Predicting insurance purchase for Indian farmers

STAT 471/571/701, Fall 2018

Contents


```
knitr::opts_chunk$set(echo = TRUE,
                      tidy = TRUE, fig.width = 7, fig.height = 4,
                      fig.align='left', dev = 'pdf')
if(!require("pacman")) install.packages("pacman")
```

```
## Loading required package: pacman
```

```
if(!require("pROC")) install.packages("pROC")
```

```
## Loading required package: pROC

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

##
## Attaching package: 'pROC'

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

```
if(!require("devtools")) install.packages("devtools")
```

```
## Loading required package: devtools
```

```
if(!require("ranger")) install.packages("ranger")
```

```
## Loading required package: ranger
```

```
if(!require("randomForest")) install.packages("randomForest")
```

```
## Loading required package: randomForest

## randomForest 4.6-14

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:ranger':
##
##     importance
```

```
if(!require("tree")) install.packages("tree")
```

```
## Loading required package: tree
```

```r
if(!require("leaps")) install.packages("leaps")
```

```
## Loading required package: leaps
```

```r
pacman::p_load(dplyr, ggplot2, glmnet, car, corrplot)
library(pROC)
library(devtools)
library(rpart)
library(ranger)
library(randomForest)
library(tree)
```

Setup and data cleansing

```r
caravan_kaggle<- read.csv("caravan-insurance-challenge.csv", header = T)
caravan_kaggle_2<- caravan_kaggle #create a copy
```

```r
summary(caravan_kaggle)
```

```
##    ORIGIN        MOSTYPE         MAANTHUI         MGEMOMV
##  test :4000   Min.   : 1.00   Min.   : 1.000   Min.   :1.000
##  train:5822   1st Qu.:10.00   1st Qu.: 1.000   1st Qu.:2.000
##               Median :30.00   Median : 1.000   Median :3.000
##               Mean   :24.25   Mean   : 1.109   Mean   :2.678
##               3rd Qu.:35.00   3rd Qu.: 1.000   3rd Qu.:3.000
##               Max.   :41.00   Max.   :10.000   Max.   :6.000
##     MGEMLEEF        MOSHOOFD          MGODRK           MGODPR
##  Min.   :1.000   Min.   : 1.000   Min.   :0.0000   Min.   :0.000
##  1st Qu.:2.000   1st Qu.: 3.000   1st Qu.:0.0000   1st Qu.:4.000
##  Median :3.000   Median : 7.000   Median :0.0000   Median :5.000
##  Mean   :2.996   Mean   : 5.779   Mean   :0.7007   Mean   :4.638
##  3rd Qu.:3.000   3rd Qu.: 8.000   3rd Qu.:1.0000   3rd Qu.:6.000
##  Max.   :6.000   Max.   :10.000   Max.   :9.0000   Max.   :9.000
##      MGODOV         MGODGE          MRELGE          MRELSA
##  Min.   :0.00   Min.   :0.000   Min.   :0.000   Min.   :0.0000
##  1st Qu.:0.00   1st Qu.:2.000   1st Qu.:5.000   1st Qu.:0.0000
##  Median :1.00   Median :3.000   Median :6.000   Median :1.0000
##  Mean   :1.05   Mean   :3.263   Mean   :6.189   Mean   :0.8731
##  3rd Qu.:2.00   3rd Qu.:4.000   3rd Qu.:7.000   3rd Qu.:1.0000
##  Max.   :5.00   Max.   :9.000   Max.   :9.000   Max.   :7.0000
##      MRELOV         MFALLEEN        MFGEKIND        MFWEKIND
##  Min.   :0.000   Min.   :0.000   Min.   :0.000   Min.   :0.000
##  1st Qu.:1.000   1st Qu.:0.000   1st Qu.:2.000   1st Qu.:3.000
##  Median :2.000   Median :2.000   Median :3.000   Median :4.000
##  Mean   :2.287   Mean   :1.887   Mean   :3.237   Mean   :4.303
##  3rd Qu.:3.000   3rd Qu.:3.000   3rd Qu.:4.000   3rd Qu.:6.000
##  Max.   :9.000   Max.   :9.000   Max.   :9.000   Max.   :9.000
##     MOPLHOOG        MOPLMIDD        MOPLLAAG        MBERHOOG
##  Min.   :0.000   Min.   :0.000   Min.   :0.000   Min.   :0.000
##  1st Qu.:0.000   1st Qu.:2.000   1st Qu.:3.000   1st Qu.:0.000
##  Median :1.000   Median :3.000   Median :5.000   Median :2.000
##  Mean   :1.485   Mean   :3.307   Mean   :4.592   Mean   :1.899
##  3rd Qu.:2.000   3rd Qu.:4.000   3rd Qu.:6.000   3rd Qu.:3.000
##  Max.   :9.000   Max.   :9.000   Max.   :9.000   Max.   :9.000
```

```
##     MBERZELF         MBERBOER         MBERMIDD        MBERARBG    
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.000   Min.   :0.000  
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:2.000   1st Qu.:1.000  
##  Median :0.0000   Median :0.0000   Median :3.000   Median :2.000  
##  Mean   :0.4033   Mean   :0.5457   Mean   :2.877   Mean   :2.227  
##  3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:4.000   3rd Qu.:3.000  
##  Max.   :5.0000   Max.   :9.0000   Max.   :9.000   Max.   :9.000  
##     MBERARBO          MSKA           MSKB1           MSKB2      
##  Min.   :0.000   Min.   :0.000   Min.   :0.000   Min.   :0.000  
##  1st Qu.:1.000   1st Qu.:0.000   1st Qu.:1.000   1st Qu.:1.000  
##  Median :2.000   Median :1.000   Median :2.000   Median :2.000  
##  Mean   :2.291   Mean   :1.651   Mean   :1.595   Mean   :2.205  
##  3rd Qu.:3.000   3rd Qu.:2.000   3rd Qu.:2.000   3rd Qu.:3.000  
##  Max.   :9.000   Max.   :9.000   Max.   :9.000   Max.   :9.000  
##       MSKC            MSKD           MHHUUR          MHKOOP     
##  Min.   :0.000   Min.   :0.000   Min.   :0.000   Min.   :0.000  
##  1st Qu.:2.000   1st Qu.:0.000   1st Qu.:2.000   1st Qu.:2.000  
##  Median :4.000   Median :1.000   Median :4.000   Median :5.000  
##  Mean   :3.742   Mean   :1.068   Mean   :4.188   Mean   :4.819  
##  3rd Qu.:5.000   3rd Qu.:2.000   3rd Qu.:7.000   3rd Qu.:7.000  
##  Max.   :9.000   Max.   :9.000   Max.   :9.000   Max.   :9.000  
##      MAUT1           MAUT2           MAUT0          MZFONDS     
##  Min.   :0.000   Min.   :0.000   Min.   :0.000   Min.   :0.000  
##  1st Qu.:5.000   1st Qu.:0.000   1st Qu.:0.000   1st Qu.:5.000  
##  Median :6.000   Median :1.000   Median :2.000   Median :7.000  
##  Mean   :6.023   Mean   :1.336   Mean   :1.957   Mean   :6.254  
##  3rd Qu.:7.000   3rd Qu.:2.000   3rd Qu.:3.000   3rd Qu.:8.000  
##  Max.   :9.000   Max.   :9.000   Max.   :9.000   Max.   :9.000  
##      MZPART         MINKM30         MINK3045        MINK4575    
##  Min.   :0.000   Min.   :0.000   Min.   :0.000   Min.   :0.000  
##  1st Qu.:1.000   1st Qu.:1.000   1st Qu.:2.000   1st Qu.:1.000  
##  Median :2.000   Median :2.000   Median :4.000   Median :3.000  
##  Mean   :2.751   Mean   :2.577   Mean   :3.505   Mean   :2.739  
##  3rd Qu.:4.000   3rd Qu.:4.000   3rd Qu.:5.000   3rd Qu.:4.000  
##  Max.   :9.000   Max.   :9.000   Max.   :9.000   Max.   :9.000  
##     MINK7512         MINK123M        MINKGEM         MKOOPKLA   
##  Min.   :0.0000   Min.   :0.000   Min.   :0.000   Min.   :1.00  
##  1st Qu.:0.0000   1st Qu.:0.000   1st Qu.:3.000   1st Qu.:3.00  
##  Median :0.0000   Median :0.000   Median :4.000   Median :4.00  
##  Mean   :0.8085   Mean   :0.208   Mean   :3.805   Mean   :4.26  
##  3rd Qu.:1.0000   3rd Qu.:0.000   3rd Qu.:4.000   3rd Qu.:6.00  
##  Max.   :9.0000   Max.   :9.000   Max.   :9.000   Max.   :8.00  
##     PWAPART          PWABEDR            PWALAND           PPERSAUT    
##  Min.   :0.0000   Min.   :0.00000   Min.   :0.00000   Min.   :0.000  
##  1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.000  
##  Median :0.0000   Median :0.00000   Median :0.00000   Median :5.000  
##  Mean   :0.7649   Mean   :0.03889   Mean   :0.07371   Mean   :2.956  
##  3rd Qu.:2.0000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:6.000  
##  Max.   :3.0000   Max.   :6.00000   Max.   :4.00000   Max.   :9.000  
##     PBESAUT            PMOTSCO          PVRAAUT             PAANHANG      
##  Min.   :0.00000   Min.   :0.0000   Min.   :0.000000   Min.   :0.00000  
##  1st Qu.:0.00000   1st Qu.:0.0000   1st Qu.:0.000000   1st Qu.:0.00000  
##  Median :0.00000   Median :0.0000   Median :0.000000   Median :0.00000  
##  Mean   :0.05488   Mean   :0.1708   Mean   :0.008858   Mean   :0.01934  
```

```
##  3rd Qu.:0.00000   3rd Qu.:0.0000   3rd Qu.:0.000000   3rd Qu.:0.00000
##  Max.   :7.00000   Max.   :7.0000   Max.   :9.000000   Max.   :5.00000
##     PTRACTOR           PWERKT            PBROM           PLEVEN
##  Min.   :0.00000   Min.   :0.0000   Min.   :0.000   Min.   :0.0000
##  1st Qu.:0.00000   1st Qu.:0.0000   1st Qu.:0.000   1st Qu.:0.0000
##  Median :0.00000   Median :0.0000   Median :0.000   Median :0.0000
##  Mean   :0.09356   Mean   :0.0115   Mean   :0.215   Mean   :0.2023
##  3rd Qu.:0.00000   3rd Qu.:0.0000   3rd Qu.:0.000   3rd Qu.:0.0000
##  Max.   :7.00000   Max.   :6.0000   Max.   :6.000   Max.   :9.0000
##     PPERSONG          PGEZONG            PWAOREG            PBRAND
##  Min.   :0.0000   Min.   :0.00000   Min.   :0.00000   Min.   :0.000
##  1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.000
##  Median :0.0000   Median :0.00000   Median :0.00000   Median :2.000
##  Mean   :0.0115   Mean   :0.01873   Mean   :0.02331   Mean   :1.849
##  3rd Qu.:0.0000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:4.000
##  Max.   :6.0000   Max.   :3.00000   Max.   :7.00000   Max.   :8.000
##     PZEILPL            PPLEZIER           PFIETS            PINBOED
##  Min.   :0.000000   Min.   :0.00000   Min.   :0.00000   Min.   :0.0000
##  1st Qu.:0.000000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.0000
##  Median :0.000000   Median :0.00000   Median :0.00000   Median :0.0000
##  Mean   :0.001629   Mean   :0.01527   Mean   :0.02535   Mean   :0.0167
##  3rd Qu.:0.000000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.0000
##  Max.   :3.000000   Max.   :6.00000   Max.   :1.00000   Max.   :6.0000
##     PBYSTAND          AWAPART      AWABEDR           AWALAND
##  Min.   :0.00000   Min.   :0.0   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.00000   1st Qu.:0.0   1st Qu.:0.00000   1st Qu.:0.00000
##  Median :0.00000   Median :0.0   Median :0.00000   Median :0.00000
##  Mean   :0.04541   Mean   :0.4   Mean   :0.01405   Mean   :0.02128
##  3rd Qu.:0.00000   3rd Qu.:1.0   3rd Qu.:0.00000   3rd Qu.:0.00000
##  Max.   :5.00000   Max.   :2.0   Max.   :5.00000   Max.   :1.00000
##     APERSAUT          ABESAUT           AMOTSCO            AVRAAUT
##  Min.   : 0.0000   Min.   :0.0000   Min.   :0.00000   Min.   :0.00000
##  1st Qu.: 0.0000   1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:0.00000
##  Median : 1.0000   Median :0.0000   Median :0.00000   Median :0.00000
##  Mean   : 0.5572   Mean   :0.0111   Mean   :0.04022   Mean   :0.00224
##  3rd Qu.: 1.0000   3rd Qu.:0.0000   3rd Qu.:0.00000   3rd Qu.:0.00000
##  Max.   :12.0000   Max.   :5.0000   Max.   :8.00000   Max.   :4.00000
##     AAANHANG          ATRACTOR            AWERKT             ABROM
##  Min.   :0.0000   Min.   :0.00000   Min.   :0.000000   Min.   :0.00000
##  1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:0.000000   1st Qu.:0.00000
##  Median :0.0000   Median :0.00000   Median :0.000000   Median :0.00000
##  Mean   :0.0114   Mean   :0.03441   Mean   :0.005192   Mean   :0.07107
##  3rd Qu.:0.0000   3rd Qu.:0.00000   3rd Qu.:0.000000   3rd Qu.:0.00000
##  Max.   :3.0000   Max.   :6.00000   Max.   :6.000000   Max.   :3.00000
##      ALEVEN           APERSONG            AGEZONG
##  Min.   :0.00000   Min.   :0.000000   Min.   :0.000000
##  1st Qu.:0.00000   1st Qu.:0.000000   1st Qu.:0.000000
##  Median :0.00000   Median :0.000000   Median :0.000000
##  Mean   :0.07982   Mean   :0.004582   Mean   :0.007941
##  3rd Qu.:0.00000   3rd Qu.:0.000000   3rd Qu.:0.000000
##  Max.   :8.00000   Max.   :1.000000   Max.   :1.000000
##     AWAOREG             ABRAND          AZEILPL             APLEZIER
##  Min.   :0.000000   Min.   :0.000   Min.   :0.0000000   Min.   :0.000000
##  1st Qu.:0.000000   1st Qu.:0.000   1st Qu.:0.0000000   1st Qu.:0.000000
```

```
##  Median :0.000000   Median :1.000   Median :0.0000000   Median :0.000000
##  Mean   :0.004276   Mean   :0.574   Mean   :0.0009163   Mean   :0.005091
##  3rd Qu.:0.000000   3rd Qu.:1.000   3rd Qu.:0.0000000   3rd Qu.:0.000000
##  Max.   :2.000000   Max.   :7.000   Max.   :1.0000000   Max.   :2.000000
##      AFIETS          AINBOED           ABYSTAND          CARAVAN
##  Min.   :0.00000   Min.   :0.00000   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000
##  Median :0.00000   Median :0.00000   Median :0.00000   Median :0.00000
##  Mean   :0.03146   Mean   :0.00845   Mean   :0.01385   Mean   :0.05966
##  3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000
##  Max.   :4.00000   Max.   :2.00000   Max.   :2.00000   Max.   :1.00000
```

