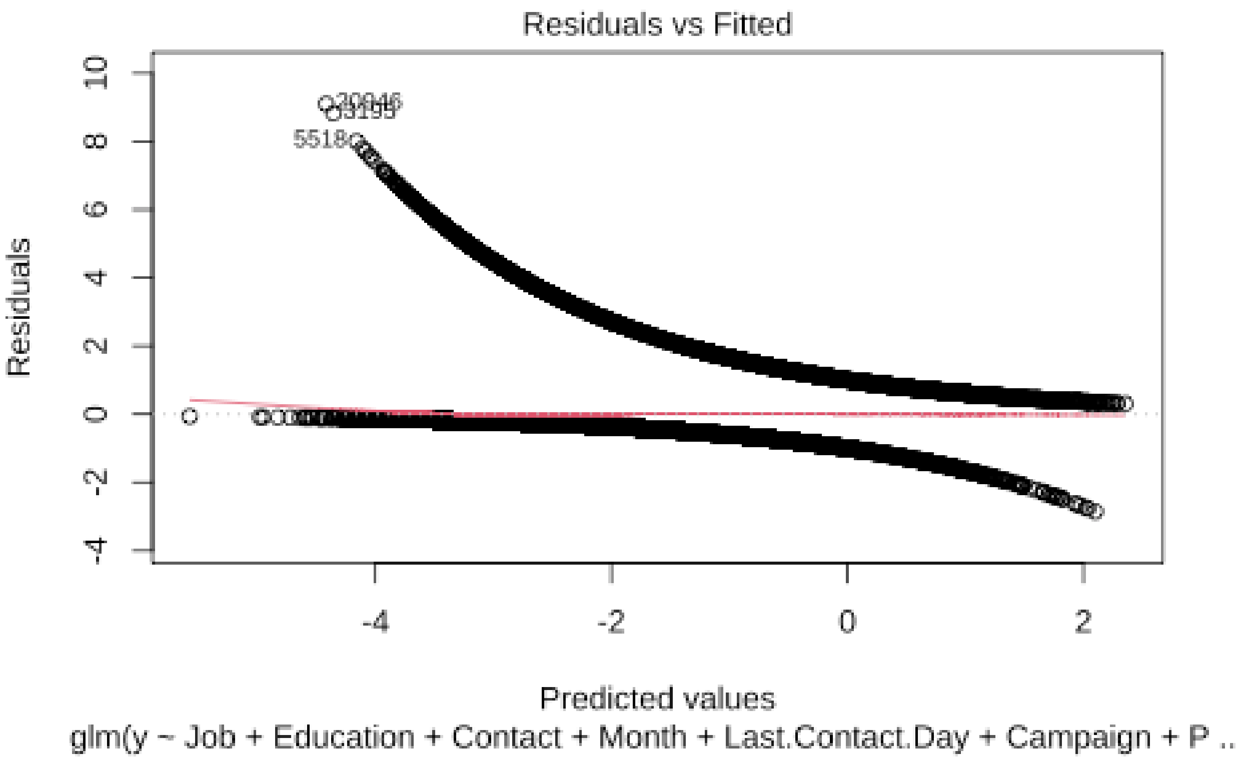
```
best.model <- glm(y ~ Job + Education + Contact + Month + Last.Contact.Day + Campaign +
    Pdays + Poutcome + Emp.var.rate + Cons.price.idx + Cons.conf.idx +
    Employment.number,data=bank.train,family = "binomial")
```

