Data621 HW 4

jim lung

April 20, 2018

---

1. DATA EXPLORATION (25 Points)

Describe the size and the variables in the insurance training data set. Consider that too much detail will cause a manager to lose interest while too little detail will make the manager consider that you aren't doing your job. Some suggestions are given below. Please do NOT treat this as a check list of things to do to complete the assignment. You should have your own thoughts on what to tell the boss. These are just ideas.

a. Mean / Standard Deviation / Median

b. Bar Chart or Box Plot of the data

c. Is the data correlated to the target variable (or to other variables?)

d. Are any of the variables missing and need to be imputed "fixed"?

---

a. Mean / Standard Deviation / Median

```
require("plyr")
require("knitr")
require("psych")
# Let's load the data

training <- read.csv(url('https://raw.githubusercontent.com/rmalarc/DATA621/m
aster/hw04/insurance_training_data.csv'),stringsAsFactors = FALSE)

evaluation <- read.csv(url('https://raw.githubusercontent.com/rmalarc/DATA621
/master/hw04/insurance-evaluation-data.csv'),stringsAsFactors = FALSE)


columns <- colnames(training)
target <- "TARGET_FLAG"
inputs <- columns[!columns %in% c(target,"INDEX")]


summary <- describe(training[,c(target,inputs)],na.rm = TRUE)[,c("n","mean","
sd","median","min","max")]
summary$completeness <- summary$n/nrow(training)
```

```r
summary$cv <- 100*summary$sd/summary$mean

kable(summary)
```

	n	mean	sd	median	min	max	completeness	cv
TARGET_FLAG	8161	0.2638157	0.4407276	0	0	1.0	1.0000000	167.05888
TARGET_AMT	8161	1504.3246481	4704.0269298	0	0	107586.1	1.0000000	312.70025
KIDSDRIV	8161	0.1710575	0.5115341	0	0	4.0	1.0000000	299.04224
AGE	8155	44.7903127	8.6275895	45	16	81.0	0.9992648	19.26218
HOMEKIDS	8161	0.7212351	1.1163233	0	0	5.0	1.0000000	154.77938
YOJ	7707	10.4992864	4.0924742	11	0	23.0	0.9443696	38.97859
INCOME*	8161	NaN	NA	NA	Inf	-Inf	1.0000000	NA
PARENT1*	8161	NaN	NA	NA	Inf	-Inf	1.0000000	NA
HOME_VAL*	8161	NaN	NA	NA	Inf	-Inf	1.0000000	NA
MSTATUS*	8161	NaN	NA	NA	Inf	-Inf	1.0000000	NA
SEX*	8161	NaN	NA	NA	Inf	-Inf	1.0000000	NA
EDUCATION*	8161	NaN	NA	NA	Inf	-Inf	1.0000000	NA
JOB*	8161	NaN	NA	NA	Inf	-Inf	1.0000000	NA
TRAVTIME	8161	33.4857248	15.9083334	33	5	142.0	1.0000000	47.50781
CAR_USE*	8161	NaN	NA	NA	Inf	-Inf	1.0000000	NA
BLUEBOOK*	8161	NaN	NA	NA	Inf	-Inf	1.0000000	NA
TIF	8161	5.3513050	4.1466353	4	1	25.0	1.0000000	77.48830