```
str(caravan_kaggle)
```

```
## 'data.frame':    9822 obs. of  87 variables:
##  $ ORIGIN  : Factor w/ 2 levels "test","train": 2 2 2 2 2 2 2 2 2 2 ...
##  $ MOSTYPE : int  33 37 37 9 40 23 39 33 33 11 ...
##  $ MAANTHUI: int  1 1 1 1 1 1 2 1 1 2 ...
##  $ MGEMOMV : int  3 2 2 3 4 2 3 2 2 3 ...
##  $ MGEMLEEF: int  2 2 2 3 2 1 2 3 4 3 ...
##  $ MOSHOOFD: int  8 8 8 3 10 5 9 8 8 3 ...
##  $ MGODRK  : int  0 1 0 2 1 0 2 0 0 3 ...
##  $ MGODPR  : int  5 4 4 3 4 5 2 7 1 5 ...
##  $ MGODOV  : int  1 1 2 2 1 0 0 0 3 0 ...
##  $ MGODGE  : int  3 4 4 4 4 5 5 2 6 2 ...
##  $ MRELGE  : int  7 6 3 5 7 0 7 7 6 7 ...
##  $ MRELSA  : int  0 2 2 2 1 6 2 2 0 0 ...
##  $ MRELOV  : int  2 2 4 2 2 3 0 0 3 2 ...
##  $ MFALLEEN: int  1 0 4 2 2 3 0 0 3 2 ...
##  $ MFGEKIND: int  2 4 4 3 4 5 3 5 3 2 ...
##  $ MFWEKIND: int  6 5 2 4 4 2 6 4 3 6 ...
##  $ MOPLHOOG: int  1 0 0 3 5 0 0 0 0 0 ...
##  $ MOPLMIDD: int  2 5 5 4 4 5 4 3 1 4 ...
##  $ MOPLLAAG: int  7 4 4 2 0 4 5 6 8 5 ...
##  $ MBERHOOG: int  1 0 0 4 0 2 0 2 1 2 ...
##  $ MBERZELF: int  0 0 0 0 5 0 0 0 1 0 ...
##  $ MBERBOER: int  1 0 0 0 4 0 0 0 0 0 ...
##  $ MBERMIDD: int  2 5 7 3 0 4 4 2 1 3 ...
##  $ MBERARBG: int  5 0 0 1 0 2 1 5 8 3 ...
##  $ MBERARBO: int  2 4 2 2 0 2 5 2 1 3 ...
##  $ MSKA    : int  1 0 0 3 9 2 0 2 1 1 ...
##  $ MSKB1   : int  1 2 5 2 0 2 1 1 1 2 ...
##  $ MSKB2   : int  2 3 0 1 0 2 4 2 0 1 ...
##  $ MSKC    : int  6 5 4 4 0 4 5 5 8 4 ...
##  $ MSKD    : int  1 0 0 0 0 2 0 2 1 2 ...
##  $ MHHUUR  : int  1 2 7 5 4 9 6 0 9 0 ...
##  $ MHKOOP  : int  8 7 2 4 5 0 3 9 0 9 ...
##  $ MAUT1   : int  8 7 7 9 6 5 8 4 5 6 ...
##  $ MAUT2   : int  0 1 0 0 2 3 0 4 2 1 ...
##  $ MAUT0   : int  1 2 2 0 1 3 1 2 3 2 ...
##  $ MZFONDS : int  8 6 9 7 5 9 9 6 7 6 ...
##  $ MZPART  : int  1 3 0 2 4 0 0 3 2 3 ...
##  $ MINKM30 : int  0 2 4 1 0 5 4 2 7 2 ...
##  $ MINK3045: int  4 0 5 5 0 2 3 5 2 3 ...
##  $ MINK4575: int  5 5 0 3 9 3 3 3 1 3 ...
```

```
## $ MINK7512: int 0 2 0 0 0 0 0 0 0 1 ...
## $ MINK123M: int 0 0 0 0 0 0 0 0 0 0 ...
## $ MINKGEM : int 4 5 3 4 6 3 3 3 2 4 ...
## $ MKOOPKLA: int 3 4 4 4 3 3 5 3 3 7 ...
## $ PWAPART : int 0 2 2 0 0 0 0 0 0 2 ...
## $ PWABEDR : int 0 0 0 0 0 0 0 0 0 0 ...
## $ PWALAND : int 0 0 0 0 0 0 0 0 0 0 ...
## $ PPERSAUT: int 6 0 6 6 0 6 6 0 5 0 ...
## $ PBESAUT : int 0 0 0 0 0 0 0 0 0 0 ...
## $ PMOTSCO : int 0 0 0 0 0 0 0 0 0 0 ...
## $ PVRAAUT : int 0 0 0 0 0 0 0 0 0 0 ...
## $ PAANHANG: int 0 0 0 0 0 0 0 0 0 0 ...
## $ PTRACTOR: int 0 0 0 0 0 0 0 0 0 0 ...
## $ PWERKT  : int 0 0 0 0 0 0 0 0 0 0 ...
## $ PBROM   : int 0 0 0 0 0 0 0 3 0 0 ...
## $ PLEVEN  : int 0 0 0 0 0 0 0 0 0 0 ...
## $ PPERSONG: int 0 0 0 0 0 0 0 0 0 0 ...
## $ PGEZONG : int 0 0 0 0 0 0 0 0 0 0 ...
## $ PWAOREG : int 0 0 0 0 0 0 0 0 0 0 ...
## $ PBRAND  : int 5 2 2 2 6 0 0 0 0 3 ...
## $ PZEILPL : int 0 0 0 0 0 0 0 0 0 0 ...
## $ PPLEZIER: int 0 0 0 0 0 0 0 0 0 0 ...
## $ PFIETS  : int 0 0 0 0 0 0 0 0 0 0 ...
## $ PINBOED : int 0 0 0 0 0 0 0 0 0 0 ...
## $ PBYSTAND: int 0 0 0 0 0 0 0 0 0 0 ...
## $ AWAPART : int 0 2 1 0 0 0 0 0 0 1 ...
## $ AWABEDR : int 0 0 0 0 0 0 0 0 0 0 ...
## $ AWALAND : int 0 0 0 0 0 0 0 0 0 0 ...
## $ APERSAUT: int 1 0 1 1 0 1 1 0 1 0 ...
## $ ABESAUT : int 0 0 0 0 0 0 0 0 0 0 ...
## $ AMOTSCO : int 0 0 0 0 0 0 0 0 0 0 ...
## $ AVRAAUT : int 0 0 0 0 0 0 0 0 0 0 ...
## $ AAANHANG: int 0 0 0 0 0 0 0 0 0 0 ...
## $ ATRACTOR: int 0 0 0 0 0 0 0 0 0 0 ...
## $ AWERKT  : int 0 0 0 0 0 0 0 0 0 0 ...
## $ ABROM   : int 0 0 0 0 0 0 0 1 0 0 ...
## $ ALEVEN  : int 0 0 0 0 0 0 0 0 0 0 ...
## $ APERSONG: int 0 0 0 0 0 0 0 0 0 0 ...
## $ AGEZONG : int 0 0 0 0 0 0 0 0 0 0 ...
## $ AWAOREG : int 0 0 0 0 0 0 0 0 0 0 ...
## $ ABRAND  : int 1 1 1 1 1 0 0 0 0 1 ...
## $ AZEILPL : int 0 0 0 0 0 0 0 0 0 0 ...
## $ APLEZIER: int 0 0 0 0 0 0 0 0 0 0 ...
## $ AFIETS  : int 0 0 0 0 0 0 0 0 0 0 ...
## $ AINBOED : int 0 0 0 0 0 0 0 0 0 0 ...
## $ ABYSTAND: int 0 0 0 0 0 0 0 0 0 0 ...
## $ CARAVAN : int 0 0 0 0 0 0 0 0 0 0 ...
```

```
#These are factors as per the table. It may help in interpretation to rename the variable's levels. It

#refactoring
caravan_kaggle$MOSTYPE <- factor(caravan_kaggle$MOSTYPE,
                                 levels=c(1:41),
                                 labels=c("High Income, expensive child",
```

```
                                       "Very Important Provincials",
                                       "High status seniors",
                                       "Affluent senior apartments",
                                       "Mixed seniors",
                                       "Career and childcare",
                                       "Dinki's (Double income no kids)",
                                       "Middle class families",
                                       "Modern, complete families",
                                       "Stable family","Family starters",
                                       "Affluent young families",
                                       "Young all american family",
                                       "Junior cosmopolitans",
                                       "Senior cosmopolitans",
                                       "Students in apartments",
                                       "Fresh masters in the city",
                                       "Single youth",
                                       "Suburban youth",
                                       "Ethnically diverse",
                                       "Young urban have-nots",
                                       "Mixed apartment dwellers",
                                       "Young and rising",
                                       "Young, low educated",
                                       "Yound seniros in the city",
                                       "Own home elderly",
                                       "Seniors in apartments",
                                       "Residential elderly",
                                       "Porchless seniors: no front yard",
                                       "Religious elderly singles",
                                       "Low income catholics",
                                       "Mixed seniors2",
                                       "Lower class large families",
                                       "Large family,employed child",
                                       "Village families",
                                       "Couples with teens 'Married with children'",
                                       "Mixed small town dwellers",
                                       "Traditional families",
                                       "Large religous families",
                                       "Large family farms",
                                       "Mixed rurals"))

#Average Age Refactor
caravan_kaggle$MGEMLEEF <- factor(caravan_kaggle$MGEMLEEF,
                     levels=c(1:6),
                     labels=c("20-30 years",
                              "30-40 years",
                              "40-50 years",
                              "50-60 years",
                              "60-70 years",
                              "70-80 years"))

#Custom Main Type Refactor
caravan_kaggle$MOSHOOFD <- factor(caravan_kaggle$MOSHOOFD,
                                levels=(1:10),
```

```
                                labels=c("Successful hedonists",
                                         "Driven Growers",
                                         "Average Family",
                                         "Career Loners",
                                         "Living well",
                                         "Cruising Seniors",
                                         "Retired and Religious",
                                         "Family with grown ups",
                                         "Conservatie Families",
                                         "Farmers"))
```

```
#Percentages Refactor
for (i in which(colnames(caravan_kaggle)=="MGODRK"):which(colnames(caravan_kaggle)=="MKOOPKLA")){
  caravan_kaggle[,i] <- factor(caravan_kaggle[,i],
                   levels=c(0:9),
                   labels=c("0%",
                            "1-10%",
                            "11-23%",
                            "24-36%",
                            "37-49%",
                            "50-62%",
                            "63-75%",
                            "76-88%",
                            "89-99%",
                            "100%"))
}
```

```
#Number of Refactor
for (i in which(colnames(caravan_kaggle)=="PWAPART"):which(colnames(caravan_kaggle)=="ABYSTAND")){
  caravan_kaggle[,i] <- factor(caravan_kaggle[,i],
                   levels=c(0:9),
                   labels=c("0",
                            "1-49",
                            "50-99",
                            "100-199",
                            "200-499",
                            "500-999",
                            "1000-4999",
                            "5000-9999",
                            "10,000-19,999",
                            ">=20,000"))
}

#Set class label as factor
caravan_kaggle$CARAVAN <- factor(caravan_kaggle$CARAVAN,levels=c("0","1"))
```

```
#Remove empty rows
sum(is.na(caravan_kaggle)) #find missing values
```

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

```
caravan_kaggle<-caravan_kaggle[complete.cases(caravan_kaggle),]
```

```
#Remove ORIGIN
caravan_kaggle<-caravan_kaggle[,-1]
```

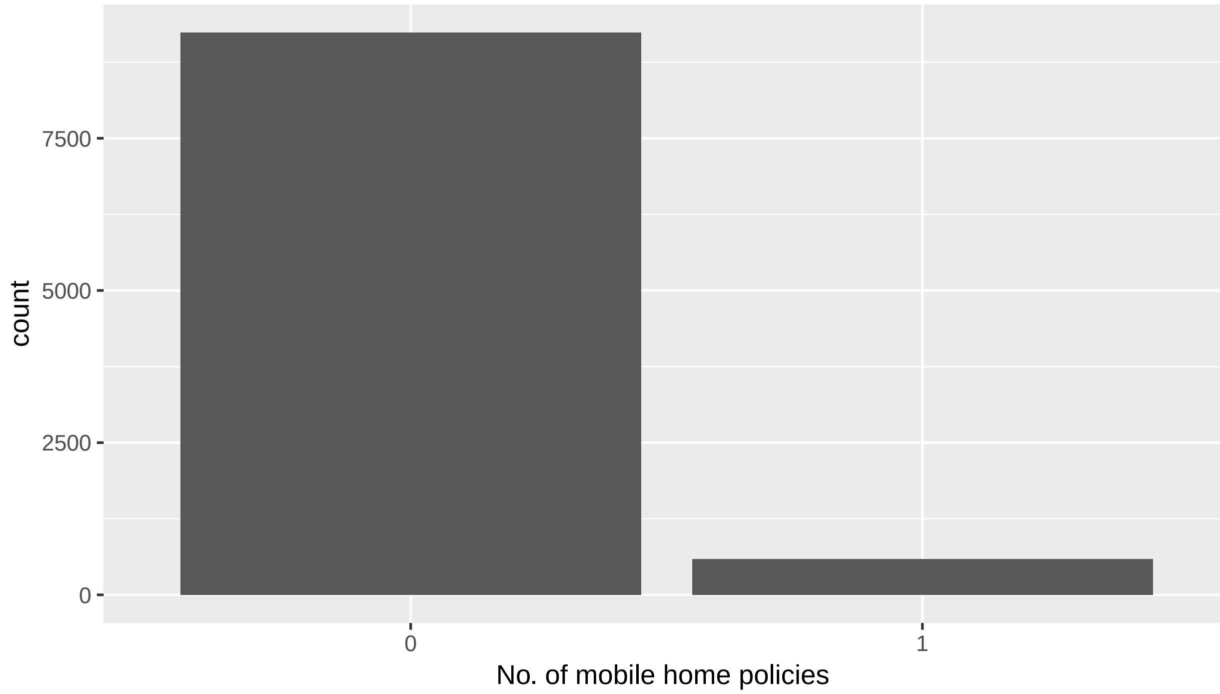
Exploratory data analysis