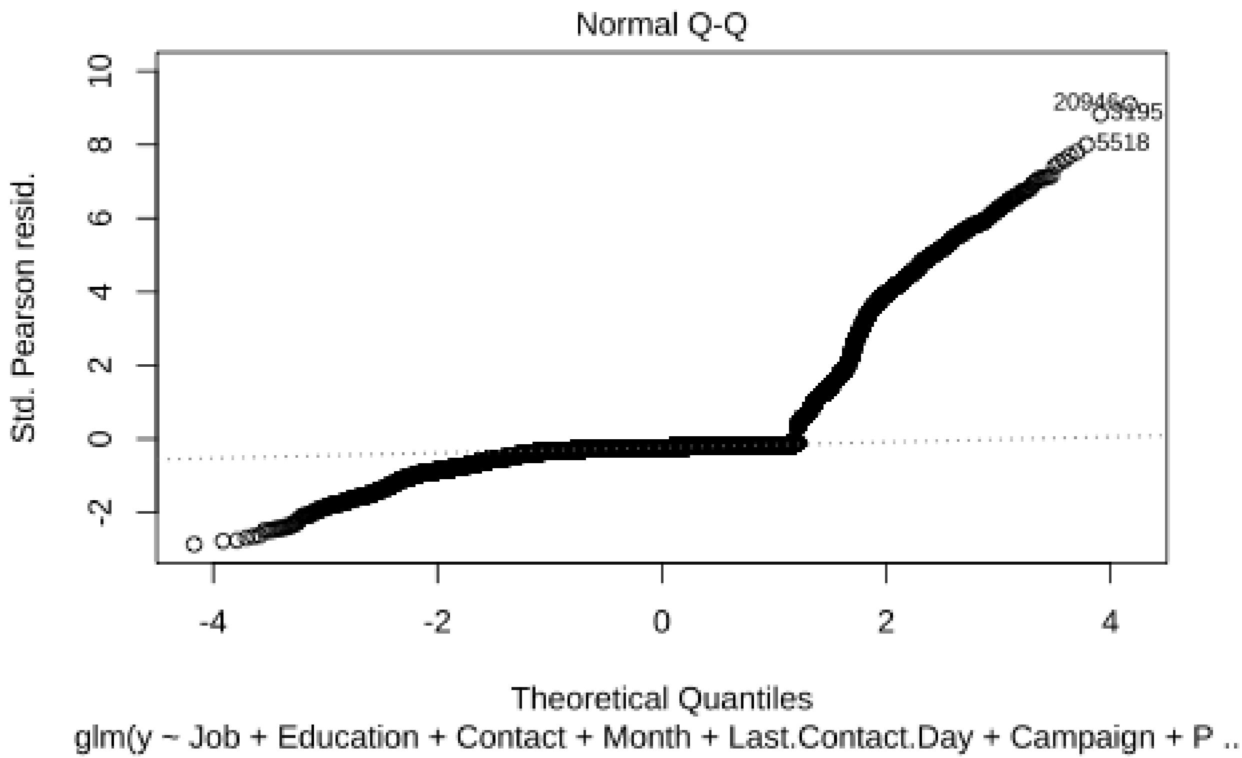
```
summary(best.model)
```

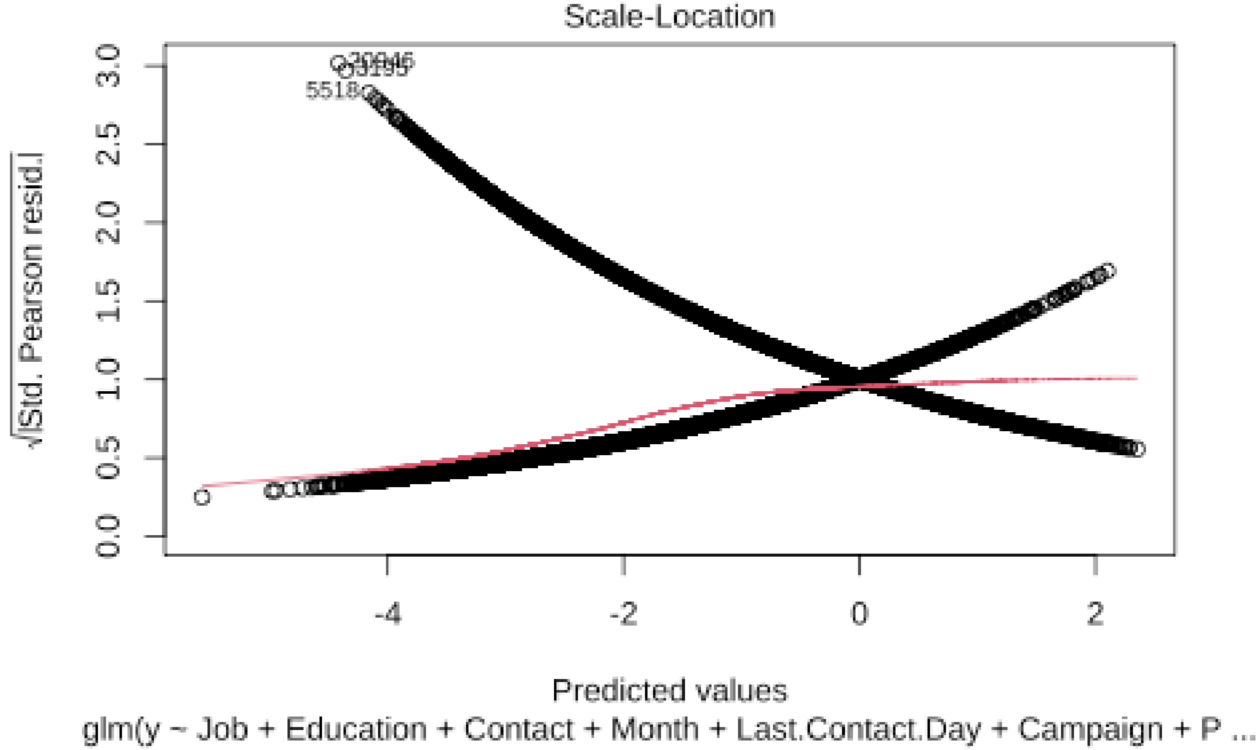
```
##
## Call:
## glm(formula = y ~ Job + Education + Contact + Month + Last.Contact.Day +
##     Campaign + Pdays + Poutcome + Emp.var.rate + Cons.price.idx +
##     Cons.conf.idx + Employment.number, family = "binomial", data = bank.train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1054  -0.3905  -0.3209  -0.2565   2.9762
##
## Coefficients:
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                   -1.837049   0.345994  -5.309 1.10e-07 ***
## Jobblue-collar                -0.114682   0.078537  -1.460 0.144226
## Jobentrepreneur               -0.010348   0.117389  -0.088 0.929755
## Jobhousemaid                  -0.104402   0.147846  -0.706 0.480092
## Jobmanagement                 -0.031733   0.082455  -0.385 0.700351
## Jobretired                     0.295164   0.092565   3.189 0.001429 **
## Jobself-employed              -0.007536   0.112572  -0.067 0.946625
## Jobservices                   -0.104016   0.084854  -1.226 0.220267
## Jobstudent                     0.305759   0.104105   2.937 0.003314 **
## Jobtechnician                  0.004669   0.071139   0.066 0.947672
## Jobunemployed                  0.012630   0.123535   0.102 0.918567
## Educationbasic.6y              0.205582   0.112640   1.825 0.067981 .
## Educationbasic.9y             -0.024806   0.089489  -0.277 0.781630
## Educationhigh.school           0.060860   0.087392   0.696 0.486178
## Educationprofessional.course  0.107723   0.097186   1.108 0.267680
## Educationuniversity.degree     0.186258   0.087092   2.139 0.032466 *
## Contacttelephone              -0.782207   0.075558 -10.352  < 2e-16 ***
## Monthaug                       0.392166   0.119120   3.292 0.000994 ***
## Monthdec                       0.596197   0.203256   2.933 0.003355 **
## Monthjul                       0.069404   0.092196   0.753 0.451576
## Monthjun                      -0.639189   0.122665  -5.211 1.88e-07 ***
## Monthmar                       1.544914   0.139630  11.064  < 2e-16 ***
## Monthmay                      -0.420445   0.079096  -5.316 1.06e-07 ***
## Monthnov                      -0.380506   0.093797  -4.057 4.98e-05 ***
## Monthoct                       0.045692   0.126066   0.362 0.717017
## Monthsep                       0.223956   0.158314   1.415 0.157176
## Last.Contact.Daymon           -0.240146   0.064322  -3.734 0.000189 ***
## Last.Contact.Daythu            0.027603   0.062258   0.443 0.657505
## Last.Contact.Daytue            0.037400   0.063848   0.586 0.558032
## Last.Contact.Daywed            0.122790   0.063632   1.930 0.053646 .
## Campaign                      -2.127050   0.554536  -3.836 0.000125 ***
## Pdays                         -1.151519   0.211903  -5.434 5.51e-08 ***
## Poutcomenonexistent            0.550012   0.063358   8.681  < 2e-16 ***
## Poutcomesuccess                0.739335   0.213574   3.462 0.000537 ***
## Emp.var.rate                  -6.932523   0.666561 -10.400  < 2e-16 ***
```

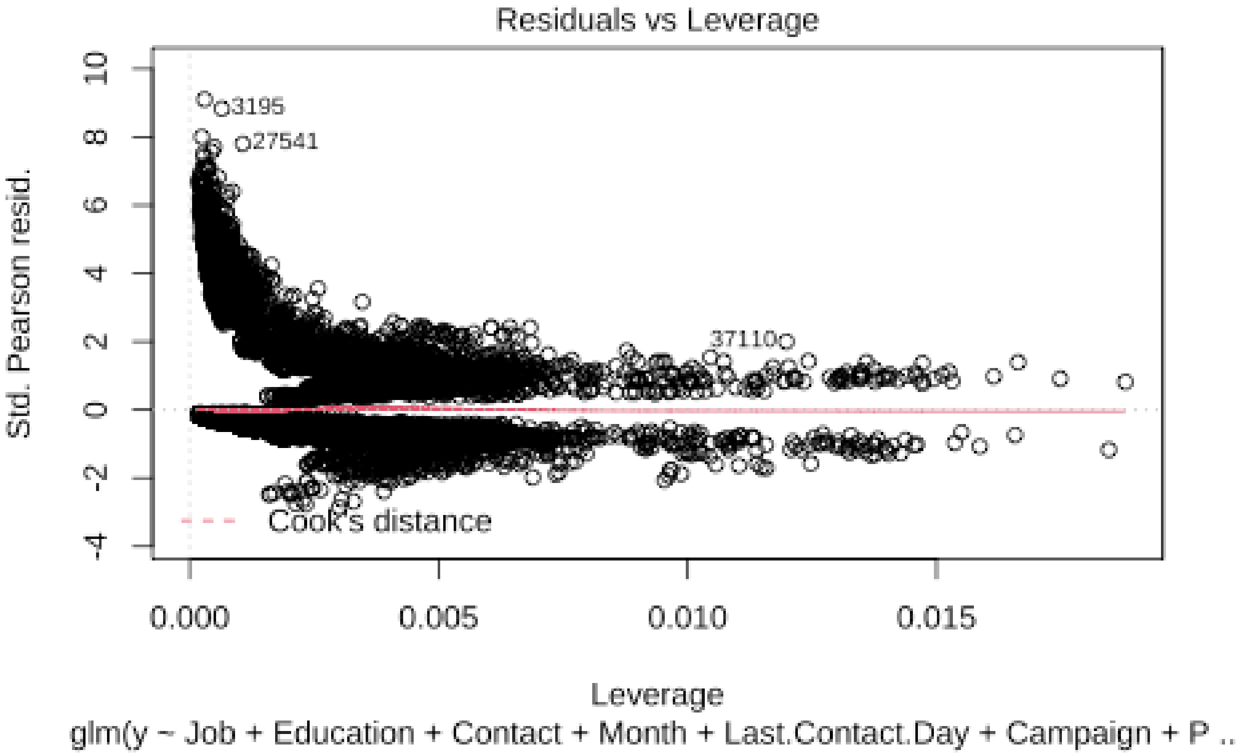
```
## Cons.price.idx               5.604686   0.569632   9.839  < 2e-16 ***
## Cons.conf.idx                1.023976   0.130896   7.823 5.17e-15 ***
## Employment.number            2.575396   0.532256   4.839 1.31e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 23219  on 32777  degrees of freedom
## Residual deviance: 18086  on 32740  degrees of freedom
## AIC: 18162
##
## Number of Fisher Scoring iterations: 6
```

```
plot(best.model)
```




Normal Q-Q
Std. Pearson resid.
Theoretical Quantiles
glm(y ~ Job + Education + Contact + Month + Last.Contact.Day + Campaign + P ...
20946
3195
5518


Scale-Location
√|Std. Pearson resid.|
Predicted values
glm(y ~ Job + Education + Contact + Month + Last.Contact.Day + Campaign + P ...
20946
3195
5518