CAR_TYPE*	8161	NaN	NA	NA	Inf	-Inf	1.0000000	NA
RED_CAR*	8161	NaN	NA	NA	Inf	-Inf	1.0000000	NA
OLDCLAIM*	8161	NaN	NA	NA	Inf	-Inf	1.0000000	NA
CLM_FREQ	8161	0.7985541	1.1584527	0	0	5.0	1.0000000	145.06878
REVOKED*	8161	NaN	NA	NA	Inf	-Inf	1.0000000	NA
MVR_PTS	8161	1.6955030	2.1471117	1	0	13.0	1.0000000	126.63568
CAR_AGE	7651	8.3283231	5.7007424	8	-3	28.0	0.9375077	68.45006
URBANICITY*	8161	NaN	NA	NA	Inf	-Inf	1.0000000	NA

```
head(training)

##   INDEX TARGET_FLAG TARGET_AMT KIDSDRIV AGE HOMEKIDS YOJ   INCOME PARENT1
## 1     1           0          0        0  60        0  11  $67,349      No
## 2     2           0          0        0  43        0  11  $91,449      No
## 3     4           0          0        0  35        1  10  $16,039      No
## 4     5           0          0        0  51        0  14                No
## 5     6           0          0        0  50        0  NA $114,986      No
## 6     7           1       2946        0  34        1  12 $125,301     Yes
##   HOME_VAL MSTATUS SEX     EDUCATION            JOB TRAVTIME    CAR_USE
## 1       $0    z_No   M           PhD   Professional       14    Private
## 2 $257,252    z_No   M z_High School z_Blue Collar       22 Commercial
## 3 $124,191     Yes z_F z_High School       Clerical        5    Private
## 4 $306,251     Yes   M  <High School z_Blue Collar       32    Private
## 5 $243,925     Yes z_F           PhD         Doctor       36    Private
## 6       $0    z_No z_F     Bachelors z_Blue Collar       46 Commercial
##   BLUEBOOK TIF   CAR_TYPE RED_CAR OLDCLAIM CLM_FREQ REVOKED MVR_PTS
## 1  $14,230  11    Minivan     yes   $4,461        2      No       3
## 2  $14,940   1    Minivan     yes       $0        0      No       0
## 3   $4,010   4      z_SUV      no  $38,690        2      No       3
## 4  $15,440   7    Minivan     yes       $0        0      No       0
## 5  $18,000   1      z_SUV      no  $19,217        2     Yes       3
## 6  $17,430   1 Sports Car      no       $0        0      No       0
##   CAR_AGE            URBANICITY
## 1      18 Highly Urban/ Urban
## 2       1 Highly Urban/ Urban
## 3      10 Highly Urban/ Urban
## 4       6 Highly Urban/ Urban
## 5      17 Highly Urban/ Urban
## 6       7 Highly Urban/ Urban
```

```r
summary(training)
```

```
##      INDEX        TARGET_FLAG       TARGET_AMT        KIDSDRIV
##  Min.   :    1   Min.   :0.0000   Min.   :     0   Min.   :0.0000
##  1st Qu.: 2559   1st Qu.:0.0000   1st Qu.:     0   1st Qu.:0.0000
##  Median : 5133   Median :0.0000   Median :     0   Median :0.0000
##  Mean   : 5152   Mean   :0.2638   Mean   :  1504   Mean   :0.1711
##  3rd Qu.: 7745   3rd Qu.:1.0000   3rd Qu.:  1036   3rd Qu.:0.0000
##  Max.   :10302   Max.   :1.0000   Max.   :107586   Max.   :4.0000
##
##       AGE          HOMEKIDS           YOJ          INCOME
##  Min.   :16.00   Min.   :0.0000   Min.   : 0.0   Length:8161
##  1st Qu.:39.00   1st Qu.:0.0000   1st Qu.: 9.0   Class :character
##  Median :45.00   Median :0.0000   Median :11.0   Mode  :character
##  Mean   :44.79   Mean   :0.7212   Mean   :10.5
##  3rd Qu.:51.00   3rd Qu.:1.0000   3rd Qu.:13.0
##  Max.   :81.00   Max.   :5.0000   Max.   :23.0
##  NA's   :6                        NA's   :454
##    PARENT1            HOME_VAL           MSTATUS
##  Length:8161        Length:8161        Length:8161
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##      SEX             EDUCATION             JOB              TRAVTIME
##  Length:8161        Length:8161        Length:8161        Min.   :  5.00
##  Class :character   Class :character   Class :character   1st Qu.: 22.00
##  Mode  :character   Mode  :character   Mode  :character   Median : 33.00
##                                                           Mean   : 33.49
##                                                           3rd Qu.: 44.00
##                                                           Max.   :142.00
##
##    CAR_USE            BLUEBOOK              TIF            CAR_TYPE
##  Length:8161        Length:8161        Min.   : 1.000   Length:8161
##  Class :character   Class :character   1st Qu.: 1.000   Class :character
##  Mode  :character   Mode  :character   Median : 4.000   Mode  :character
##                                        Mean   : 5.351
##                                        3rd Qu.: 7.000
##                                        Max.   :25.000
##
##    RED_CAR            OLDCLAIM            CLM_FREQ         REVOKED
##  Length:8161        Length:8161        Min.   :0.0000   Length:8161
##  Class :character   Class :character   1st Qu.:0.0000   Class :character
##  Mode  :character   Mode  :character   Median :0.0000   Mode  :character
##                                        Mean   :0.7986
##                                        3rd Qu.:2.0000
##                                        Max.   :5.0000
##
```

```
##     MVR_PTS          CAR_AGE        URBANICITY
##  Min.   : 0.000   Min.   :-3.000   Length:8161
##  1st Qu.: 0.000   1st Qu.: 1.000   Class :character
##  Median : 1.000   Median : 8.000   Mode  :character
##  Mean   : 1.696   Mean   : 8.328
##  3rd Qu.: 3.000   3rd Qu.:12.000
##  Max.   :13.000   Max.   :28.000
##                   NA's   :510
```