```r
str(caravan_kaggle)
```

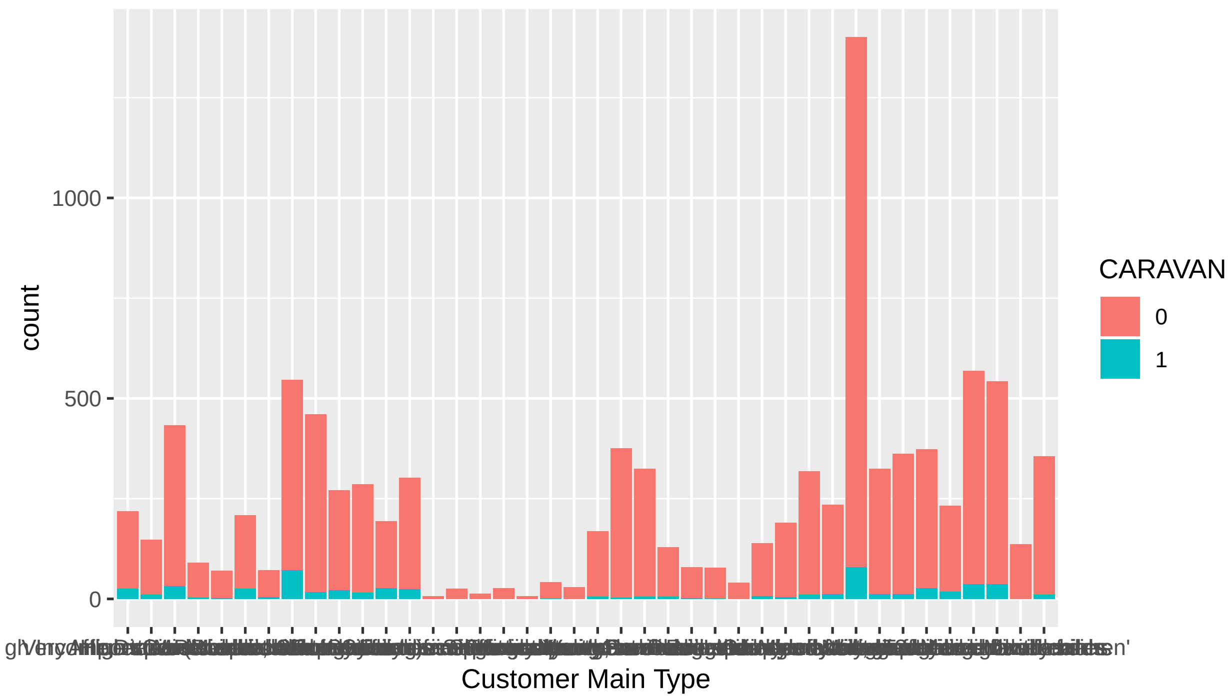
```
## 'data.frame':	9821 obs. of  86 variables:
##  $ MOSTYPE : Factor w/ 41 levels "High Income, expensive child",..: 33 37 37 9 40 23 39 33 33 11 ...
##  $ MAANTHUI: int  1 1 1 1 1 1 2 1 1 2 ...
##  $ MGEMOMV : int  3 2 2 3 4 2 3 2 2 3 ...
##  $ MGEMLEEF: Factor w/ 6 levels "20-30 years",..: 2 2 2 3 2 1 2 3 4 3 ...
##  $ MOSHOOFD: Factor w/ 10 levels "Successful hedonists",..: 8 8 8 3 10 5 9 8 8 3 ...
##  $ MGODRK  : Factor w/ 10 levels "0%","1-10%","11-23%",..: 1 2 1 3 2 1 3 1 1 4 ...
##  $ MGODPR  : Factor w/ 10 levels "0%","1-10%","11-23%",..: 6 5 5 4 5 6 3 8 2 6 ...
##  $ MGODOV  : Factor w/ 10 levels "0%","1-10%","11-23%",..: 2 2 3 3 2 1 1 1 4 1 ...
##  $ MGODGE  : Factor w/ 10 levels "0%","1-10%","11-23%",..: 4 5 5 5 5 6 6 3 7 3 ...
##  $ MRELGE  : Factor w/ 10 levels "0%","1-10%","11-23%",..: 8 7 4 6 8 1 8 8 7 8 ...
##  $ MRELSA  : Factor w/ 10 levels "0%","1-10%","11-23%",..: 1 3 3 3 2 7 3 3 1 1 ...
##  $ MRELOV  : Factor w/ 10 levels "0%","1-10%","11-23%",..: 3 3 5 3 3 4 1 1 4 3 ...
##  $ MFALLEEN: Factor w/ 10 levels "0%","1-10%","11-23%",..: 2 1 5 3 3 4 1 1 4 3 ...
##  $ MFGEKIND: Factor w/ 10 levels "0%","1-10%","11-23%",..: 3 5 5 4 5 6 4 6 4 3 ...
##  $ MFWEKIND: Factor w/ 10 levels "0%","1-10%","11-23%",..: 7 6 3 5 5 3 7 5 4 7 ...
##  $ MOPLHOOG: Factor w/ 10 levels "0%","1-10%","11-23%",..: 2 1 1 4 6 1 1 1 1 1 ...
##  $ MOPLMIDD: Factor w/ 10 levels "0%","1-10%","11-23%",..: 3 6 6 5 5 6 5 4 2 5 ...
##  $ MOPLLAAG: Factor w/ 10 levels "0%","1-10%","11-23%",..: 8 5 5 3 1 5 6 7 9 6 ...
##  $ MBERHOOG: Factor w/ 10 levels "0%","1-10%","11-23%",..: 2 1 1 5 1 3 1 3 2 3 ...
##  $ MBERZELF: Factor w/ 10 levels "0%","1-10%","11-23%",..: 1 1 1 1 6 1 1 1 2 1 ...
##  $ MBERBOER: Factor w/ 10 levels "0%","1-10%","11-23%",..: 2 1 1 1 5 1 1 1 1 1 ...
##  $ MBERMIDD: Factor w/ 10 levels "0%","1-10%","11-23%",..: 3 6 8 4 1 5 5 3 2 4 ...
##  $ MBERARBG: Factor w/ 10 levels "0%","1-10%","11-23%",..: 6 1 1 2 1 3 2 6 9 4 ...
##  $ MBERARBO: Factor w/ 10 levels "0%","1-10%","11-23%",..: 3 5 3 3 1 3 6 3 2 4 ...
##  $ MSKA    : Factor w/ 10 levels "0%","1-10%","11-23%",..: 2 1 1 4 10 3 1 3 2 2 ...
##  $ MSKB1   : Factor w/ 10 levels "0%","1-10%","11-23%",..: 2 3 6 3 1 3 2 2 2 3 ...
##  $ MSKB2   : Factor w/ 10 levels "0%","1-10%","11-23%",..: 3 4 1 2 1 3 5 3 1 2 ...
##  $ MSKC    : Factor w/ 10 levels "0%","1-10%","11-23%",..: 7 6 5 5 1 5 6 6 9 5 ...
##  $ MSKD    : Factor w/ 10 levels "0%","1-10%","11-23%",..: 2 1 1 1 1 3 1 3 2 3 ...
##  $ MHHUUR  : Factor w/ 10 levels "0%","1-10%","11-23%",..: 2 3 8 6 5 10 7 1 10 1 ...
##  $ MHKOOP  : Factor w/ 10 levels "0%","1-10%","11-23%",..: 9 8 3 5 6 1 4 10 1 10 ...
##  $ MAUT1   : Factor w/ 10 levels "0%","1-10%","11-23%",..: 9 8 8 10 7 6 9 5 6 7 ...
##  $ MAUT2   : Factor w/ 10 levels "0%","1-10%","11-23%",..: 1 2 1 1 3 4 1 5 3 2 ...
##  $ MAUT0   : Factor w/ 10 levels "0%","1-10%","11-23%",..: 2 3 3 1 2 4 2 3 4 3 ...
##  $ MZFONDS : Factor w/ 10 levels "0%","1-10%","11-23%",..: 9 7 10 8 6 10 10 7 8 7 ...
##  $ MZPART  : Factor w/ 10 levels "0%","1-10%","11-23%",..: 2 4 1 3 5 1 1 4 3 4 ...
##  $ MINKM30 : Factor w/ 10 levels "0%","1-10%","11-23%",..: 1 3 5 2 1 6 5 3 8 3 ...
##  $ MINK3045: Factor w/ 10 levels "0%","1-10%","11-23%",..: 5 1 6 6 1 3 4 6 3 4 ...
##  $ MINK4575: Factor w/ 10 levels "0%","1-10%","11-23%",..: 6 6 1 4 10 4 4 4 2 4 ...
##  $ MINK7512: Factor w/ 10 levels "0%","1-10%","11-23%",..: 1 3 1 1 1 1 1 1 1 2 ...
##  $ MINK123M: Factor w/ 10 levels "0%","1-10%","11-23%",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ MINKGEM : Factor w/ 10 levels "0%","1-10%","11-23%",..: 5 6 4 5 7 4 4 4 3 5 ...
##  $ MKOOPKLA: Factor w/ 10 levels "0%","1-10%","11-23%",..: 4 5 5 5 4 4 6 4 4 8 ...
##  $ PWAPART : Factor w/ 10 levels "0","1-49","50-99",..: 1 3 3 1 1 1 1 1 1 3 ...
##  $ PWABEDR : Factor w/ 10 levels "0","1-49","50-99",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ PWALAND : Factor w/ 10 levels "0","1-49","50-99",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ PPERSAUT: Factor w/ 10 levels "0","1-49","50-99",..: 7 1 7 7 1 7 7 1 6 1 ...
##  $ PBESAUT : Factor w/ 10 levels "0","1-49","50-99",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ PMOTSCO : Factor w/ 10 levels "0","1-49","50-99",..: 1 1 1 1 1 1 1 1 1 1 ...
```

```
##  $ PVRAAUT : Factor w/ 10 levels "0","1-49","50-99",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ PAANHANG: Factor w/ 10 levels "0","1-49","50-99",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ PTRACTOR: Factor w/ 10 levels "0","1-49","50-99",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ PWERKT  : Factor w/ 10 levels "0","1-49","50-99",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ PBROM   : Factor w/ 10 levels "0","1-49","50-99",..: 1 1 1 1 1 1 1 4 1 1 ...
##  $ PLEVEN  : Factor w/ 10 levels "0","1-49","50-99",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ PPERSONG: Factor w/ 10 levels "0","1-49","50-99",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ PGEZONG : Factor w/ 10 levels "0","1-49","50-99",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ PWAOREG : Factor w/ 10 levels "0","1-49","50-99",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ PBRAND  : Factor w/ 10 levels "0","1-49","50-99",..: 6 3 3 3 7 1 1 1 1 4 ...
##  $ PZEILPL : Factor w/ 10 levels "0","1-49","50-99",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ PPLEZIER: Factor w/ 10 levels "0","1-49","50-99",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ PFIETS  : Factor w/ 10 levels "0","1-49","50-99",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ PINBOED : Factor w/ 10 levels "0","1-49","50-99",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ PBYSTAND: Factor w/ 10 levels "0","1-49","50-99",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ AWAPART : Factor w/ 10 levels "0","1-49","50-99",..: 1 3 2 1 1 1 1 1 1 2 ...
##  $ AWABEDR : Factor w/ 10 levels "0","1-49","50-99",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ AWALAND : Factor w/ 10 levels "0","1-49","50-99",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ APERSAUT: Factor w/ 10 levels "0","1-49","50-99",..: 2 1 2 2 1 2 2 1 2 1 ...
##  $ ABESAUT : Factor w/ 10 levels "0","1-49","50-99",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ AMOTSCO : Factor w/ 10 levels "0","1-49","50-99",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ AVRAAUT : Factor w/ 10 levels "0","1-49","50-99",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ AAANHANG: Factor w/ 10 levels "0","1-49","50-99",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ ATRACTOR: Factor w/ 10 levels "0","1-49","50-99",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ AWERKT  : Factor w/ 10 levels "0","1-49","50-99",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ ABROM   : Factor w/ 10 levels "0","1-49","50-99",..: 1 1 1 1 1 1 1 2 1 1 ...
##  $ ALEVEN  : Factor w/ 10 levels "0","1-49","50-99",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ APERSONG: Factor w/ 10 levels "0","1-49","50-99",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ AGEZONG : Factor w/ 10 levels "0","1-49","50-99",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ AWAOREG : Factor w/ 10 levels "0","1-49","50-99",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ ABRAND  : Factor w/ 10 levels "0","1-49","50-99",..: 2 2 2 2 2 1 1 1 1 2 ...
##  $ AZEILPL : Factor w/ 10 levels "0","1-49","50-99",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ APLEZIER: Factor w/ 10 levels "0","1-49","50-99",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ AFIETS  : Factor w/ 10 levels "0","1-49","50-99",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ AINBOED : Factor w/ 10 levels "0","1-49","50-99",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ ABYSTAND: Factor w/ 10 levels "0","1-49","50-99",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ CARAVAN : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
```

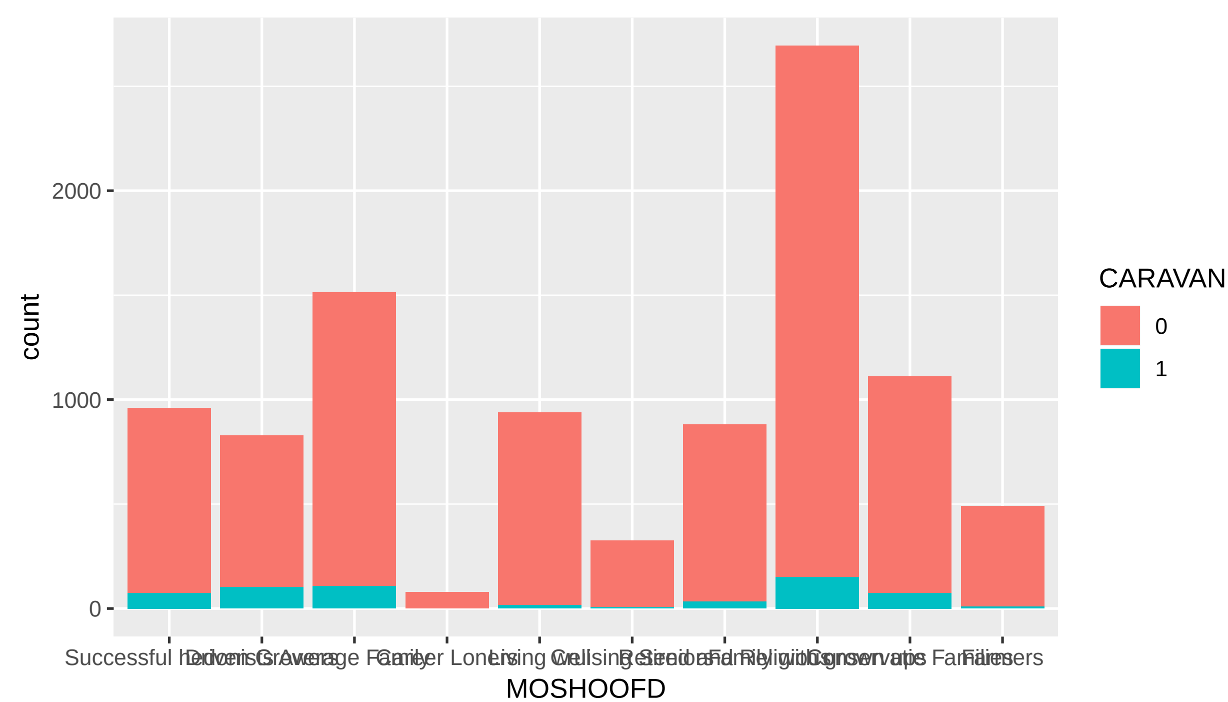
```r
#RESPONSE VARIABLE
ggplot(caravan_kaggle,aes(x=CARAVAN)) + geom_bar() + labs(x="No. of mobile home policies ")
```



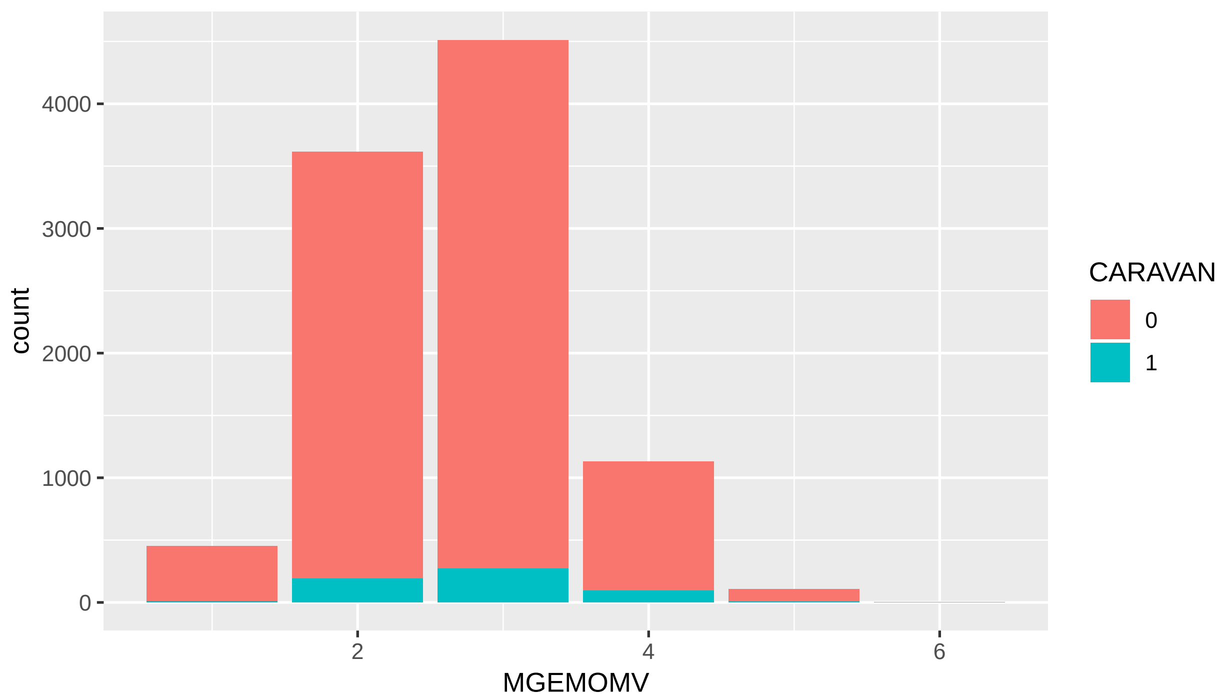

```
#There is about a 80/20 split in response variable i.e. approx. 20% of the data population has a mobile

#to determine which variables should be considered in our model, we plot each variable and see if there

# Var 44 (pr_num) is ignored for this analysis as it is an accounting or idenitification variable, and

#Analyze main customer type
plot<-ggplot(caravan_kaggle,aes(x=MOSTYPE, fill= CARAVAN))
plot<-plot + geom_bar()
plot<-plot + labs(x="Customer Main Type")
plot
```




```
#There is reasonable variation across customer types; this varaible should be left as is