1)These plots have been made for linear models, they help us to identify some irregularities in the data, but they don't have to affect the model since they are not have been designed for logistic regression. 2) In the first plot lower line is showing the negative residuals when we are predicting the label as 0 and the superior line of points is when we have positive residuals when predicting 1. 3)The second plot helps us to find out if we are using the right distribution and to detect skewness in our data, we can observe that is skewed and doesn't fit adequately to the dashed line which would be the ideal scenario. 4)This third plot helps us to identify homoscedasticity in the residuals from this spread we can infer that the residuals are spread wider and then decrease. 5)The fouth graph shows the Cooks distance to identify the influence that have the outliers, overall we can observe that they don't have a big effect because all the points are spread along the red dashed line.

interpreting odds ratio

```
oddsratio <- data.frame(exp(best.model$coefficients))
oddsratio
```

```
##                                 exp.best.model.coefficients.
## (Intercept)                                     1.592868e-01
## Jobblue-collar                                  8.916498e-01
## Jobentrepreneur                                 9.897052e-01
## Jobhousemaid                                    9.008628e-01
## Jobmanagement                                   9.687657e-01
## Jobretired                                      1.343347e+00
## Jobself-employed                                9.924921e-01
## Jobservices                                     9.012110e-01
## Jobstudent                                      1.357655e+00
## Jobtechnician                                   1.004680e+00
## Jobunemployed                                   1.012710e+00
## Educationbasic.6y                               1.228240e+00
## Educationbasic.9y                               9.754992e-01
## Educationhigh.school                            1.062750e+00
## Educationprofessional.course                    1.113739e+00
```

```
## Educationuniversity.degree                         1.204732e+00
## Contacttelephone                                   4.573952e-01
## Monthaug                                           1.480184e+00
## Monthdec                                           1.815202e+00
## Monthjul                                           1.071870e+00
## Monthjun                                           5.277200e-01
## Monthmar                                           4.687567e+00
## Monthmay                                           6.567545e-01
## Monthnov                                           6.835156e-01
## Monthoct                                           1.046752e+00
## Monthsep                                           1.251016e+00
## Last.Contact.Daymon                                7.865130e-01
## Last.Contact.Daythu                                1.027987e+00
## Last.Contact.Daytue                                1.038108e+00
## Last.Contact.Daywed                                1.130647e+00
## Campaign                                           1.191884e-01
## Pdays                                              3.161560e-01
## Poutcomenonexistent                                1.733273e+00
## Poutcomesuccess                                    2.094543e+00
## Emp.var.rate                                       9.755366e-04
## Cons.price.idx                                     2.716966e+02
## Cons.conf.idx                                      2.784243e+00
## Employment.number                                  1.313652e+01
```

seeing for any VIF

```
car::vif(best.model)
```

```
##                         GVIF Df GVIF^(1/(2*Df))
## Job                 3.771678 10        1.068628
## Education           3.359351  5        1.128822
## Contact             2.425169  1        1.557295
## Month              27.190240  9        1.201405
## Last.Contact.Day    1.047319  4        1.005796
## Campaign            1.042488  1        1.021023
## Pdays               9.448266  1        3.073803
## Poutcome           10.594971  2        1.804160
## Emp.var.rate      144.033959  1       12.001415
## Cons.price.idx     53.853756  1        7.338512
## Cons.conf.idx       2.602511  1        1.613230
## Employment.number  75.155914  1        8.669251
```

```
best.model1 <- glm(y ~ Job + Education + Contact + Month + Last.Contact.Day + Campaign +
    Pdays + Poutcome + Cons.price.idx + Cons.conf.idx +
    Employment.number,data=bank.train,family = "binomial")
```

```
car::vif(best.model1)
```

```
##                        GVIF Df GVIF^(1/(2*Df))
## Job                3.767926 10        1.068575
## Education          3.367212  5        1.129086
## Contact            1.900338  1        1.378527
## Month              5.112846  9        1.094889
## Last.Contact.Day   1.044074  4        1.005406
## Campaign           1.040604  1        1.020100
## Pdays              9.436236  1        3.071846
```

```
## Poutcome          10.562847  2        1.802790
## Cons.price.idx     1.882988  1        1.372220
## Cons.conf.idx      2.293264  1        1.514353
## Employment.number  2.098835  1        1.448736
```

```
summary(best.model1)
```

```
##
## Call:
## glm(formula = y ~ Job + Education + Contact + Month + Last.Contact.Day +
##     Campaign + Pdays + Poutcome + Cons.price.idx + Cons.conf.idx +
##     Employment.number, family = "binomial", data = bank.train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1921  -0.3938  -0.3250  -0.2551   2.9968
##
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  0.7383418  0.2408773   3.065 0.002175 **
## Jobblue-collar              -0.1256084  0.0785121  -1.600 0.109630
## Jobentrepreneur              0.0015065  0.1171113   0.013 0.989736
## Jobhousemaid                -0.1107733  0.1472191  -0.752 0.451788
## Jobmanagement               -0.0302277  0.0822008  -0.368 0.713075
## Jobretired                   0.3255445  0.0920386   3.537 0.000405 ***
## Jobself-employed            -0.0003107  0.1120773  -0.003 0.997788
## Jobservices                 -0.1096263  0.0846959  -1.294 0.195544
## Jobstudent                   0.3295450  0.1037242   3.177 0.001487 **
## Jobtechnician               -0.0200663  0.0707188  -0.284 0.776603
## Jobunemployed                0.0031783  0.1235209   0.026 0.979472
## Educationbasic.6y            0.2086569  0.1127230   1.851 0.064161 .
## Educationbasic.9y           -0.0149365  0.0893547  -0.167 0.867245
## Educationhigh.school         0.0593791  0.0873653   0.680 0.496717
## Educationprofessional.course  0.1016416  0.0970727   1.047 0.295069
## Educationuniversity.degree    0.1790453  0.0870118   2.058 0.039618 *
## Contacttelephone            -0.5210061  0.0674987  -7.719 1.17e-14 ***
## Monthaug                    -0.3032106  0.0992104  -3.056 0.002241 **
## Monthdec                     0.1787919  0.1978971   0.903 0.366282
## Monthjul                     0.1382763  0.0909884   1.520 0.128583
## Monthjun                     0.2420050  0.0890664   2.717 0.006585 **
## Monthmar                     0.7900992  0.1215627   6.500 8.06e-11 ***
## Monthmay                    -0.7359764  0.0725790 -10.140  < 2e-16 ***
## Monthnov                    -0.4400200  0.0934616  -4.708 2.50e-06 ***
## Monthoct                    -0.3568712  0.1237028  -2.885 0.003915 **
## Monthsep                    -0.6641458  0.1351169  -4.915 8.86e-07 ***
## Last.Contact.Daymon         -0.2535979  0.0640991  -3.956 7.61e-05 ***
## Last.Contact.Daythu          0.0107951  0.0619277   0.174 0.861615
## Last.Contact.Daytue          0.0143894  0.0636035   0.226 0.821018
## Last.Contact.Daywed          0.0950994  0.0633458   1.501 0.133285
## Campaign                    -2.2939666  0.5583876  -4.108 3.99e-05 ***
## Pdays                       -1.2007223  0.2109326  -5.692 1.25e-08 ***
## Poutcomenonexistent          0.5456340  0.0631671   8.638  < 2e-16 ***
## Poutcomesuccess              0.7015141  0.2124949   3.301 0.000962 ***
## Cons.price.idx              -0.1896196  0.1079512  -1.757 0.078998 .
## Cons.conf.idx                0.5909928  0.1224579   4.826 1.39e-06 ***
```

```
## Employment.number            -2.8865492  0.0883866 -32.658  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 23219  on 32777  degrees of freedom
## Residual deviance: 18194  on 32741  degrees of freedom
## AIC: 18268
##
## Number of Fisher Scoring iterations: 6
```