2. DATA PREPARATION (25 Points)

Describe how you have transformed the data by changing the original variables or creating new variables. If you did transform the data or create new variables, discuss why you did this. Here are some possible transformations.

a. Fix missing values (maybe with a Mean or Median value)

b. Create flags to suggest if a variable was missing

c. Transform data by putting it into buckets

d. Mathematical transforms such as log or square root (or use Box-Cox)

e. Combine variables (such as ratios or adding or multiplying) to create new variables

Data Transformations

Based on the dataset description we need to:

- Convert INCOME to numeric, replace 0 for NA
- Convert PARENT1 to flag (1/0)
- Convert HOME_VAL to NON_HOMEOWNER flag
- Convert MSTATUS to Flag IS_SINGLE (1/0)
- Convert SEX to Flag (IS_MALE)
- Breakout EDUCATION into ED_HS, ED_BACHELORS,ED_MASTERS, ED_PHD
- Breakout JOB into JOB_BLUE_COLLAR, JOB_CLERICAL, JOB_PROFESSIONAL, JOB_MANAGERIAL, JOB_LAWYER, JOB_STUDENT,JOB_DOCTOR, JOB_HOME_MAKER
- Convert CAR_USE to flat(1/0 IS_COMMERCIAL)
- Convert BLUEBOOK to numeric
- Breakout CAR_TYPE into: CAR_PANEL_TRUCK,CAR_PICKUP,CAR_SPORTS_CAR,CAR_VAN,CAR_SUV
- Convert RED_CAR to flag (1/0)
- Convert OLDCLAIM to numeric

- Convert REVOKED to flag (1/0)
- Convert URBANICITY to flag (1/0 IS_URBAN)