#Analyze customer subtype
plot<-ggplot(caravan_kaggle,aes(x=MOSHOOFD, fill= CARAVAN))
plot<-plot + geom_bar()
plot
```

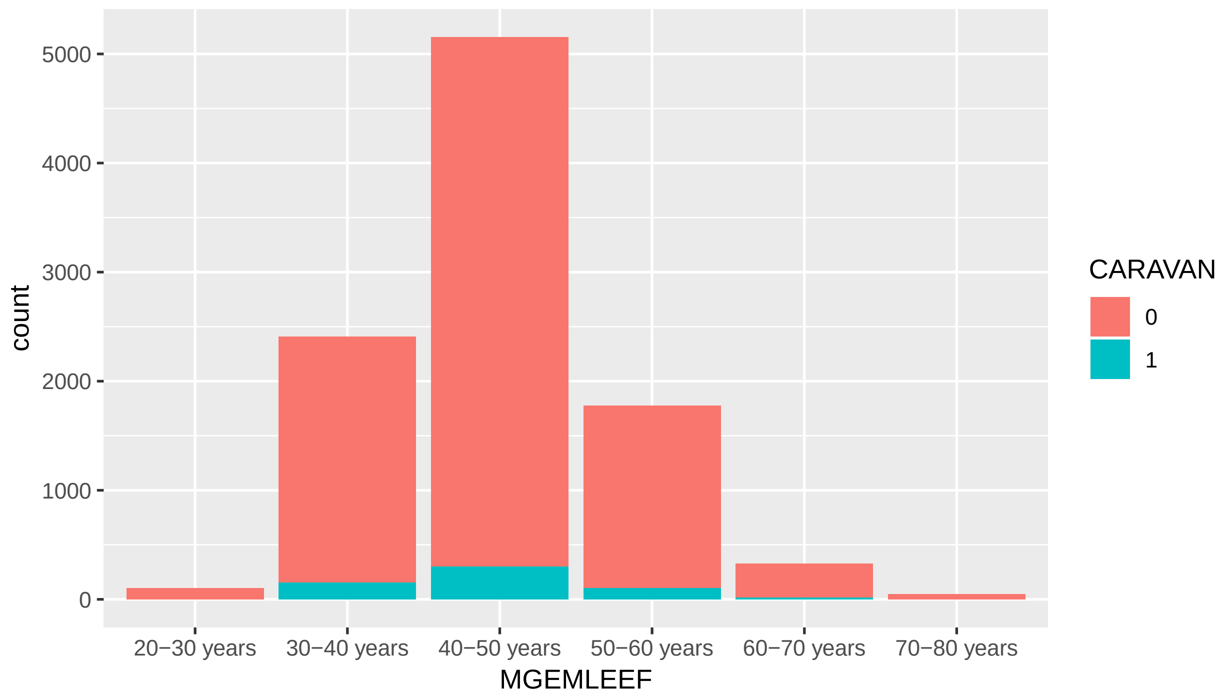


```
#There is reasonable variation across customer subtypes; all levels are represented. This varaible shou


#Analyzing var 4- avg size household
plot<-ggplot(caravan_kaggle,aes(x=MGEMOMV, fill= CARAVAN))
plot<-plot + geom_bar()
plot
```



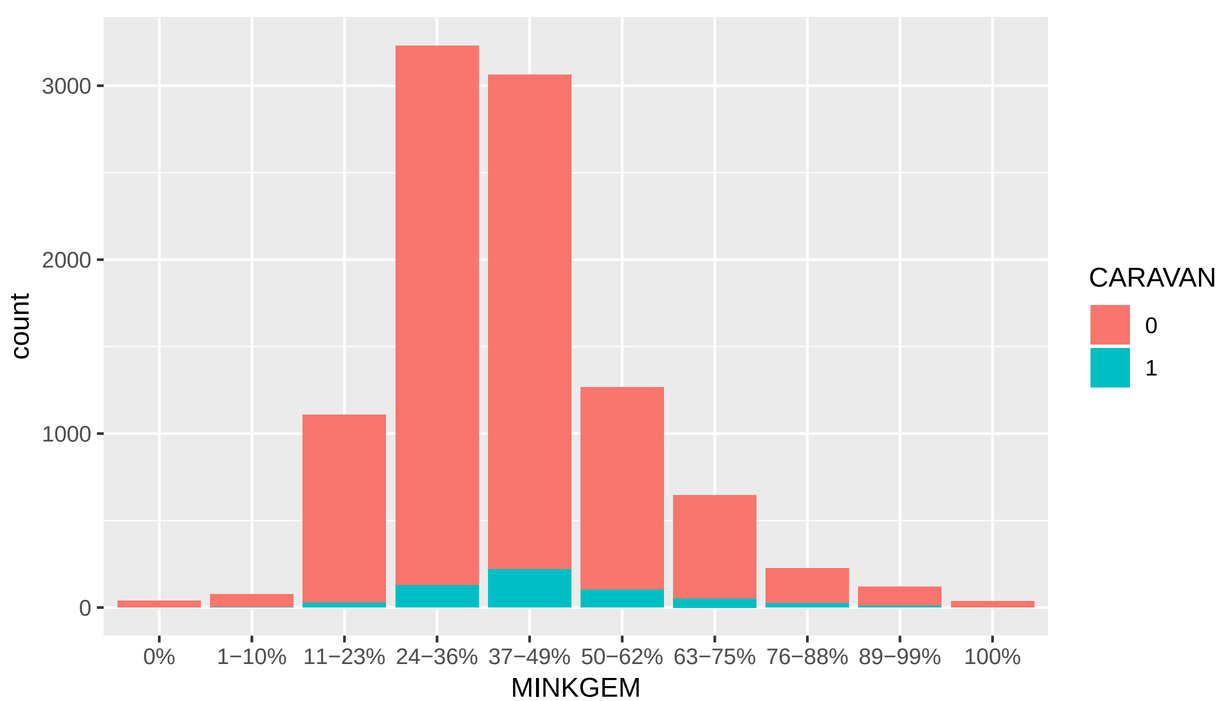

```
#Data is normal and has significant variation, so leave the variable as is
```

```
#Plot age data
plot<-ggplot(caravan_kaggle,aes(x=MGEMLEEF, fill= CARAVAN))
plot<-plot + geom_bar()
plot
```

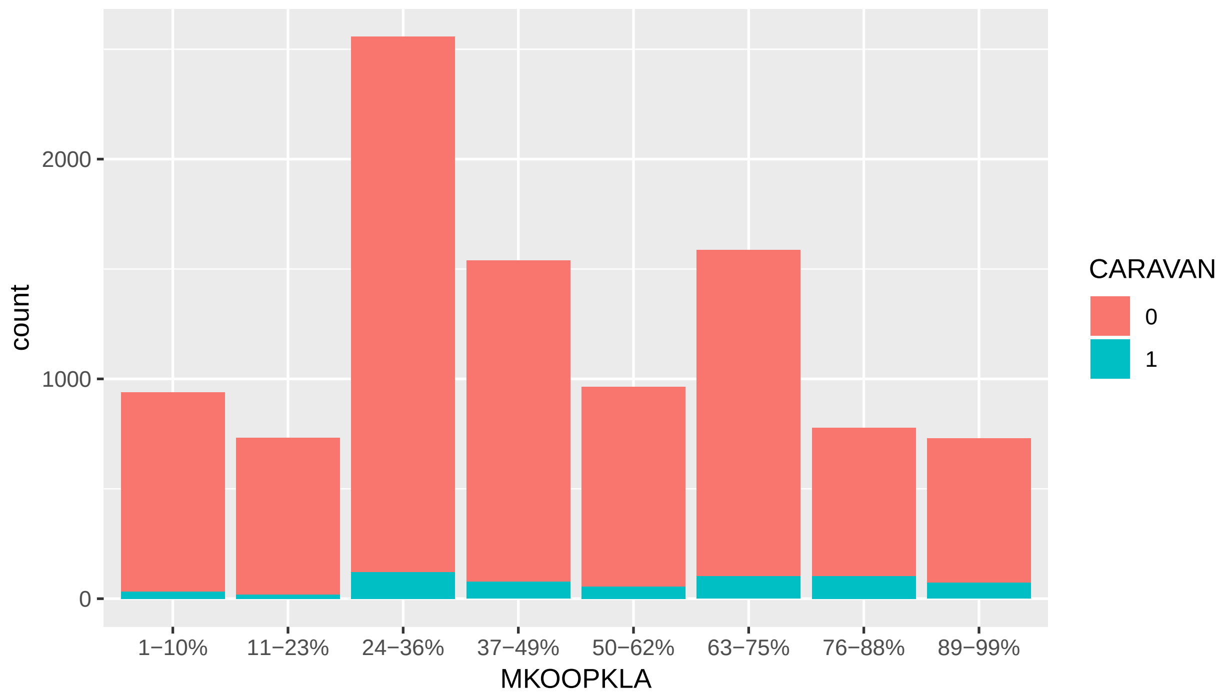



```
#Data is normal and has approximately normal distribution; we can move on
```

```
#Plot income
plot<-ggplot(caravan_kaggle,aes(x=MINKGEM, fill= CARAVAN))
plot<-plot + geom_bar()
plot
```



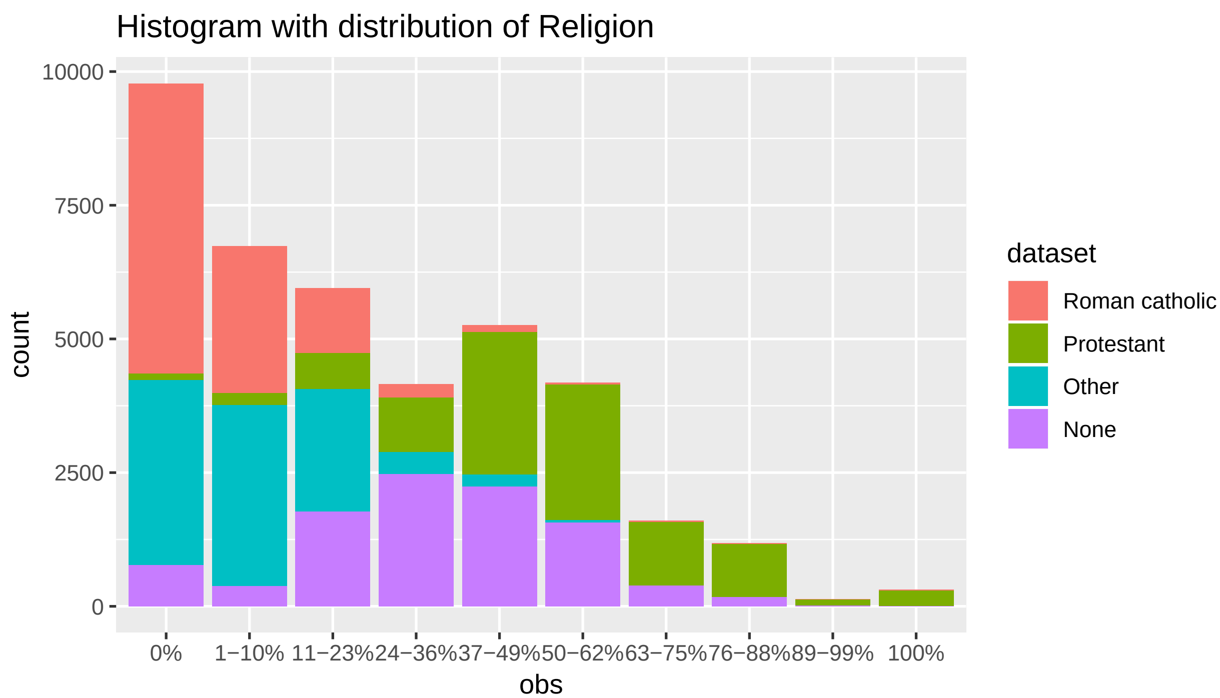
```
#Data is normal and has approximately normal distribution; we notice that, at first glance, it appears

#Plot purchasing power
plot<-ggplot(caravan_kaggle,aes(x=MKOOPKLA, fill= CARAVAN))
plot<-plot + geom_bar()
plot
```



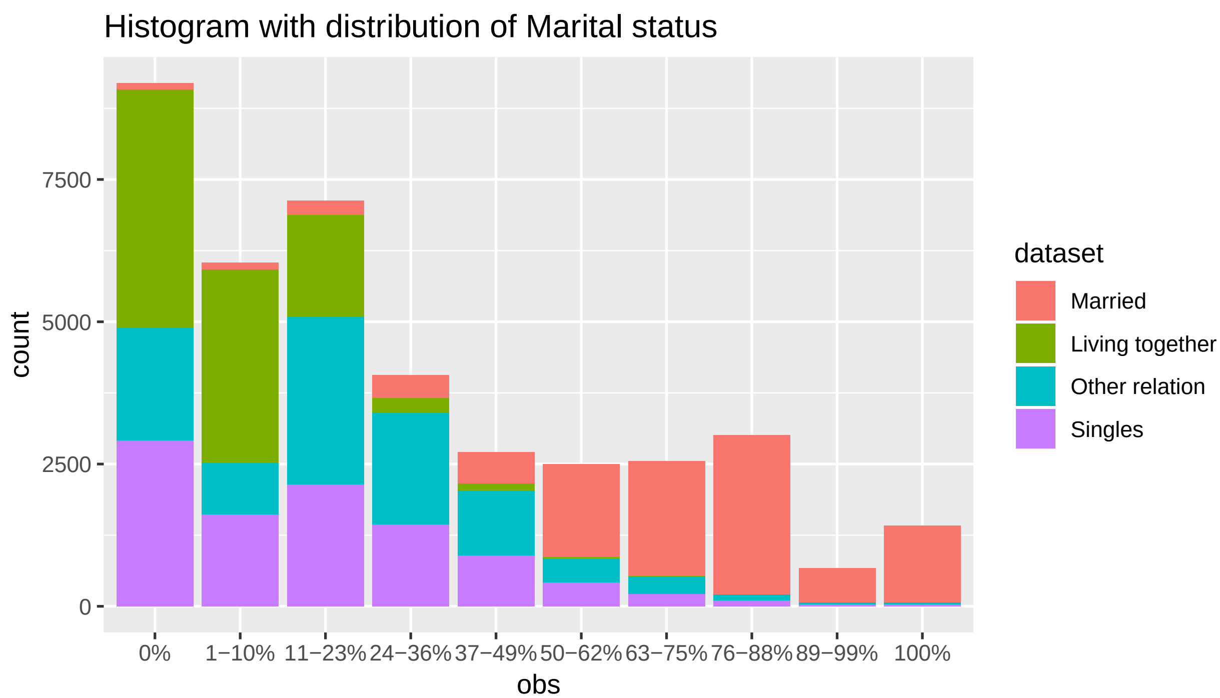

```
#No concerns with the distribution
```

```
#Certain demographic and behavioral factors are another great place to explore.
#Among the demographic factors, we thought religion, marital status, level of education, occupation, an
#Among the behavioral factors, variables such as contribution/spend and a number of other insurance var
```