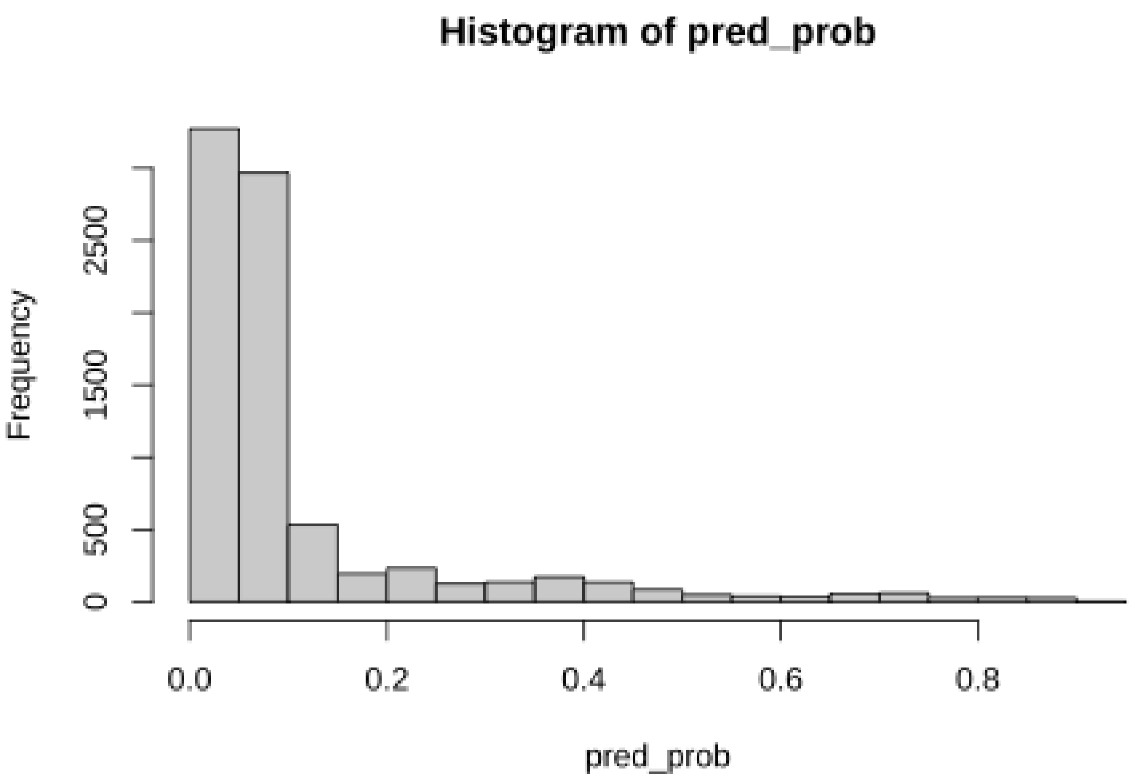
**Takeaway:** - It is not always preferred to take out multicollinearity induced in the models

confusion matrix for best model

```
pred_prob <- predict(best.model, bank.test ,type="response")
```

histogram of prediction probability

```
hist(pred_prob)
```



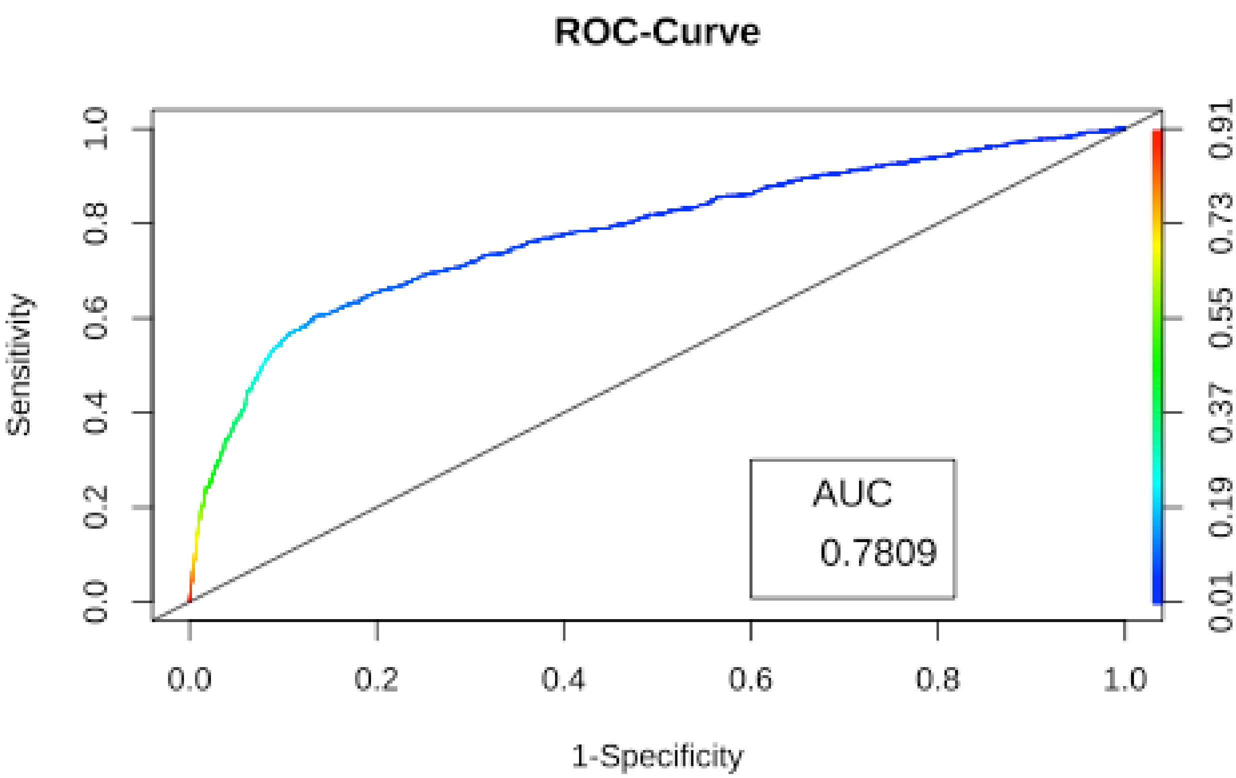

```
confusionMatrix(table(Predicted=ifelse(pred_prob>0.5,1,0),Actual=bank.test.class))
```

```
## Confusion Matrix and Statistics
##
##          Actual
## Predicted    0    1
##         0 7181  678
##         1  125  210
##
##                Accuracy : 0.902
```

```
##                  95% CI : (0.8954, 0.9084)
##     No Information Rate : 0.8916
##     P-Value [Acc > NIR] : 0.001175
##
##                   Kappa : 0.302
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9829
##             Specificity : 0.2365
##          Pos Pred Value : 0.9137
##          Neg Pred Value : 0.6269
##              Prevalence : 0.8916
##          Detection Rate : 0.8764
##    Detection Prevalence : 0.9591
##       Balanced Accuracy : 0.6097
##
##        'Positive' Class : 0
##
```

#AUC-ROC-curve

```
pred <- prediction(pred_prob,bin.test.class)
perf <- performance(pred,"tpr","fpr")
plot(perf,colorize=TRUE,main="ROC-Curve",xlab="1-Specificity",ylab="Sensitivity")
abline(a=0,b=1)
auc <- performance(pred,"auc")
auc <- unlist(slot(auc,"y.values"))
auc <- round(auc,4)
legend(.6,.3,auc,title="AUC",cex = 1.2)
```



## understanding Logistic Regression

- Median of Deviance Residual is Low meaning that the model is not baised. Thus the model is not over or under estimating the output
- Null deviance: A low null deviance implies that the data can be modeled well merely using the intercept. If the null deviance is low, you should consider using few features for modeling the data.
- Residual deviance: A low residual deviance implies that the model you have trained is appropriate.
- These results are somehow reassuring. First, the null deviance is high, which means it makes sense to use more than a single parameter for fitting the model. Second, the residual deviance is relatively low, which indicates that the log likelihood of our model is close to the log likelihood of the saturated model. However, for a well-fitting model, the residual deviance should be close to the degrees of freedom (74), which is not the case here. For example, this could be a result of overdispersion (underdispersion in our case because residual deviance is much lower than )where the variation is greater than predicted by the model. This can happen for a Poisson model when the actual variance exceeds the assumed mean of = ( ).

##LDA The main purpose of LDA is to find the linear combination of the different variables that persuade a customer to get a bank term deposit and we have two different groups so we can find only one useful discriminant function.

Renaming blue-collar and self-employed in both binary train and test sets

```
colnames(bin.train)[11] <- "Job_bluecollar"
colnames(bin.train)[16] <- "Job_selfemployed"
colnames(bin.test)[10] <- "Job_bluecollar"
colnames(bin.test)[15] <- "Job_selfemployed"
```

Dropping Housing unknown because of raising errors due to collinearity with Loan unknown

```
bin.train <- subset(bin.train, select=-c(Housing_unknown))
bin.test <- subset(bin.test, select=-c(Housing_unknown))
```

```
linear <- lda(y~.,data=bin.train)
linear
```

```
## Call:
## lda(y ~ ., data = bin.train)
##
## Prior probabilities of groups:
##         0         1
## 0.8862957 0.1137043
##
## Group means:
##         Age   Campaign     Pdays Previous.Contacts Emp.var.rate Cons.price.idx
## 0 0.2827080 0.02978086 0.9854601        0.01895681    0.7607132      0.5474033
## 1 0.2941531 0.01940630 0.7916849        0.07117942    0.4470866      0.4470957
##   Cons.conf.idx Euribor3m Employment.number Job_bluecollar Job_entrepreneur
## 0     0.4265200 0.7208477         0.8037802      0.2391312       0.03676293
## 1     0.4610066 0.3335517         0.4953261      0.1400590       0.02790448
##   Job_housemaid Job_management Job_retired Job_selfemployed Job_services
## 0    0.02550687     0.07214898  0.03538605       0.03493856   0.09948022
## 1    0.02092836     0.07298095  0.09605581       0.03353904   0.06895627
##   Job_student Job_technician Job_unemployed Marital_married Marital_single
## 0  0.01655709      0.1658463     0.02419882       0.6118206      0.2741386
## 1  0.06412664      0.1558895     0.03139254       0.5527234      0.3469278
##   Education_basic.6y Education_basic.9y Education_high.school
## 0         0.05776049          0.1650546             0.2400262
```

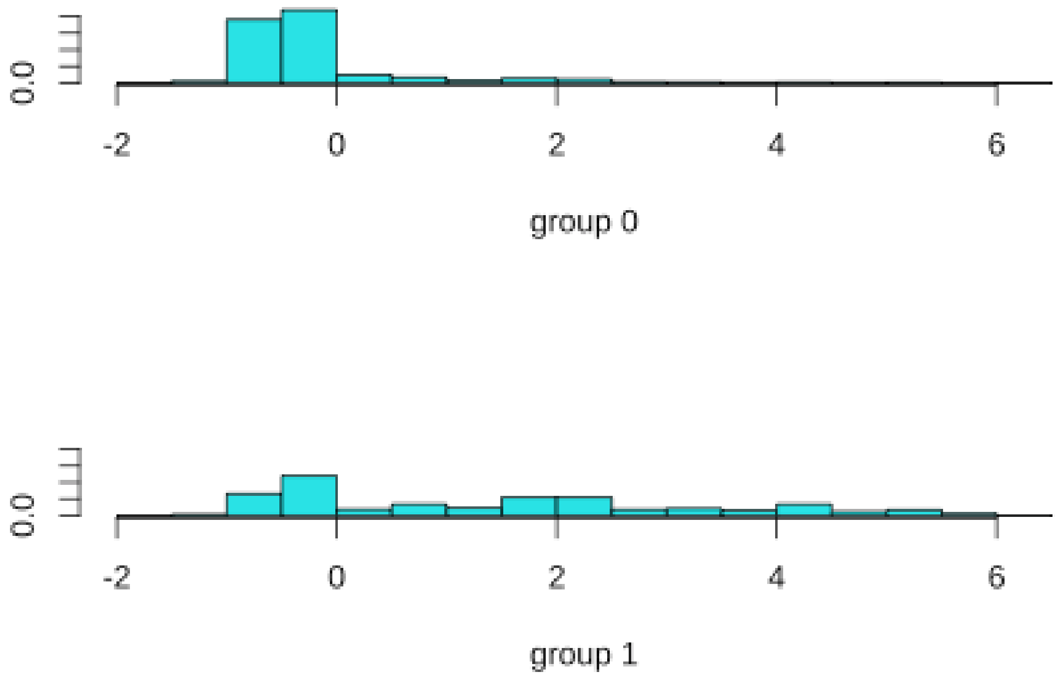
```
## 1        0.04292997          0.1065200            0.2363831
##   Education_professional.course Education_university.degree Housing_yes
## 0                     0.1321813                   0.2982341   0.5227015
## 1                     0.1341562                   0.3799302   0.5446740
##   Loan_unknown  Loan_yes Contact_telephone Month_aug   Month_dec Month_jul
## 0   0.02457747 0.1514922         0.3917593 0.1507349 0.002375133 0.1776187
## 1   0.02280655 0.1435471         0.1671586 0.1408640 0.020123424 0.1389858
##   Month_jun   Month_mar Month_may  Month_nov  Month_oct   Month_sep
## 0 0.1307012 0.007194245 0.3530687 0.10133902 0.01118722 0.008536711
## 1 0.1253019 0.058223772 0.1888919 0.09122619 0.06627314 0.052589214
##   Last.Contact.Day_mon Last.Contact.Day_thu Last.Contact.Day_tue
## 0            0.2100100            0.2072562            0.1955871
## 1            0.1843306            0.2208210            0.2063322
##   Last.Contact.Day_wed Poutcome_nonexistent Poutcome_success
## 0            0.1955182            0.8869574       0.01290833
## 1            0.2020392            0.6748055       0.19318487
##
## Coefficients of linear discriminants:
##                                        LD1
## Age                           -0.049952195
## Campaign                      -0.708352274
## Pdays                         -1.755790133
## Previous.Contacts             -0.559270780
## Emp.var.rate                  -7.170172538
## Cons.price.idx                 5.024595071
## Cons.conf.idx                  1.024463201
## Euribor3m                      2.121984235
## Employment.number              0.174768224
## Job_bluecollar                -0.050605323
## Job_entrepreneur              -0.005795040
## Job_housemaid                 -0.056515140
## Job_management                -0.022121485
## Job_retired                    0.280777579
## Job_selfemployed              -0.001915341
## Job_services                  -0.051502219
## Job_student                    0.317267162
## Job_technician                 0.006051044
## Job_unemployed                -0.004188277
## Marital_married                0.019352108
## Marital_single                 0.023167305
## Education_basic.6y             0.093365019
## Education_basic.9y            -0.024969081
## Education_high.school          0.017167334
## Education_professional.course  0.039832296
## Education_university.degree    0.097599366
## Housing_yes                   -0.008853847
## Loan_unknown                  -0.054819686
## Loan_yes                      -0.041471903
## Contact_telephone             -0.573020653
## Month_aug                      0.585413694
## Month_dec                      0.882042476
## Month_jul                      0.181558663
## Month_jun                     -0.518337693
## Month_mar                      1.997633411
```

```
## Month_may                      -0.347517141
## Month_nov                      -0.334663956
## Month_oct                      -0.128459535
## Month_sep                      -0.003035936
## Last.Contact.Day_mon           -0.158134491
## Last.Contact.Day_thu            0.022844934
## Last.Contact.Day_tue            0.014834289
## Last.Contact.Day_wed            0.064549694
## Poutcome_nonexistent            0.368945911
## Poutcome_success                1.033230018
```

The discriminant function is -0.0499$age-0.7083$Campaign+…+1.0332*Poutcome_success. We can observe by the prior probabilities that 88.62% of the training set belongs to group 0 and only 11.37% belongs to 1.

```
p <- predict(linear, bin.test)
```

```
ldahist(data = p$x, g=bin.test.class)
```




We can notice that both groups are overlapping which is not a good signal, so we can infer that there is not a proper separation between the groups.