```
#Varaibles 6-9 are all linked to religion, let us interpret them together
JUST.FOR.PLOT <- rbind(data.frame(dataset="Roman catholic", obs=caravan_kaggle$MGODRK),
            data.frame(dataset="Protestant", obs=caravan_kaggle$MGODPR),
            data.frame(dataset="Other ", obs=caravan_kaggle$MGODOV),
            data.frame(dataset="None", obs=caravan_kaggle$MGODGE))
JUST.FOR.PLOT$dataset <- as.factor(JUST.FOR.PLOT$dataset)
ggplot(JUST.FOR.PLOT, aes(x=obs, fill=dataset)) +geom_bar() +
ggtitle("Histogram with distribution of Religion")
```

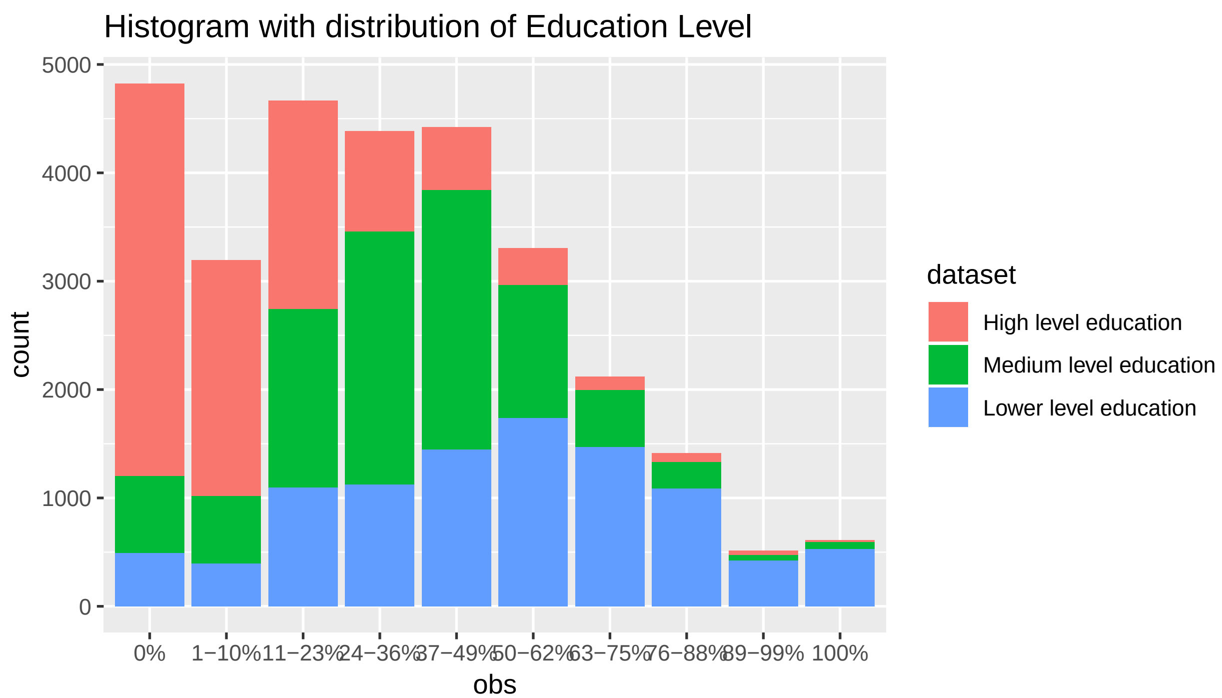



```
#We can see there is significant variation between each type of religion, and therefore these varaibles

#Variables 10-13 are all linked to Marital status, let us interpret them together
JUST.FOR.PLOT <- rbind(data.frame(dataset="Married", obs=caravan_kaggle$MRELGE),
            data.frame(dataset="Living together", obs=caravan_kaggle$MRELSA),
            data.frame(dataset="Other relation ", obs=caravan_kaggle$MRELOV),
            data.frame(dataset="Singles", obs=caravan_kaggle$MFALLEEN))
JUST.FOR.PLOT$dataset <- as.factor(JUST.FOR.PLOT$dataset)
ggplot(JUST.FOR.PLOT, aes(x=obs, fill=dataset)) +geom_bar() +
ggtitle("Histogram with distribution of Marital status")
```

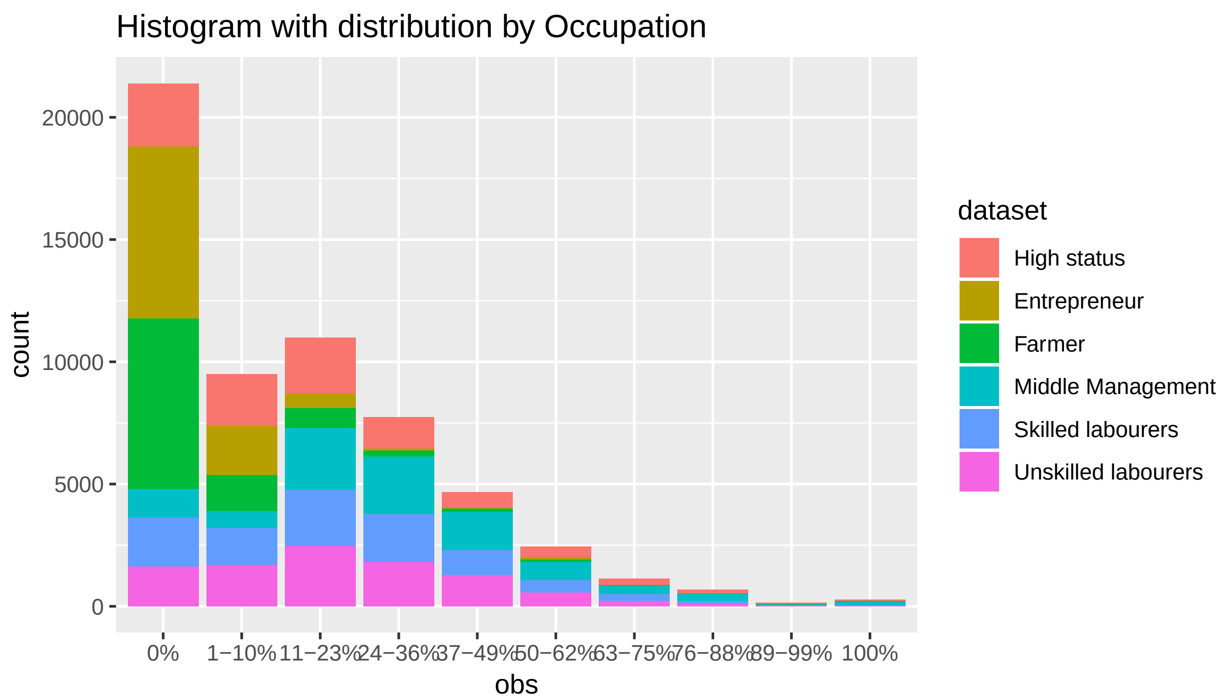



```
#We can see there is significant variation between each type of marital status, and therefore these var
```

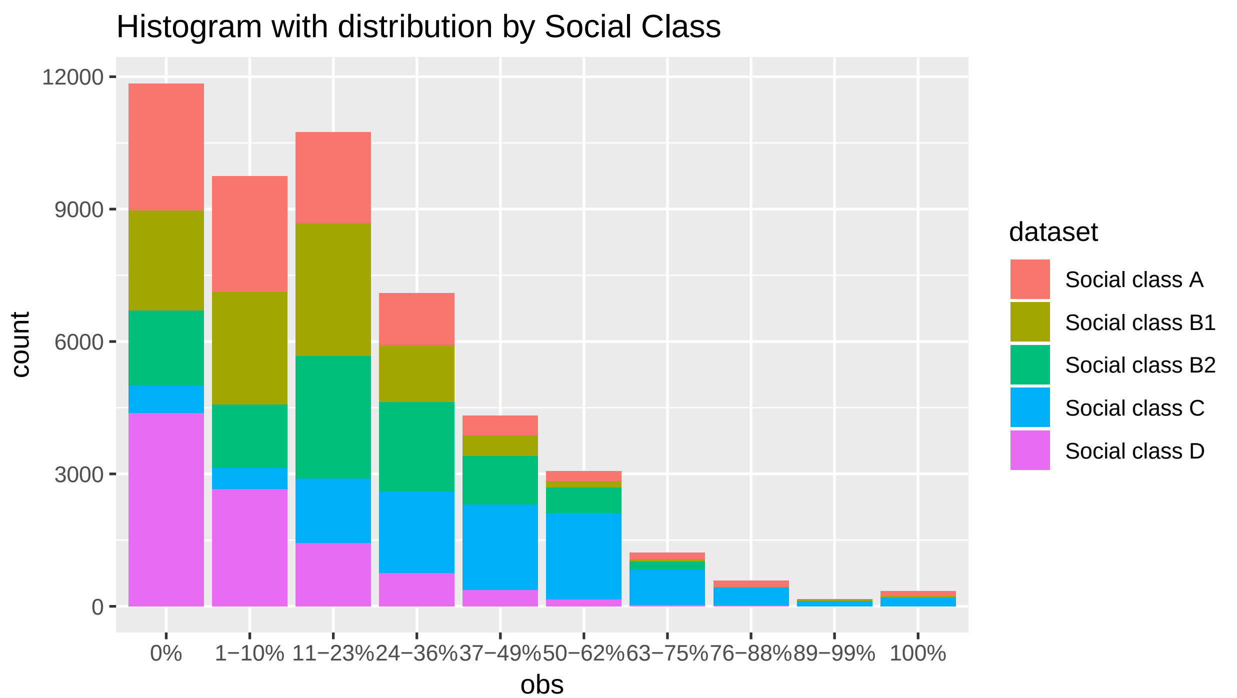
```
#histogram by education
JUST.FOR.PLOT <- rbind(data.frame(dataset="High level education", obs=caravan_kaggle$MOPLHOOG),
            data.frame(dataset="Medium level education", obs=caravan_kaggle$MOPLMIDD),
            data.frame(dataset="Lower level education", obs=caravan_kaggle$MOPLLAAG)
            )
JUST.FOR.PLOT$dataset <- as.factor(JUST.FOR.PLOT$dataset)
ggplot(JUST.FOR.PLOT, aes(x=obs, fill=dataset)) +geom_bar() +
ggtitle("Histogram with distribution of Education Level")
```



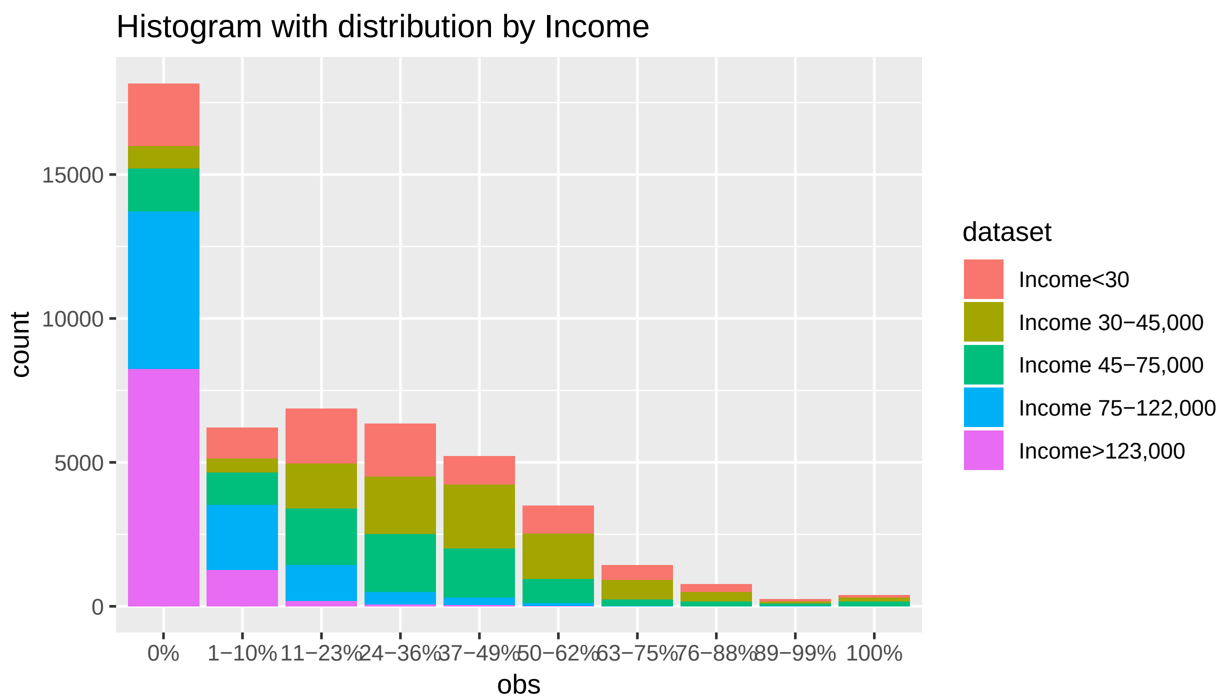

```
#histogram by occupation
JUST.FOR.PLOT <- rbind(data.frame(dataset="High status", obs=caravan_kaggle$MBERHOOG),
            data.frame(dataset="Entrepreneur", obs=caravan_kaggle$MBERZELF),
            data.frame(dataset="Farmer", obs=caravan_kaggle$MBERBOER),
            data.frame(dataset="Middle Management", obs=caravan_kaggle$MBERMIDD),
            data.frame(dataset="Skilled labourers", obs=caravan_kaggle$MBERARBG),
            data.frame(dataset="Unskilled labourers", obs=caravan_kaggle$MBERARBO)
            )
JUST.FOR.PLOT$dataset <- as.factor(JUST.FOR.PLOT$dataset)
ggplot(JUST.FOR.PLOT, aes(x=obs, fill=dataset)) +geom_bar() +
ggtitle("Histogram with distribution by Occupation")
```




```
#histogram by social class
JUST.FOR.PLOT <- rbind(data.frame(dataset="Social class A", obs=caravan_kaggle$MSKA),
            data.frame(dataset="Social class B1", obs=caravan_kaggle$MSKB1),
            data.frame(dataset="Social class B2", obs=caravan_kaggle$MSKB2),
            data.frame(dataset="Social class C", obs=caravan_kaggle$MSKC),
            data.frame(dataset="Social class D", obs=caravan_kaggle$MSKD)
            )
JUST.FOR.PLOT$dataset <- as.factor(JUST.FOR.PLOT$dataset)
ggplot(JUST.FOR.PLOT, aes(x=obs, fill=dataset)) +geom_bar() +
ggtitle("Histogram with distribution by Social Class")
```




```r
#histogram by Income
JUST.FOR.PLOT <- rbind(data.frame(dataset="Income<30", obs=caravan_kaggle$MINKM30),
            data.frame(dataset="Income 30-45,000", obs=caravan_kaggle$MINK3045),
            data.frame(dataset="Income 45-75,000", obs=caravan_kaggle$MINK4575),
            data.frame(dataset="Income 75-122,000", obs=caravan_kaggle$MINK7512),
            data.frame(dataset="Income>123,000", obs=caravan_kaggle$MINK123M)
            )
JUST.FOR.PLOT$dataset <- as.factor(JUST.FOR.PLOT$dataset)
ggplot(JUST.FOR.PLOT, aes(x=obs, fill=dataset)) +geom_bar() +
ggtitle("Histogram with distribution by Income")
```



Logistical models

```
caravan.train <- caravan_kaggle_2[caravan_kaggle_2$ORIGIN %in% "train",]
caravan.train <- caravan.train[-1] #delete "ORIGIN" column
caravan.test <- caravan_kaggle_2[caravan_kaggle_2$ORIGIN %in% "test",]
caravan.test <- caravan.test[-1] #delete "ORIGIN" column

# Create full logistic regression model
fit.logit.0 <- glm(CARAVAN~., family=binomial, data=caravan.train)
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(fit.logit.0)
```

```
##
## Call:
## glm(formula = CARAVAN ~ ., family = binomial, data = caravan.train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7047  -0.3711  -0.2450  -0.1588   3.2916
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  2.542e+02  1.116e+04   0.023  0.98183
## MOSTYPE      6.580e-02  4.624e-02   1.423  0.15468
## MAANTHUI    -1.832e-01  1.927e-01  -0.951  0.34157
## MGEMOMV     -2.696e-02  1.399e-01  -0.193  0.84723
## MGEMLEEF     2.096e-01  1.016e-01   2.063  0.03911 *
## MOSHOOFD    -2.767e-01  2.076e-01  -1.333  0.18247
## MGODRK      -1.142e-01  1.069e-01  -1.068  0.28535
```

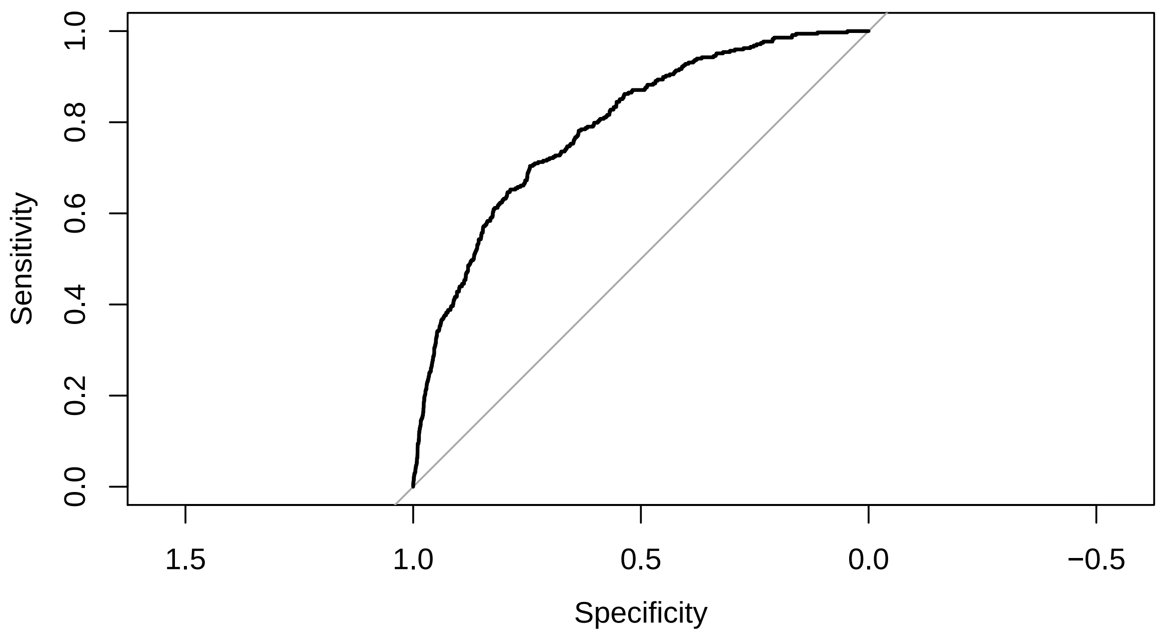
```
## MGODPR      -1.910e-02  1.177e-01  -0.162  0.87112
## MGODOV      -1.618e-02  1.055e-01  -0.153  0.87818
## MGODGE      -6.817e-02  1.113e-01  -0.612  0.54024
## MRELGE       2.310e-01  1.566e-01   1.475  0.14031
## MRELSA       8.509e-02  1.466e-01   0.580  0.56169
## MRELOV       1.467e-01  1.562e-01   0.939  0.34759
## MFALLEEN    -8.291e-02  1.311e-01  -0.633  0.52702
## MFGEKIND    -1.154e-01  1.337e-01  -0.863  0.38813
## MFWEKIND    -8.140e-02  1.417e-01  -0.575  0.56561
## MOPLHOOG     9.717e-04  1.311e-01   0.007  0.99408
## MOPLMIDD    -9.077e-02  1.365e-01  -0.665  0.50605
## MOPLLAAG    -1.994e-01  1.376e-01  -1.449  0.14740
## MBERHOOG     8.883e-02  9.349e-02   0.950  0.34204
## MBERZELF     3.918e-02  9.897e-02   0.396  0.69219
## MBERBOER    -1.169e-01  1.104e-01  -1.059  0.28951
## MBERMIDD     1.353e-01  9.191e-02   1.472  0.14106
## MBERARBG     3.976e-02  9.067e-02   0.438  0.66104
## MBERARBO     9.954e-02  9.143e-02   1.089  0.27628
## MSKA         2.690e-02  1.035e-01   0.260  0.79502
## MSKB1       -8.801e-03  1.011e-01  -0.087  0.93064
## MSKB2        1.200e-02  9.081e-02   0.132  0.89485
## MSKC         9.016e-02  9.958e-02   0.905  0.36527
## MSKD        -2.468e-02  9.724e-02  -0.254  0.79967
## MHHUUR      -1.472e+01  8.140e+02  -0.018  0.98557
## MHKOOP      -1.469e+01  8.140e+02  -0.018  0.98561
## MAUT1        1.819e-01  1.514e-01   1.202  0.22953
## MAUT2        1.507e-01  1.371e-01   1.099  0.27162
## MAUT0        9.325e-02  1.436e-01   0.649  0.51603
## MZFONDS     -1.445e+01  9.359e+02  -0.015  0.98768
## MZPART      -1.451e+01  9.359e+02  -0.016  0.98763
## MINKM30      1.181e-01  1.006e-01   1.174  0.24039
## MINK3045     1.366e-01  9.650e-02   1.415  0.15694
## MINK4575     1.009e-01  9.667e-02   1.043  0.29678
## MINK7512     1.144e-01  1.027e-01   1.114  0.26513
## MINK123M    -1.607e-01  1.449e-01  -1.109  0.26738
## MINKGEM      9.214e-02  9.945e-02   0.927  0.35417
## MKOOPKLA     6.856e-02  4.642e-02   1.477  0.13966
## PWAPART      5.954e-01  3.901e-01   1.526  0.12693
## PWABEDR     -2.757e-01  4.635e-01  -0.595  0.55196
## PWALAND     -4.405e-01  1.035e+00  -0.425  0.67052
## PPERSAUT     2.306e-01  4.199e-02   5.491 4.01e-08 ***
## PBESAUT      1.215e+01  4.029e+02   0.030  0.97595
## PMOTSCO     -8.101e-02  1.147e-01  -0.706  0.48006
## PVRAAUT     -2.106e+00  2.557e+03  -0.001  0.99934
## PAANHANG     1.014e+00  9.371e-01   1.082  0.27917
## PTRACTOR     7.229e-01  4.278e-01   1.690  0.09107 .
## PWERKT      -5.525e+00  4.805e+03  -0.001  0.99908
## PBROM        2.170e-01  4.865e-01   0.446  0.65559
## PLEVEN      -2.382e-01  1.170e-01  -2.036  0.04173 *
## PPERSONG    -4.523e-01  2.094e+00  -0.216  0.82901
## PGEZONG      1.444e+00  1.029e+00   1.404  0.16033
## PWAOREG      8.239e-01  5.943e-01   1.386  0.16565
## PBRAND       2.401e-01  7.714e-02   3.113  0.00185 **
## PZEILPL     -8.658e+00  3.261e+03  -0.003  0.99788
```

```
## PPLEZIER    -1.886e-01  3.259e-01  -0.579  0.56289
## PFIETS       3.664e-01  8.325e-01   0.440  0.65985
## PINBOED     -1.068e+00  8.764e-01  -1.219  0.22301
## PBYSTAND    -1.676e-01  3.321e-01  -0.505  0.61373
## AWAPART     -9.293e-01  7.802e-01  -1.191  0.23364
## AWABEDR      4.197e-01  1.082e+00   0.388  0.69824
## AWALAND      2.762e-01  3.528e+00   0.078  0.93758
## APERSAUT    -3.902e-02  1.772e-01  -0.220  0.82566
## ABESAUT     -7.298e+01  2.417e+03  -0.030  0.97591
## AMOTSCO      2.418e-01  3.772e-01   0.641  0.52142
## AVRAAUT     -4.490e+00  1.078e+04   0.000  0.99967
## AAANHANG    -1.351e+00  1.687e+00  -0.801  0.42322
## ATRACTOR    -2.376e+00  1.524e+00  -1.559  0.11899
## AWERKT      -8.749e-01  9.682e+03   0.000  0.99993
## ABROM       -1.060e+00  1.549e+00  -0.684  0.49367
## ALEVEN       4.789e-01  2.245e-01   2.133  0.03291 *
## APERSONG     3.997e-01  4.329e+00   0.092  0.92644
## AGEZONG     -3.163e+00  2.706e+00  -1.169  0.24247
## AWAOREG     -3.212e+00  3.433e+00  -0.936  0.34939
## ABRAND      -4.118e-01  2.787e-01  -1.477  0.13956
## AZEILPL      1.047e+01  3.261e+03   0.003  0.99744
## APLEZIER     2.516e+00  1.010e+00   2.490  0.01276 *
## AFIETS       2.318e-01  5.699e-01   0.407  0.68420
## AINBOED      1.947e+00  1.412e+00   1.378  0.16812
## ABYSTAND     1.078e+00  1.103e+00   0.977  0.32870
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2635.5  on 5821  degrees of freedom
## Residual deviance: 2243.5  on 5736  degrees of freedom
## AIC: 2415.5
##
## Number of Fisher Scoring iterations: 17
```

```r
# Get ROC and AUC
prob=predict(fit.logit.0,type=c("response"))
caravan.train$prob=prob
library(pROC)
g <- roc(CARAVAN ~ prob, data = caravan.train)
g
```

```
##
## Call:
## roc.formula(formula = CARAVAN ~ prob, data = caravan.train)
##
## Data: prob in 5474 controls (CARAVAN 0) < 348 cases (CARAVAN 1).
## Area under the curve: 0.7903
```

```r
plot(g)
```




```
# Incorporate loss of 0.2 since we are much more comfortable marketing to those who are less likely to p
fit.pred.2 <- rep("0", 5822)
fit.pred.2[fit.logit.0$fitted > .2] <- "1"

# Find MCE
MCE.2 <- (sum(5*(fit.pred.2[caravan.train$CARAVAN == "1"] != "1")) + sum(fit.pred.2[caravan.train$CARAVA
MCE.2
```

```
## [1] 0.2579869
```

Backward selection

```
# Logistic with backward selection
caravan.train <- caravan.train[-87] #delete "prob" column
fit.backward <- regsubsets(CARAVAN ~., caravan.train, nvmax=8, method="backward")
f.b <- summary(fit.backward)
f.b
```

```
## Subset selection object
## Call: regsubsets.formula(CARAVAN ~ ., caravan.train, nvmax = 8, method = "backward")
## 85 Variables  (and intercept)
##          Forced in Forced out
## MOSTYPE      FALSE      FALSE
## MAANTHUI     FALSE      FALSE
## MGEMOMV      FALSE      FALSE
## MGEMLEEF     FALSE      FALSE
## MOSHOOFD     FALSE      FALSE
## MGODRK       FALSE      FALSE
## MGODPR       FALSE      FALSE
## MGODOV       FALSE      FALSE
## MGODGE       FALSE      FALSE
## MRELGE       FALSE      FALSE
```

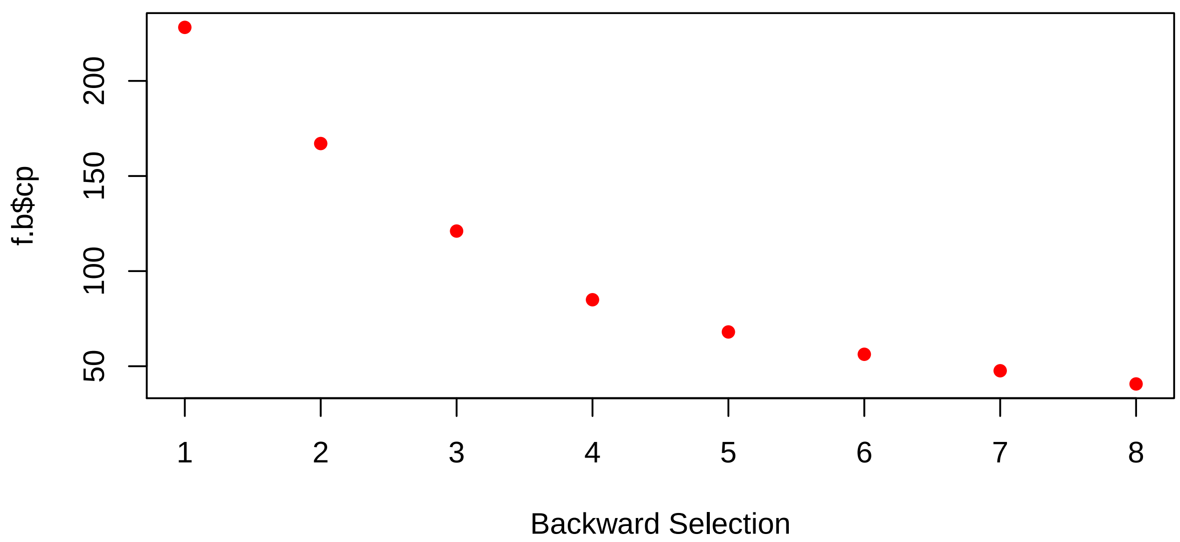
```
## MRELSA       FALSE      FALSE
## MRELOV       FALSE      FALSE
## MFALLEEN     FALSE      FALSE
## MFGEKIND     FALSE      FALSE
## MFWEKIND     FALSE      FALSE
## MOPLHOOG     FALSE      FALSE
## MOPLMIDD     FALSE      FALSE
## MOPLLAAG     FALSE      FALSE
## MBERHOOG     FALSE      FALSE
## MBERZELF     FALSE      FALSE
## MBERBOER     FALSE      FALSE
## MBERMIDD     FALSE      FALSE
## MBERARBG     FALSE      FALSE
## MBERARBO     FALSE      FALSE
## MSKA         FALSE      FALSE
## MSKB1        FALSE      FALSE
## MSKB2        FALSE      FALSE
## MSKC         FALSE      FALSE
## MSKD         FALSE      FALSE
## MHHUUR       FALSE      FALSE
## MHKOOP       FALSE      FALSE
## MAUT1        FALSE      FALSE
## MAUT2        FALSE      FALSE
## MAUT0        FALSE      FALSE
## MZFONDS      FALSE      FALSE
## MZPART       FALSE      FALSE
## MINKM30      FALSE      FALSE
## MINK3045     FALSE      FALSE
## MINK4575     FALSE      FALSE
## MINK7512     FALSE      FALSE
## MINK123M     FALSE      FALSE
## MINKGEM      FALSE      FALSE
## MKOOPKLA     FALSE      FALSE
## PWAPART      FALSE      FALSE
## PWABEDR      FALSE      FALSE
## PWALAND      FALSE      FALSE
## PPERSAUT     FALSE      FALSE
## PBESAUT      FALSE      FALSE
## PMOTSCO      FALSE      FALSE
## PVRAAUT      FALSE      FALSE
## PAANHANG     FALSE      FALSE
## PTRACTOR     FALSE      FALSE
## PWERKT       FALSE      FALSE
## PBROM        FALSE      FALSE
## PLEVEN       FALSE      FALSE
## PPERSONG     FALSE      FALSE
## PGEZONG      FALSE      FALSE
## PWAOREG      FALSE      FALSE
## PBRAND       FALSE      FALSE
## PZEILPL      FALSE      FALSE
## PPLEZIER     FALSE      FALSE
## PFIETS       FALSE      FALSE
## PINBOED      FALSE      FALSE
## PBYSTAND     FALSE      FALSE
```

```
## AWAPART      FALSE      FALSE
## AWABEDR      FALSE      FALSE
## AWALAND      FALSE      FALSE
## APERSAUT     FALSE      FALSE
## ABESAUT      FALSE      FALSE
## AMOTSCO      FALSE      FALSE
## AVRAAUT      FALSE      FALSE
## AAANHANG     FALSE      FALSE
## ATRACTOR     FALSE      FALSE
## AWERKT       FALSE      FALSE
## ABROM        FALSE      FALSE
## ALEVEN       FALSE      FALSE
## APERSONG     FALSE      FALSE
## AGEZONG      FALSE      FALSE
## AWAOREG      FALSE      FALSE
## ABRAND       FALSE      FALSE
## AZEILPL      FALSE      FALSE
## APLEZIER     FALSE      FALSE
## AFIETS       FALSE      FALSE
## AINBOED      FALSE      FALSE
## ABYSTAND     FALSE      FALSE
## 1 subsets of each size up to 8
## Selection Algorithm: backward
##          MOSTYPE MAANTHUI MGEMOMV MGEMLEEF MOSHOOFD MGODRK MGODPR MGODOV
## 1  ( 1 ) " "     " "      " "     " "      " "      " "    " "    " "
## 2  ( 1 ) " "     " "      " "     " "      " "      " "    " "    " "
## 3  ( 1 ) " "     " "      " "     " "      " "      " "    " "    " "
## 4  ( 1 ) " "     " "      " "     " "      " "      " "    " "    " "
## 5  ( 1 ) " "     " "      " "     " "      " "      " "    " "    " "
## 6  ( 1 ) " "     " "      " "     " "      " "      " "    " "    " "
## 7  ( 1 ) " "     " "      " "     " "      " "      " "    " "    " "
## 8  ( 1 ) " "     " "      " "     " "      " "      " "    " "    " "
##          MGODGE MRELGE MRELSA MRELOV MFALLEEN MFGEKIND MFWEKIND MOPLHOOG
## 1  ( 1 ) " "    " "    " "    " "    " "      " "      " "      " "
## 2  ( 1 ) " "    " "    " "    " "    " "      " "      " "      " "
## 3  ( 1 ) " "    " "    " "    " "    " "      " "      " "      " "
## 4  ( 1 ) " "    " "    " "    " "    " "      " "      " "      " "
## 5  ( 1 ) " "    "*"    " "    " "    " "      " "      " "      " "
## 6  ( 1 ) " "    "*"    " "    " "    " "      " "      " "      " "
## 7  ( 1 ) " "    "*"    " "    " "    " "      " "      " "      " "
## 8  ( 1 ) " "    "*"    " "    " "    " "      " "      " "      " "
##          MOPLMIDD MOPLLAAG MBERHOOG MBERZELF MBERBOER MBERMIDD MBERARBG
## 1  ( 1 ) " "      " "      " "      " "      " "      " "      " "
## 2  ( 1 ) " "      " "      " "      " "      " "      " "      " "
## 3  ( 1 ) " "      "*"      " "      " "      " "      " "      " "
## 4  ( 1 ) " "      "*"      " "      " "      " "      " "      " "
## 5  ( 1 ) " "      "*"      " "      " "      " "      " "      " "
## 6  ( 1 ) " "      "*"      " "      " "      " "      " "      " "
## 7  ( 1 ) " "      "*"      " "      " "      " "      " "      " "
## 8  ( 1 ) " "      "*"      " "      " "      "*"      " "      " "
##          MBERARBO MSKA MSKB1 MSKB2 MSKC MSKD MHHUUR MHKOOP MAUT1 MAUT2
## 1  ( 1 ) " "      " "  " "   " "   " "  " "  " "    " "    " "   " "
## 2  ( 1 ) " "      " "  " "   " "   " "  " "  " "    " "    " "   " "
## 3  ( 1 ) " "      " "  " "   " "   " "  " "  " "    " "    " "   " "
```