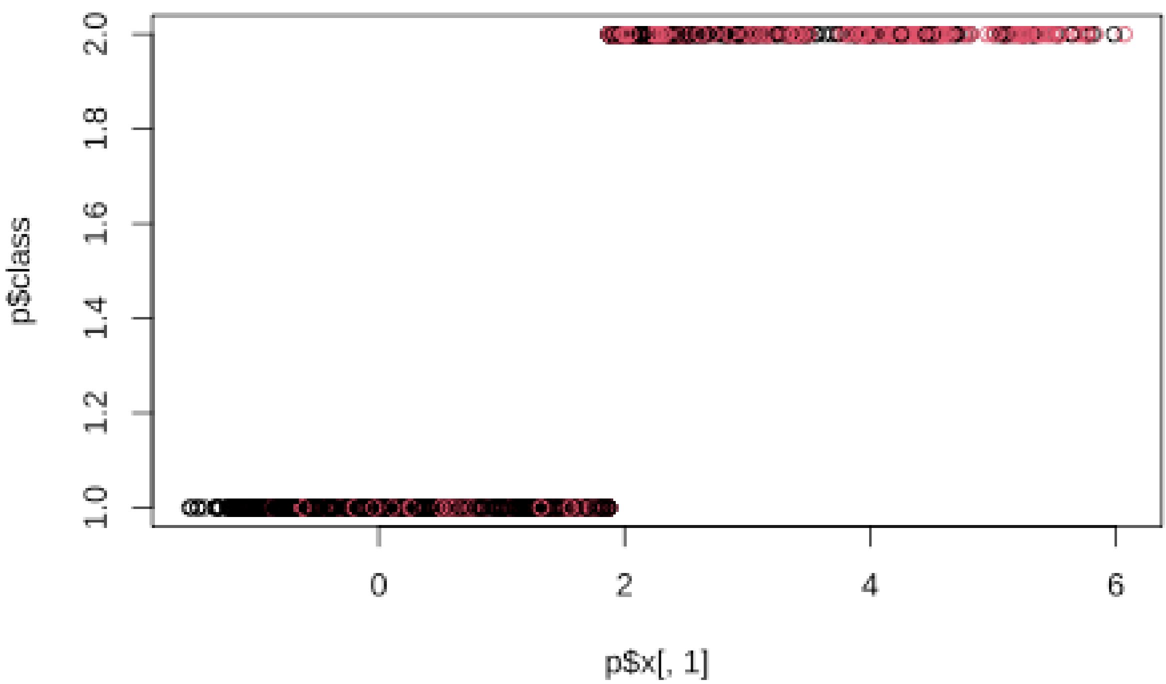
##LDA confusion matrix

```
confusionMatrix(table(Predicted=p$class,Actual=bin.test.class))
```

```
## Confusion Matrix and Statistics
##
##          Actual
## Predicted    0    1
##         0 6936  542
##         1  370  346
##
##                Accuracy : 0.8887
##                  95% CI : (0.8817, 0.8954)
##     No Information Rate : 0.8916
##     P-Value [Acc > NIR] : 0.8084
```

```
##
##                   Kappa : 0.3705
##
##  Mcnemar's Test P-Value : 1.493e-08
##
##             Sensitivity : 0.9494
##             Specificity : 0.3896
##          Pos Pred Value : 0.9275
##          Neg Pred Value : 0.4832
##              Prevalence : 0.8916
##          Detection Rate : 0.8465
##    Detection Prevalence : 0.9126
##       Balanced Accuracy : 0.6695
##
##        'Positive' Class : 0
##
```

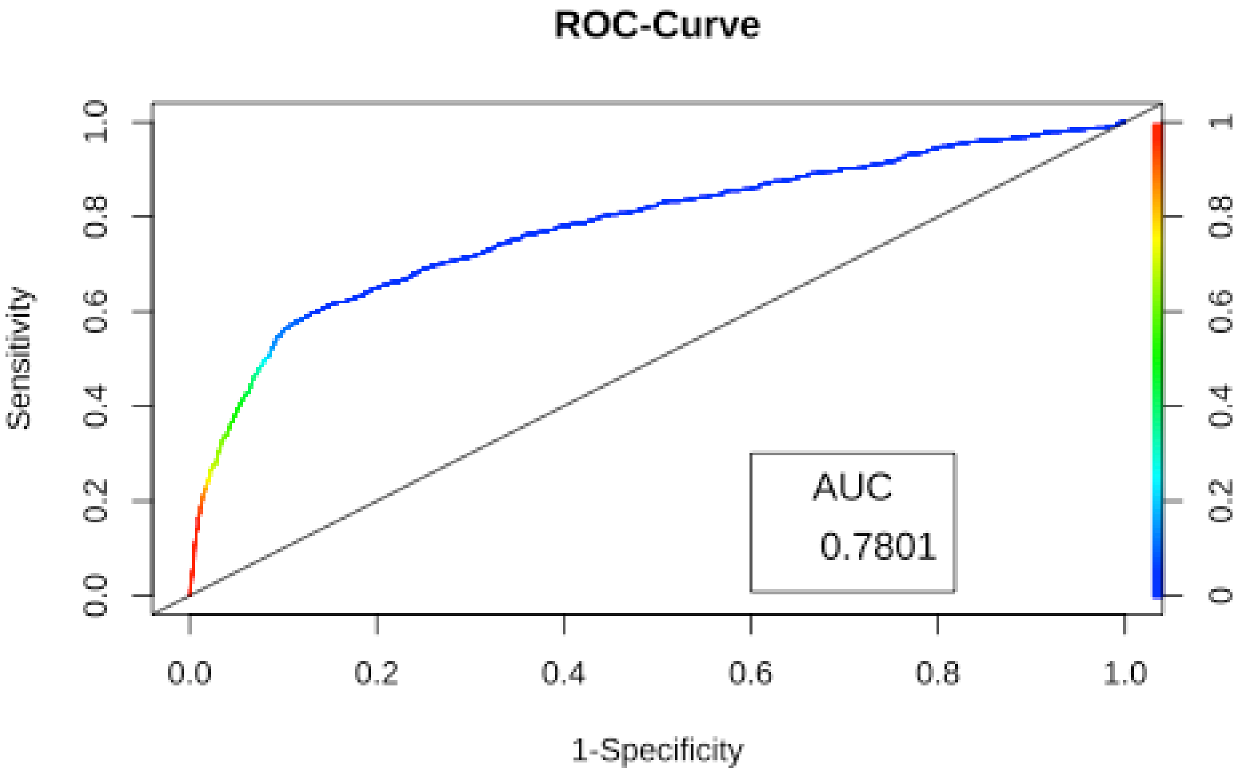
```
par(mfrow=c(1,1))
plot(p$x[,1], p$class, col=bin.test.class)
```




**Insights:** - The LDA has got an accuracy of 88.87% and the above graph corresponds to posterior probability vs output There is a lot of overlap of output with each other but in general a good seperation.

#AUC-ROC-curve

```
pred <- prediction(p$posterior[,2],bin.test.class)
perf <- performance(pred,"tpr","fpr")
plot(perf,colorize=TRUE,main="ROC-Curve",xlab="1-Specificity",ylab="Sensitivity")
abline(a=0,b=1)
auc <- performance(pred,"auc")
auc <- unlist(slot(auc,"y.values"))
auc <- round(auc,4)
legend(.6,.3,auc,title="AUC",cex = 1.2)
```




**Insights:** - The model gives a AUC score of 78%. Since, the output variable is quite imbalanced and separation of output by variables are not so significant even then LDA performs well on the test set.

##QDA

```
qda <- qda(y~.,data=bin.train)
qda
```

```
## Call:
## qda(y ~ ., data = bin.train)
##
## Prior probabilities of groups:
##         0         1
## 0.8862957 0.1137043
##
## Group means:
##         Age   Campaign     Pdays Previous.Contacts Emp.var.rate Cons.price.idx
## 0 0.2827080 0.02978086 0.9854601        0.01895681    0.7607132      0.5474033
## 1 0.2941531 0.01940630 0.7916849        0.07117942    0.4470866      0.4470957
##   Cons.conf.idx Euribor3m Employment.number Job_bluecollar Job_entrepreneur
## 0     0.4265200 0.7208477         0.8037802      0.2391312       0.03676293
## 1     0.4610066 0.3335517         0.4953261      0.1400590       0.02790448
##   Job_housemaid Job_management Job_retired Job_selfemployed Job_services
## 0    0.02550687     0.07214898  0.03538605       0.03493856   0.09948022
## 1    0.02092836     0.07298095  0.09605581       0.03353904   0.06895627
##   Job_student Job_technician Job_unemployed Marital_married Marital_single
## 0  0.01655709      0.1658463     0.02419882       0.6118206      0.2741386
## 1  0.06412664      0.1558895     0.03139254       0.5527234      0.3469278
##   Education_basic.6y Education_basic.9y Education_high.school
## 0         0.05776049          0.1650546             0.2400262
## 1         0.04292997          0.1065200             0.2363831
```

```
##   Education_professional.course Education_university.degree Housing_yes
## 0                     0.1321813                   0.2982341   0.5227015
## 1                     0.1341562                   0.3799302   0.5446740
##   Loan_unknown  Loan_yes Contact_telephone Month_aug   Month_dec Month_jul
## 0   0.02457747 0.1514922         0.3917593 0.1507349 0.002375133 0.1776187
## 1   0.02280655 0.1435471         0.1671586 0.1408640 0.020123424 0.1389858
##   Month_jun   Month_mar Month_may  Month_nov  Month_oct   Month_sep
## 0 0.1307012 0.007194245 0.3530687 0.10133902 0.01118722 0.008536711
## 1 0.1253019 0.058223772 0.1888919 0.09122619 0.06627314 0.052589214
##   Last.Contact.Day_mon Last.Contact.Day_thu Last.Contact.Day_tue
## 0            0.2100100            0.2072562            0.1955871
## 1            0.1843306            0.2208210            0.2063322
##   Last.Contact.Day_wed Poutcome_nonexistent Poutcome_success
## 0            0.1955182            0.8869574       0.01290833
## 1            0.2020392            0.6748055       0.19318487
```

```
pred <- predict(qda,bin.test)
```

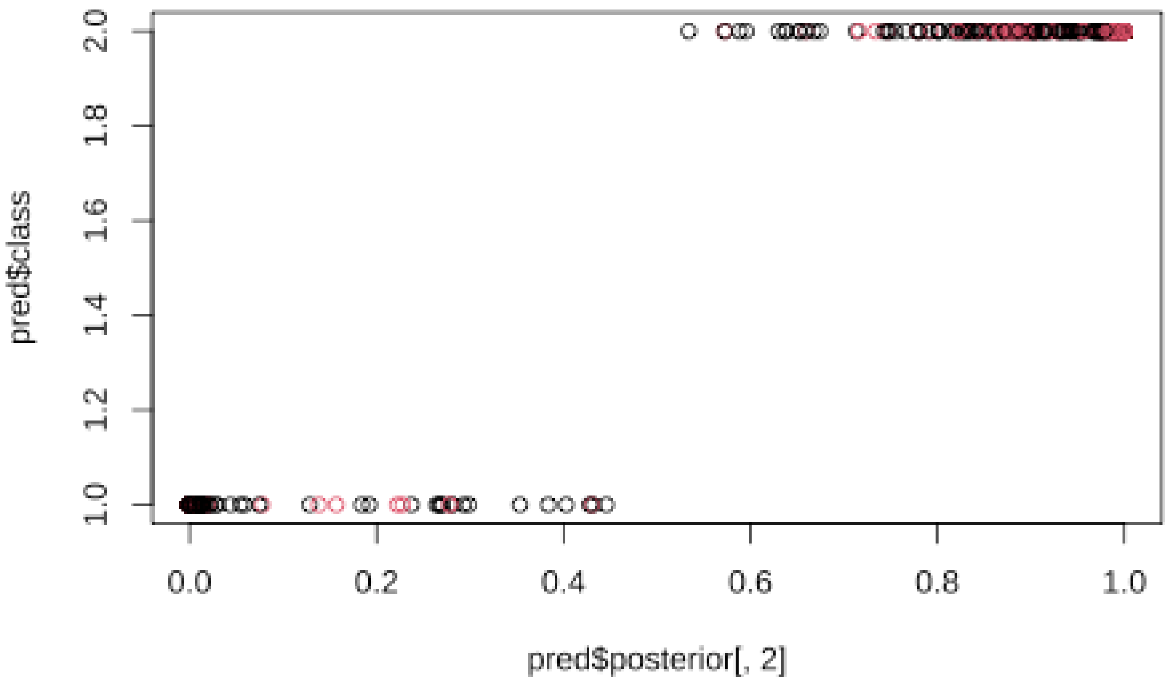
##qda confusion matrix

```
confusionMatrix(table(Predicted=pred$class, bin.test.class))
```

```
## Confusion Matrix and Statistics
##
##          bin.test.class
## Predicted    0    1
##         0 6579  411
##         1  727  477
##
##                Accuracy : 0.8611
##                  95% CI : (0.8534, 0.8685)
##     No Information Rate : 0.8916
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.3785
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.9005
##             Specificity : 0.5372
##          Pos Pred Value : 0.9412
##          Neg Pred Value : 0.3962
##              Prevalence : 0.8916
##          Detection Rate : 0.8029
##    Detection Prevalence : 0.8531
##       Balanced Accuracy : 0.7188
##
##        'Positive' Class : 0
##
```

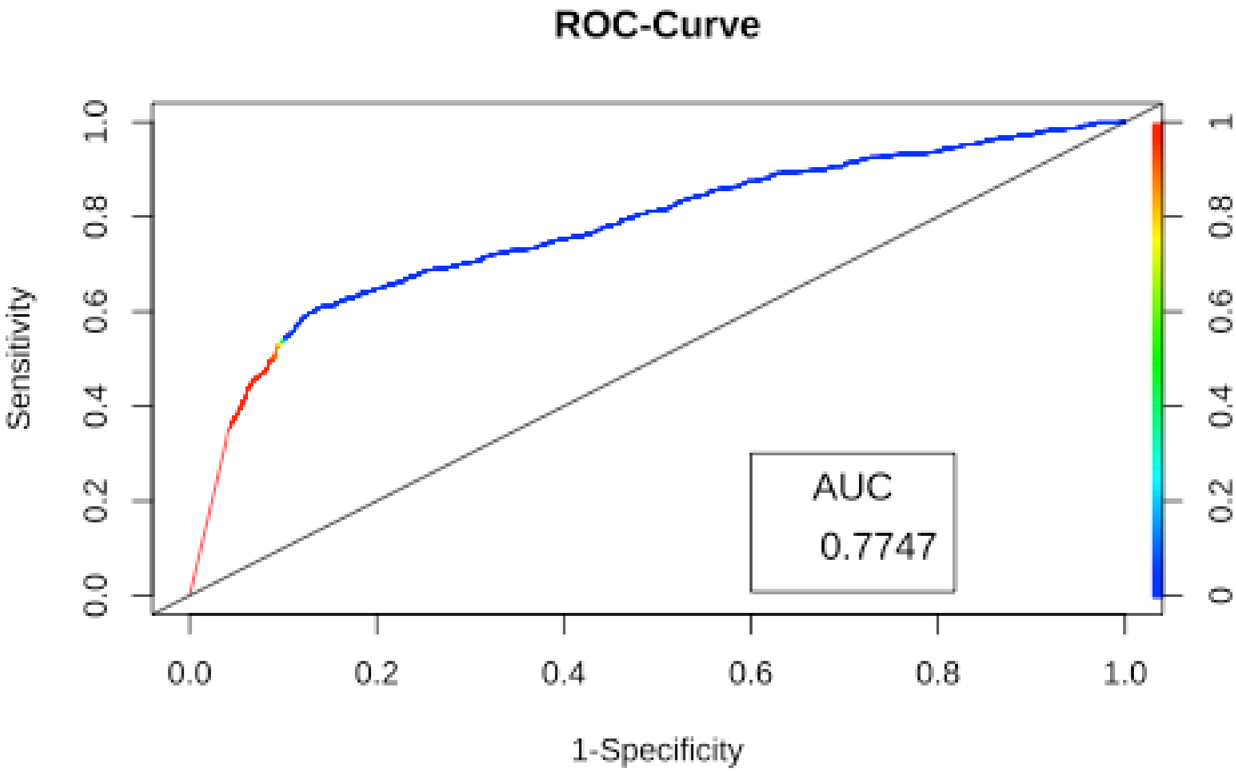
The accuracy is 86%

```
par(mfrow=c(1,1))
plot(pred$posterior[,2], pred$class, col=bin.test.class)
```




AUC-ROC of QDA

```
pred <- prediction(pred$posterior[,2],bin.test.class)
perf <- performance(pred,"tpr","fpr")
plot(perf,colorize=TRUE,main="ROC-Curve",xlab="1-Specificity",ylab="Sensitivity")
abline(a=0,b=1)
auc <- performance(pred,"auc")
auc <- unlist(slot(auc,"y.values"))
auc <- round(auc,4)
legend(.6,.3,auc,title="AUC",cex = 1.2)
```




##Knn

```
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.

##
## Attaching package: 'pROC'

## The following object is masked from 'package:colorspace':
##
##     coords

## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

KNN is a model that classifies according to the distance of the new observations, usually is used the Euclidean distance for this purpose, and uses the voting method to choose the most frequent label. The inductive bias of this model is that similar points should have similar labels.