```
## 4  ( 1 ) " "      " "  " "   " "   " "  " "  " "    " "    " "   " "
## 5  ( 1 ) " "      " "  " "   " "   " "  " "  " "    " "    " "   " "
## 6  ( 1 ) " "      " "  " "   " "   " "  " "  " "    " "    " "   " "
## 7  ( 1 ) " "      " "  " "   " "   " "  " "  " "    " "    " "   " "
## 8  ( 1 ) " "      " "  " "   " "   " "  " "  " "    " "    " "   " "
##          MAUTO MZFONDS MZPART MINKM30 MINK3045 MINK4575 MINK7512 MINK123M
## 1  ( 1 ) " "   " "     " "    " "     " "      " "      " "      " "
## 2  ( 1 ) " "   " "     " "    " "     " "      " "      " "      " "
## 3  ( 1 ) " "   " "     " "    " "     " "      " "      " "      " "
## 4  ( 1 ) " "   " "     " "    " "     " "      " "      " "      " "
## 5  ( 1 ) " "   " "     " "    " "     " "      " "      " "      " "
## 6  ( 1 ) " "   " "     " "    " "     " "      " "      " "      " "
## 7  ( 1 ) " "   " "     " "    " "     " "      " "      " "      " "
## 8  ( 1 ) " "   " "     " "    " "     " "      " "      " "      " "
##          MINKGEM MKOOPKLA PWAPART PWABEDR PWALAND PPERSAUT PBESAUT PMOTSCO
## 1  ( 1 ) " "     " "      " "     " "     " "     "*"      " "     " "
## 2  ( 1 ) " "     " "      " "     " "     " "     "*"      " "     " "
## 3  ( 1 ) " "     " "      " "     " "     " "     "*"      " "     " "
## 4  ( 1 ) " "     " "      " "     " "     " "     "*"      " "     " "
## 5  ( 1 ) " "     " "      " "     " "     " "     "*"      " "     " "
## 6  ( 1 ) " "     " "      " "     " "     "*"     "*"      " "     " "
## 7  ( 1 ) " "     " "      " "     " "     "*"     "*"      " "     " "
## 8  ( 1 ) " "     " "      " "     " "     "*"     "*"      " "     " "
##          PVRAAUT PAANHANG PTRACTOR PWERKT PBROM PLEVEN PPERSONG PGEZONG
## 1  ( 1 ) " "     " "      " "      " "    " "   " "    " "      " "
## 2  ( 1 ) " "     " "      " "      " "    " "   " "    " "      " "
## 3  ( 1 ) " "     " "      " "      " "    " "   " "    " "      " "
## 4  ( 1 ) " "     " "      " "      " "    " "   " "    " "      " "
## 5  ( 1 ) " "     " "      " "      " "    " "   " "    " "      " "
## 6  ( 1 ) " "     " "      " "      " "    " "   " "    " "      " "
## 7  ( 1 ) " "     " "      " "      " "    " "   " "    " "      " "
## 8  ( 1 ) " "     " "      " "      " "    " "   " "    " "      " "
##          PWAOREG PBRAND PZEILPL PPLEZIER PFIETS PINBOED PBYSTAND AWAPART
## 1  ( 1 ) " "     " "    " "     " "      " "    " "     " "      " "
## 2  ( 1 ) " "     " "    " "     " "      " "    " "     " "      " "
## 3  ( 1 ) " "     " "    " "     " "      " "    " "     " "      " "
## 4  ( 1 ) " "     "*"    " "     " "      " "    " "     " "      " "
## 5  ( 1 ) " "     "*"    " "     " "      " "    " "     " "      " "
## 6  ( 1 ) " "     "*"    " "     " "      " "    " "     " "      " "
## 7  ( 1 ) " "     "*"    " "     " "      " "    " "     " "      " "
## 8  ( 1 ) " "     "*"    " "     " "      " "    " "     " "      " "
##          AWABEDR AWALAND APERSAUT ABESAUT AMOTSCO AVRAAUT AAANHANG
## 1  ( 1 ) " "     " "     " "      " "     " "     " "     " "
## 2  ( 1 ) " "     " "     " "      " "     " "     " "     " "
## 3  ( 1 ) " "     " "     " "      " "     " "     " "     " "
## 4  ( 1 ) " "     " "     " "      " "     " "     " "     " "
## 5  ( 1 ) " "     " "     " "      " "     " "     " "     " "
## 6  ( 1 ) " "     " "     " "      " "     " "     " "     " "
## 7  ( 1 ) " "     " "     " "      " "     " "     " "     " "
## 8  ( 1 ) " "     " "     " "      " "     " "     " "     " "
##          ATRACTOR AWERKT ABROM ALEVEN APERSONG AGEZONG AWAOREG ABRAND
## 1  ( 1 ) " "      " "    " "   " "    " "      " "     " "     " "
## 2  ( 1 ) " "      " "    " "   " "    " "      " "     " "     " "
## 3  ( 1 ) " "      " "    " "   " "    " "      " "     " "     " "
```

```
## 4  ( 1 ) " "      " "    " "   " "    " "      " "     " "     " "
## 5  ( 1 ) " "      " "    " "   " "    " "      " "     " "     " "
## 6  ( 1 ) " "      " "    " "   " "    " "      " "     " "     " "
## 7  ( 1 ) " "      " "    " "   " "    " "      " "     " "     " "
## 8  ( 1 ) " "      " "    " "   " "    " "      " "     " "     " "
##          AZEILPL APLEZIER AFIETS AINBOED ABYSTAND
## 1  ( 1 ) " "     " "      " "    " "     " "
## 2  ( 1 ) " "     "*"      " "    " "     " "
## 3  ( 1 ) " "     "*"      " "    " "     " "
## 4  ( 1 ) " "     "*"      " "    " "     " "
## 5  ( 1 ) " "     "*"      " "    " "     " "
## 6  ( 1 ) " "     "*"      " "    " "     " "
## 7  ( 1 ) " "     "*"      " "    " "     "*"
## 8  ( 1 ) " "     "*"      " "    " "     "*"
```

```
plot(f.b$cp,  col="red", type="p", pch=16,
   xlab="Backward Selection")
```




```
coef(fit.backward, 8)
```

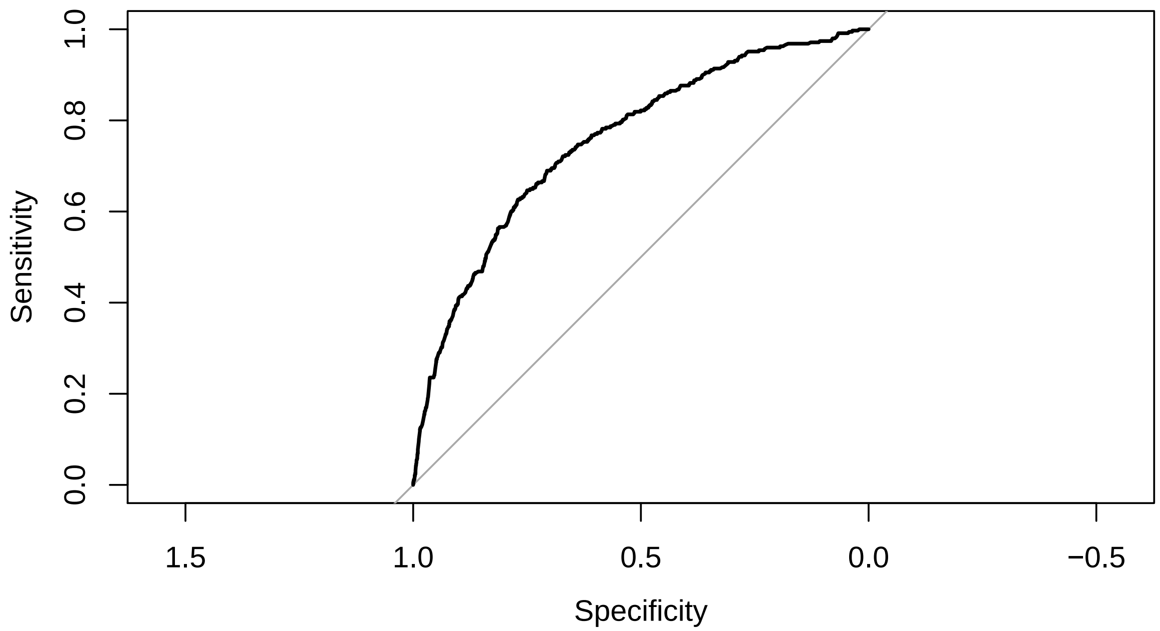
```
##   (Intercept)        MRELGE      MOPLLAAG      MBERBOER       PWALAND
##   0.001850234   0.006879012  -0.007523787  -0.008752079  -0.019827878
##      PPERSAUT        PBRAND      APLEZIER      ABYSTAND
##   0.011057523   0.010985109   0.283583028   0.080852868
```

```
# Fit glm model
fit.logit.1 <- glm(CARAVAN~MRELGE+MOPLLAAG+MBERBOER+PWALAND+PPERSAUT+PBRAND+APLEZIER+ABYSTAND, family=b

# Get ROC and AUC
prob=predict(fit.logit.1,type=c("response"))
caravan.train$prob=prob
g <- roc(CARAVAN ~ prob, data = caravan.train)
```

```
g
```

```
##
## Call:
## roc.formula(formula = CARAVAN ~ prob, data = caravan.train)
##
## Data: prob in 5474 controls (CARAVAN 0) < 348 cases (CARAVAN 1).
## Area under the curve: 0.7561
```

```
plot(g)
```

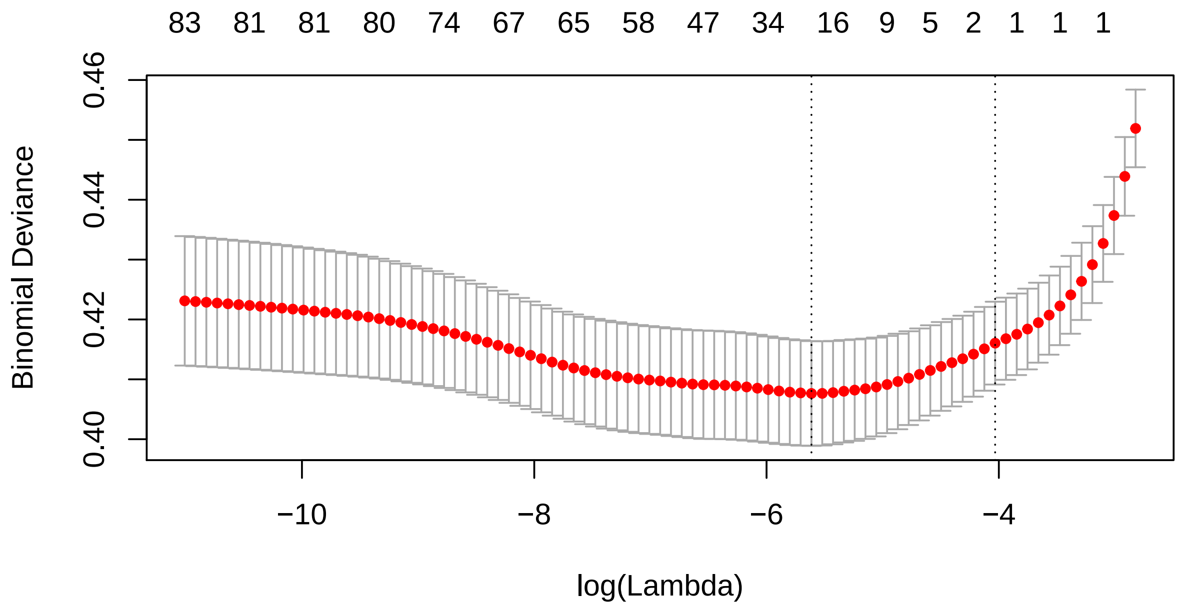



```
# Incorporate loss of 0.2 since we are much more comfortable marketing to those who are less likely to p
fit.pred.2 <- rep("0", 5822)
fit.pred.2[fit.logit.1$fitted > .2] <- "1"

# Find MCE
MCE.2 <- (sum(5*(fit.pred.2[caravan.train$CARAVAN == "1"] != "1")) + sum(fit.pred.2[caravan.train$CARAVA
MCE.2
```

```
## [1] 0.2748196
```

LASSO and Elastic Net

```
# LASSO technique and elastic net
# First, we prepare the design matrix and response
X <- model.matrix(CARAVAN~., caravan.train)[,-1]
Y <- caravan.train[, 86]

set.seed(10) # to have same sets of K folds
fit2.cv <- cv.glmnet(X, Y, alpha=1, family="binomial", nfolds = 10, type.measure = "deviance")
plot(fit2.cv)
```




```
coef.min <-coef(fit2.cv, s="lambda.min")
coef.min <- coef.min[which(coef.min !=0), ]
as.matrix(coef.min)
```

```
##                      [,1]
## (Intercept) -4.570501944
## MGEMLEEF     0.014142903
## MGODPR       0.018396259
## MOPLHOOG     0.034736834
## MBERBOER    -0.018094718
## MBERMIDD     0.021115940
## MHHUUR      -0.014083402
## MAUT1        0.044190797
## MINKM30     -0.002391603
## MINK7512     0.024284061
## MINK123M    -0.066022536
## MINKGEM      0.033331645
## MKOOPKLA     0.036547521
## PWAPART      0.111340510
## PPERSAUT     0.113785250
## PGEZONG      0.044957846
## PWAOREG      0.113404091
## PBRAND       0.005489312
## PFIETS       0.022576284
## ABROM       -0.008462238
## AZEILPL      0.993823681
## AFIETS       0.293820438
## prob         6.240121771
```

```
# Next, we use glm() with the variables obtained from LASSO above
beta.min <- rownames(as.matrix(coef.min))
```

```r
beta.min
```

```
##  [1] "(Intercept)" "MGEMLEEF"    "MGODPR"      "MOPLHOOG"    "MBERBOER"
##  [6] "MBERMIDD"    "MHHUUR"      "MAUT1"       "MINKM30"     "MINK7512"
## [11] "MINK123M"    "MINKGEM"     "MKOOPKLA"    "PWAPART"     "PPERSAUT"
## [16] "PGEZONG"     "PWAOREG"     "PBRAND"      "PFIETS"      "ABROM"
## [21] "AZEILPL"     "AFIETS"      "prob"
```