```
classifier_knn <- knn(train = subset(bin.train, select = -c(y)),
                      test = bin.test,
                      cl = bin.train$y,
                      k = 4)
```

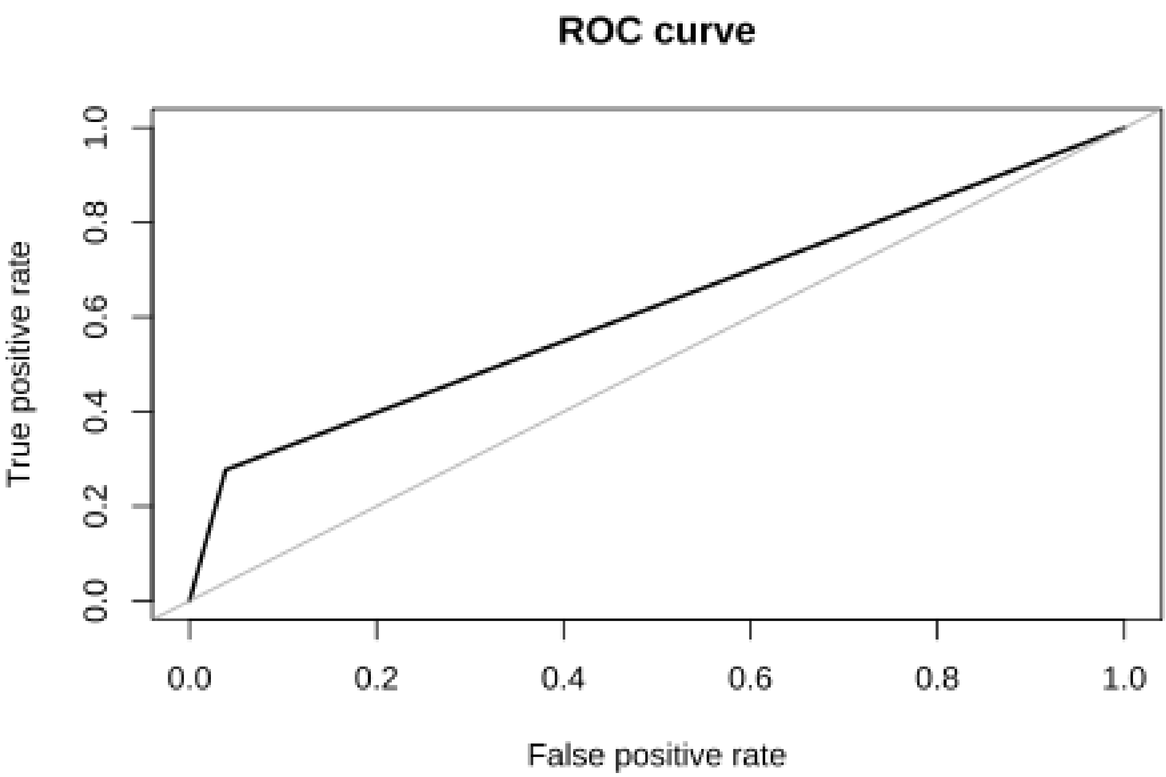
```
# Confusion Matrix
cm <- confusionMatrix(table(Predicted=classifier_knn, Actual=bin.test.class))
cm
```

```
## Confusion Matrix and Statistics
##
##          Actual
## Predicted    0    1
```

```
##          0 7024  642
##          1  282  246
##
##                Accuracy : 0.8872
##                  95% CI : (0.8802, 0.894)
##     No Information Rate : 0.8916
##     P-Value [Acc > NIR] : 0.9022
##
##                   Kappa : 0.2901
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.9614
##             Specificity : 0.2770
##          Pos Pred Value : 0.9163
##          Neg Pred Value : 0.4659
##              Prevalence : 0.8916
##          Detection Rate : 0.8572
##    Detection Prevalence : 0.9356
##       Balanced Accuracy : 0.6192
##
##        'Positive' Class : 0
##
```

The accuracy is 88.7% which is similar to the above presented models

```
roc.curve(bin.test.class, classifier_knn)
```




```
## Area under the curve (AUC): 0.619
```

It is found that knn has an auc of 61% which is quite low when compared with others

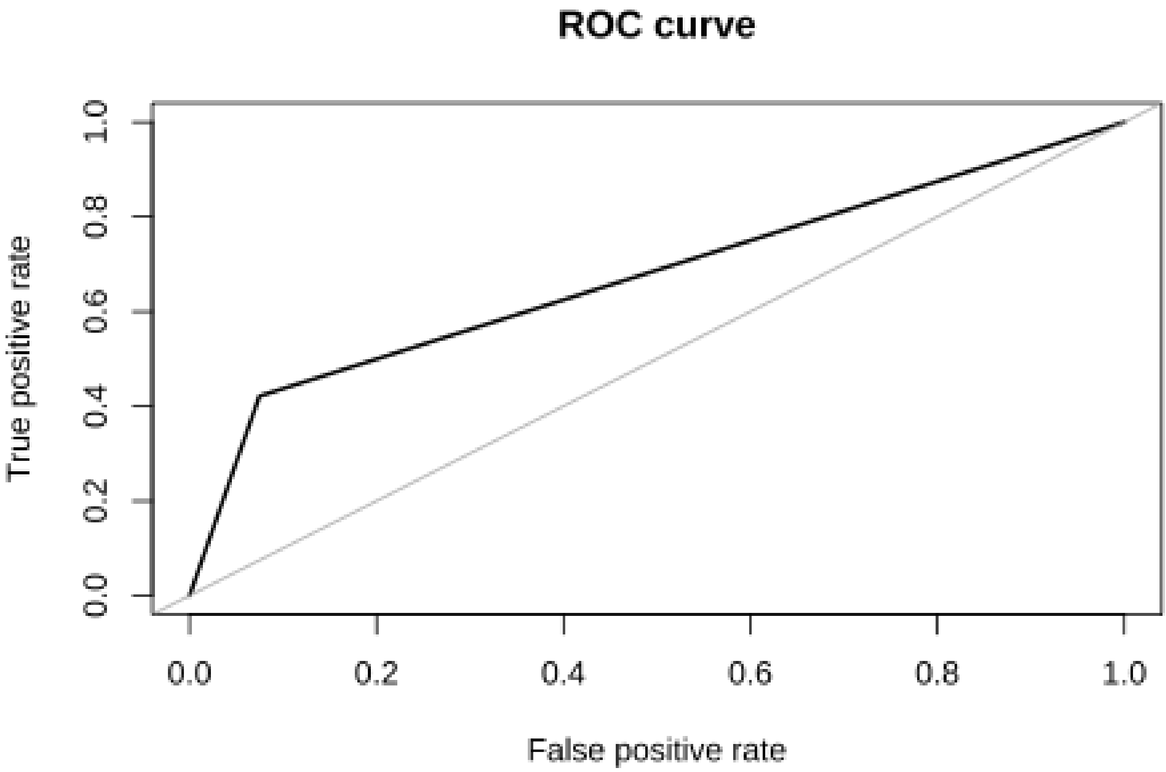
##naive bayes This algorithm follows a probabilistic approach according to the Bayes Theorem, the inductive bias assumes the independence of the predictors. To realize this first the algorithm builds a frequency table, after it creates a likelihood table and finally is calculated the posterior probability for each class and it selects the greatest probability to classify.

```r
classifier_cl <- naiveBayes(y ~ ., data = bin.train, type="prob")

# Predicting on test data'
y_pred <- predict(classifier_cl, newdata = bin.test)

# Confusion Matrix
cm <- table(Predicted=y_pred, Actual=bin.test.class)

# Model Evaluation
confusionMatrix(cm)
```

```
## Confusion Matrix and Statistics
##
##          Actual
## Predicted    0    1
##         0 6760  514
##         1  546  374
##
##                Accuracy : 0.8706
##                  95% CI : (0.8632, 0.8778)
##     No Information Rate : 0.8916
##     P-Value [Acc > NIR] : 1.000
##
##                   Kappa : 0.341
##
##  Mcnemar's Test P-Value : 0.341
##
##             Sensitivity : 0.9253
##             Specificity : 0.4212
##          Pos Pred Value : 0.9293
##          Neg Pred Value : 0.4065
##              Prevalence : 0.8916
##          Detection Rate : 0.8250
##    Detection Prevalence : 0.8877
##       Balanced Accuracy : 0.6732
##
##        'Positive' Class : 0
##
```

The naive bayes is giving an accuracy of 87% less accuracy compared to Knn

```r
roc.curve(bin.test.class, y_pred)
```




```
## Area under the curve (AUC): 0.673
```

Auc score is 67% which is better than the Auc of knn(61%)

##Conclusions - Interms of Accuracy Logistic regression has an accuracy of 90% and other models have an accuracy ranging betwween 86 to 88% - AUC score is similar for Logistic, LDA and QDA whereas It is not significant in case of knn and naive bayes - Job + Education + Contact + Month + Last.Contact.Day + Campaign + Pdays + Poutcome + Emp.var.rate + Cons.price.idx+Cons.conf.idx + Employment.number decides whether client subscribe the term deposit or not