```r
# Create the logistic regression summary
fit.logit.2 <- glm(CARAVAN~MGEMLEEF+MGODRK+MGODPR+MGODGE+MRELGE+MRELSA+MOPLHOOG+MOPLLAAG+MBERBOER+MBERMI
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```r
summary(fit.logit.2)
```

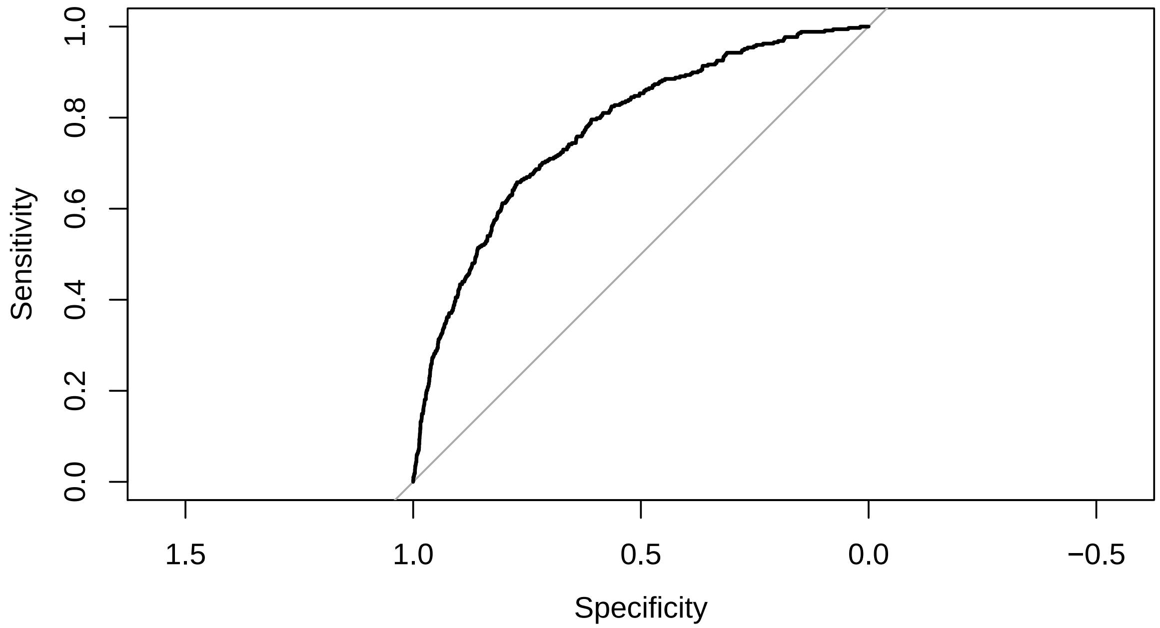
```
##
## Call:
## glm(formula = CARAVAN ~ MGEMLEEF + MGODRK + MGODPR + MGODGE +
##     MRELGE + MRELSA + MOPLHOOG + MOPLLAAG + MBERBOER + MBERMIDD +
##     MSKD + MHHUUR + MAUT1 + MINKM30 + MINK7512 + MINK123M + MINKGEM +
##     MKOOPKLA + PWAPART + PWALAND + PPERSAUT + PWERKT + PGEZONG +
##     PWAOREG + PBRAND + PFIETS + ATRACTOR + AZEILPL + APLEZIER +
##     AFIETS + ABYSTAND, family = binomial, data = caravan.train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6048  -0.3737  -0.2545  -0.1723   3.2100
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -5.097710   0.860185  -5.926 3.10e-09 ***
## MGEMLEEF      0.131799   0.081770   1.612 0.106998
## MGODRK       -0.097256   0.077559  -1.254 0.209855
## MGODPR       -0.005362   0.065432  -0.082 0.934687
## MGODGE       -0.047108   0.062390  -0.755 0.450213
## MRELGE        0.055622   0.046310   1.201 0.229724
## MRELSA       -0.039267   0.083106  -0.472 0.636574
## MOPLHOOG      0.064297   0.045600   1.410 0.158530
## MOPLLAAG     -0.050413   0.037633  -1.340 0.180377
## MBERBOER     -0.189241   0.081109  -2.333 0.019638 *
## MBERMIDD      0.059254   0.032745   1.810 0.070364 .
## MSKD         -0.037815   0.061828  -0.612 0.540795
## MHHUUR       -0.026421   0.025301  -1.044 0.296364
## MAUT1         0.049682   0.044044   1.128 0.259322
## MINKM30      -0.013434   0.044691  -0.301 0.763714
## MINK7512      0.064076   0.060656   1.056 0.290790
## MINK123M     -0.217145   0.124267  -1.747 0.080566 .
## MINKGEM       0.036568   0.076831   0.476 0.634109
## MKOOPKLA      0.044924   0.036491   1.231 0.218284
## PWAPART       0.121397   0.073804   1.645 0.099997 .
## PWALAND      -0.275223   0.202658  -1.358 0.174442
## PPERSAUT      0.230589   0.024245   9.511  < 2e-16 ***
## PWERKT       -4.948670 151.550097  -0.033 0.973951
## PGEZONG       0.185727   0.190334   0.976 0.329166
## PWAOREG       0.242599   0.103320   2.348 0.018872 *
```

```
## PBRAND        0.133751   0.039777   3.363 0.000772 ***
## PFIETS        0.526572   0.806282   0.653 0.513701
## ATRACTOR     -0.221934   0.400809  -0.554 0.579774
## AZEILPL       1.511240   1.382779   1.093 0.274437
## APLEZIER      2.057383   0.385044   5.343 9.13e-08 ***
## AFIETS        0.154498   0.552106   0.280 0.779605
## ABYSTAND      0.453355   0.308976   1.467 0.142299
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2635.5  on 5821  degrees of freedom
## Residual deviance: 2296.3  on 5790  degrees of freedom
## AIC: 2360.3
##
## Number of Fisher Scoring iterations: 15
```

```r
# Get ROC and AUC
prob=predict(fit.logit.2,type=c("response"))
caravan.train$prob=prob
g <- roc(CARAVAN ~ prob, data = caravan.train)
g
```

```
##
## Call:
## roc.formula(formula = CARAVAN ~ prob, data = caravan.train)
##
## Data: prob in 5474 controls (CARAVAN 0) < 348 cases (CARAVAN 1).
## Area under the curve: 0.7741
```

```r
plot(g)
```




```
# Incorporate loss of 0.2 since we are much more comfortable marketing to those who are less likely to p
fit.pred.2 <- rep("0", 5822)
fit.pred.2[fit.logit.2$fitted > .2] <- "1"

# Find MCE
MCE.2 <- (sum(5*(fit.pred.2[caravan.train$CARAVAN == "1"] != "1")) + sum(fit.pred.2[caravan.train$CARAVA
MCE.2
```

```
## [1] 0.2672621
```

Random Forest

```
#Building model on training data using randomForest package
set.seed(123)
n <- nrow(caravan_kaggle)
n1 <- (2/3)*n
train_index <- sample(n, n1, replace=FALSE)
length(train_index)
```

```
## [1] 6547
```

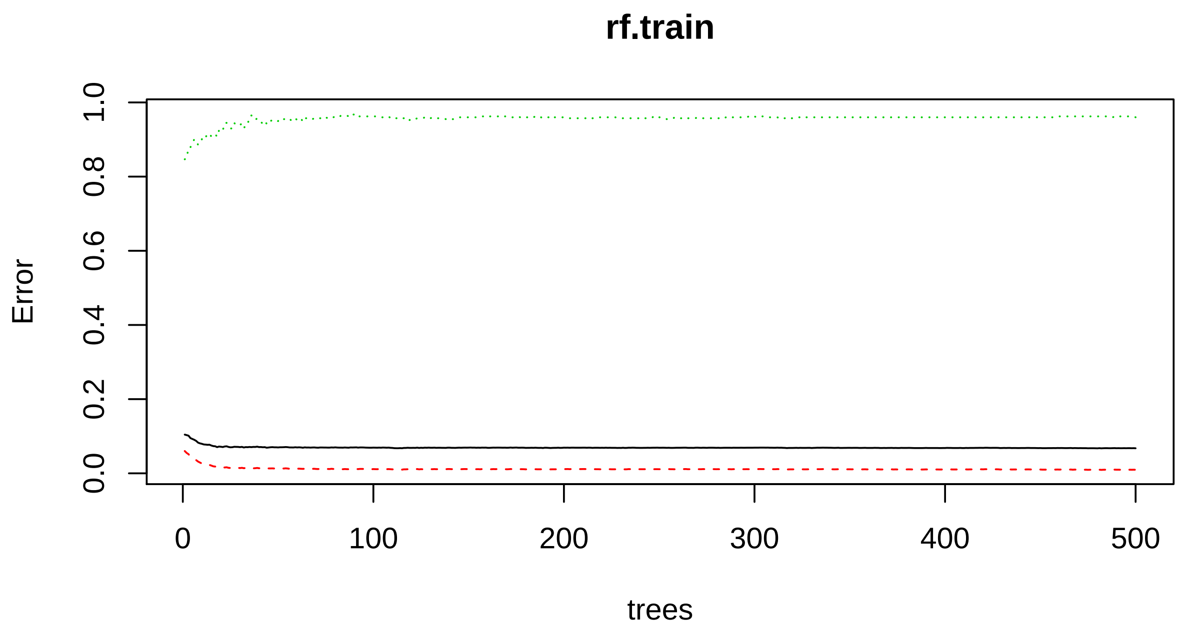
```
ctrain <- caravan_kaggle[train_index, ]
ctest <- caravan_kaggle[-train_index, ]
dim(ctrain)
```

```
## [1] 6547   86
```

```
dim(ctest)
```

```
## [1] 3274   86
```

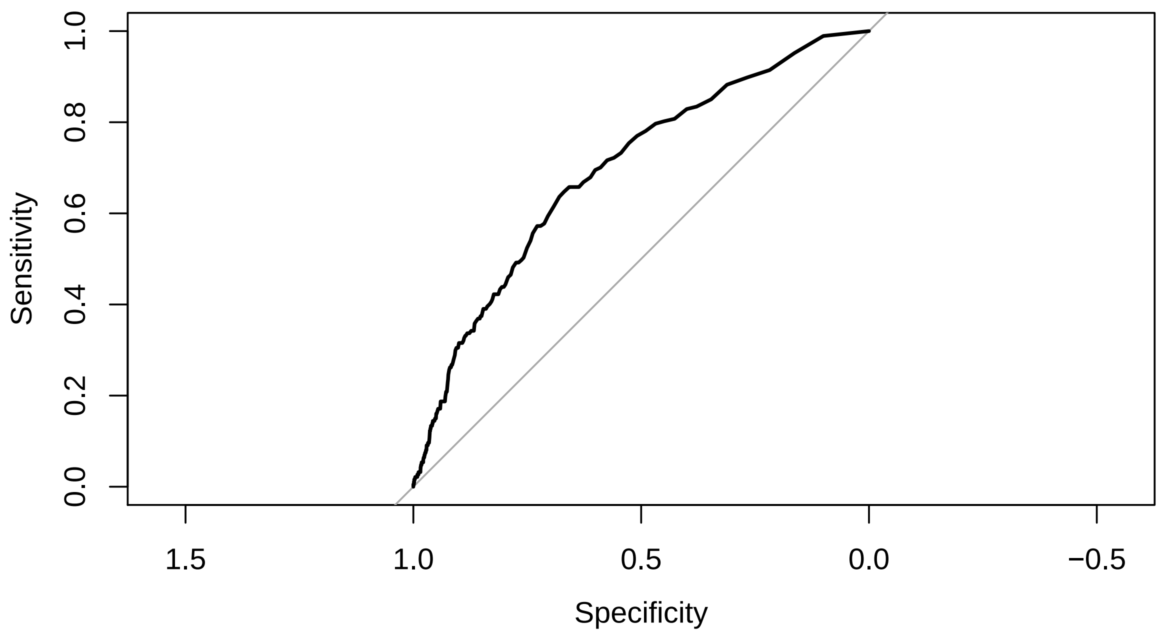
```
rf.train <- randomForest(CARAVAN~., ctrain)
plot(rf.train) #plotting the error vs number of trees to find optimal forest size
```




```r
predict.rf.yvar <- predict(rf.train, newdata=ctest)
predict.rf.prob <- predict(rf.train, newdata=ctest, type="prob")  #predicting probabilities for ROC cur
#Testing errors
mean(ctest$CARAVAN != predict.rf.yvar)
```

```
## [1] 0.06200367
```

```r
roc(ctest$CARAVAN, predict.rf.prob[,2], plot=TRUE)
```




```
##
## Call:
## roc.default(response = ctest$CARAVAN, predictor = predict.rf.prob[,     2], plot = TRUE)
##
## Data: predict.rf.prob[, 2] in 3087 controls (ctest$CARAVAN 0) < 187 cases (ctest$CARAVAN 1).
## Area under the curve: 0.6957
```

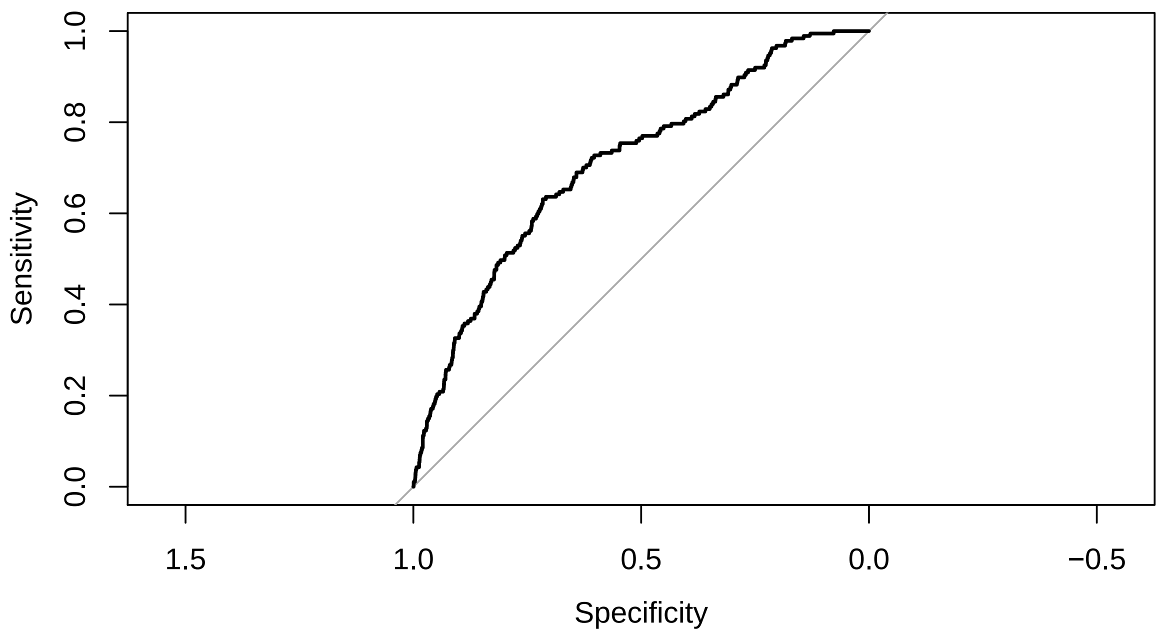
```
#Using ranger package since randomForest uses "majority vote" to grow the trees instead of offering cus
#Running on overall data to find out OOB Error
library(ranger)
rf.ranger <- ranger(CARAVAN~., caravan_kaggle, mtry = 9,
                    num.trees = 500, splitrule = "gini", importance = "impurity")
rf.ranger$prediction.error ##OOB Error
```

```
## [1] 0.0652683
```

```
#Using Test data for finding MCE/Testing Error
rf.ranger.mce <- ranger(CARAVAN~., ctrain, mtry = 9,
                    num.trees = 500, splitrule = "gini", importance = "impurity")
rf.range.pred.mce <- predict(rf.ranger.mce, ctest, type = "response")
mean(ctest$CARAVAN != rf.range.pred.mce$predictions) ##Testing error
```

```
## [1] 0.06322541
```

```
#ROC Curve and AUC
rf.ranger.ROC <- ranger(CARAVAN~., ctrain, mtry = 9,
                    num.trees = 500, splitrule = "gini", importance = "impurity", probability = T)
rf.ranger.pred.ROC <- predict(rf.ranger.ROC, ctest)$predictions[,1]
roc(ctest$CARAVAN, rf.ranger.pred.ROC, plot=TRUE)
```




```
##
## Call:
## roc.default(response = ctest$CARAVAN, predictor = rf.ranger.pred.ROC,     plot = TRUE)
##
## Data: rf.ranger.pred.ROC in 3087 controls (ctest$CARAVAN 0) > 187 cases (ctest$CARAVAN 1).
## Area under the curve: 0.7103
```

```
#Bayes Rule - Loss Function of 0.2
rf.test <- predict(rf.ranger.ROC, ctest)
rf.test.pred <- ifelse(rf.test$predictions[,2]<0.2,"0","1") #classifying probabilities less than 0.2 as
mean(ctest$CARAVAN != rf.test.pred) #MCE in testing data = Testing Error with loss function of 0.2
```

```
## [1] 0.101405
```