As a convention, all binary variables will be prefixed with “_BIN"

```r
parseStringValue <- function(v, zeroToNa){
  tmpVal <- as.numeric(gsub("[\\$,]","", v))
  if (!is.na(tmpVal) && tmpVal == 0 && zeroToNa) { NA } else {tmpVal}
}

transform <- function(d){
  outputCols<- c("TARGET_FLAG","TARGET_AMT","AGE", "YOJ", "CAR_AGE","KIDSDRIV
","HOMEKIDS","TRAVTIME","TIF","CLM_FREQ","MVR_PTS")


  #* Convert INCOME to numeric, replace 0 for NA
  d['INCOME'] <- parseStringValue(d['INCOME'],TRUE)
  outputCols <- c(outputCols,'INCOME')

  #* Convert PARENT1 to flag (1/0)
  d['PARENT1_BIN'] <- if (d['PARENT1']=="Yes") {1} else {0}
  outputCols <- c(outputCols,'PARENT1_BIN')

  #* Convert HOME_VAL to NON_HOMEOWNER flag
  d['NON_HOMEOWNER_BIN'] <- if (is.na(parseStringValue(d['HOME_VAL'],TRUE)))
{1} else {0}
  outputCols <- c(outputCols,'NON_HOMEOWNER_BIN')

  #* Convert MSTATUS to Flag  IS_SINGLE (1/0
  #levels(training$MSTATUS)
  d['IS_SINGLE_BIN'] <- if (d['MSTATUS']=="z_No") {1} else {0}
  outputCols <- c(outputCols,'IS_SINGLE_BIN')

  #* Convert SEX to Flag (IS_MALE)
  d['IS_MALE_BIN'] <- if (d['SEX']=="M") {1} else {0}
  outputCols <- c(outputCols,'IS_MALE_BIN')

  #* Breakout EDUCATION into ED_HS, ED_BACHELORS,ED_MASTERS, ED_PHD
  d['ED_HS_BIN'] <- if (d['EDUCATION']=="z_High School") {1} else {0}
  d['ED_BACHELORS_BIN'] <- if (d['EDUCATION']=="Bachelors") {1} else {0}
  d['ED_MASTERS_BIN'] <- if (d['EDUCATION']=="Masters") {1} else {0}
  d['ED_PHD_BIN'] <- if (d['EDUCATION']=="PhD") {1} else {0}
  outputCols <- c(outputCols,'ED_HS_BIN','ED_BACHELORS_BIN','ED_MASTERS_BIN',
'ED_PHD_BIN')

  #* Breakout JOB into JOB_BLUE_COLLAR, JOB_CLERICAL, JOB_PROFESSIONAL, JOB_M
ANAGERIAL, JOB_LAWYER, JOB_STUDENT, JOB_DOCTOR, JOB_HOME_MAKER
  d['JOB_BLUE_COLLAR_BIN'] <- if (d['JOB']=="z_Blue Collar") {1} else {0}
  d['JOB_CLERICAL_BIN'] <- if (d['JOB']=="Clerical") {1} else {0}
  d['JOB_PROFESSIONAL_BIN'] <- if (d['JOB']=="Professional") {1} else {0}
  d['JOB_MANAGERIAL_BIN'] <- if (d['JOB']=="Manager") {1} else {0}
  d['JOB_LAWYER_BIN'] <- if (d['JOB']=="Lawyer") {1} else {0}
  d['JOB_STUDENT_BIN'] <- if (d['JOB']=="Student") {1} else {0}
  d['JOB_DOCTOR_BIN'] <- if (d['JOB']=="Doctor") {1} else {0}
```

```r
  d['JOB_HOME_MAKER_BIN'] <- if (d['JOB']=="Home Maker") {1} else {0}
  outputCols <- c(outputCols,'JOB_BLUE_COLLAR_BIN', 'JOB_CLERICAL_BIN', 'JOB_PROFESSIONAL_BIN', 'JOB_MANAGERIAL_BIN', 'JOB_LAWYER_BIN', 'JOB_STUDENT_BIN', 'JOB_DOCTOR_BIN', 'JOB_HOME_MAKER_BIN')

  #* Convert CAR_USE to flat(1/0 IS_COMMERCIAL)
  #levels(training$CAR_USE)
  d['IS_COMMERCIAL_BIN'] <- if (d['CAR_USE']=="Commercial") {1} else {0}
  outputCols <- c(outputCols,'IS_COMMERCIAL_BIN')


  #* Convert BLUEBOOK to numeric
  d['BLUEBOOK'] <- parseStringValue(d['BLUEBOOK'],FALSE)
  outputCols <- c(outputCols,'BLUEBOOK')

  #* Breakout CAR_TYPE into: CAR_PANEL_TRUCK,CAR_PICKUP,CAR_SPORTS_CAR,CAR_VAN,CAR_SUV
  #levels(training$CAR_TYPE)
  d['CAR_PANEL_TRUCK_BIN'] <- if (d['CAR_TYPE']=="Panel Truck") {1} else {0}
  d['CAR_PICKUP_BIN'] <- if (d['CAR_TYPE']=="Pickup") {1} else {0}
  d['CAR_SPORTS_CAR_BIN'] <- if (d['CAR_TYPE']=="Sports Car") {1} else {0}
  d['CAR_VAN_BIN'] <- if (d['CAR_TYPE']=="Van") {1} else {0}
  d['CAR_SUV_BIN'] <- if (d['CAR_TYPE']=="z_SUV") {1} else {0}
  outputCols <- c(outputCols,'CAR_PANEL_TRUCK_BIN','CAR_PICKUP_BIN','CAR_SPORTS_CAR_BIN','CAR_VAN_BIN','CAR_SUV_BIN')

  #* Convert RED_CAR to flag (1/0)
  #levels(training$RED_CAR)
  d['RED_CAR_BIN'] <- if (d['RED_CAR']=="yes") {1} else {0}
  outputCols <- c(outputCols,'RED_CAR_BIN')

  #* Convert OLDCLAIM to numeric
  #levels(training$OLDCLAIM)
  d['OLDCLAIM'] <- parseStringValue(d['OLDCLAIM'],TRUE)
  outputCols <- c(outputCols,'OLDCLAIM')

  #* Convert REVOKED to flag (1/0)
  #levels(training$REVOKED)
  d['REVOKED_BIN'] <- if (d['REVOKED']=="Yes") {1} else {0}
  outputCols <- c(outputCols,'REVOKED_BIN')

  #* Convert URBANICITY to flag (1/0 IS_URBAN)
  #levels(training$URBANICITY)
  d['IS_URBAN_BIN'] <- if (d['URBANICITY']=="Highly Urban/ Urban") {1} else {0}
  outputCols <- c(outputCols,'IS_URBAN_BIN')


  r <- as.numeric(d[outputCols])
```

```
  names(r) <- outputCols
  r
}

# form dataframe by function
training_trans<-data.frame(t(rbind(apply(training,1,transform))))
evaluation_trans<-data.frame(t(rbind(apply(evaluation,1,transform))))

columns <- colnames(training_trans)
target_bin <- c("TARGET_FLAG")
target_lm <- c("TARGET_AMT")
target <- c(target_bin,target_lm)
inputs_bin <- columns[grep("_BIN",columns)]
inputs_num <- columns[!columns %in% c(target,"INDEX",inputs_bin)]
inputs<- c(inputs_bin,inputs_num)
```

Data Imputations

Imputations

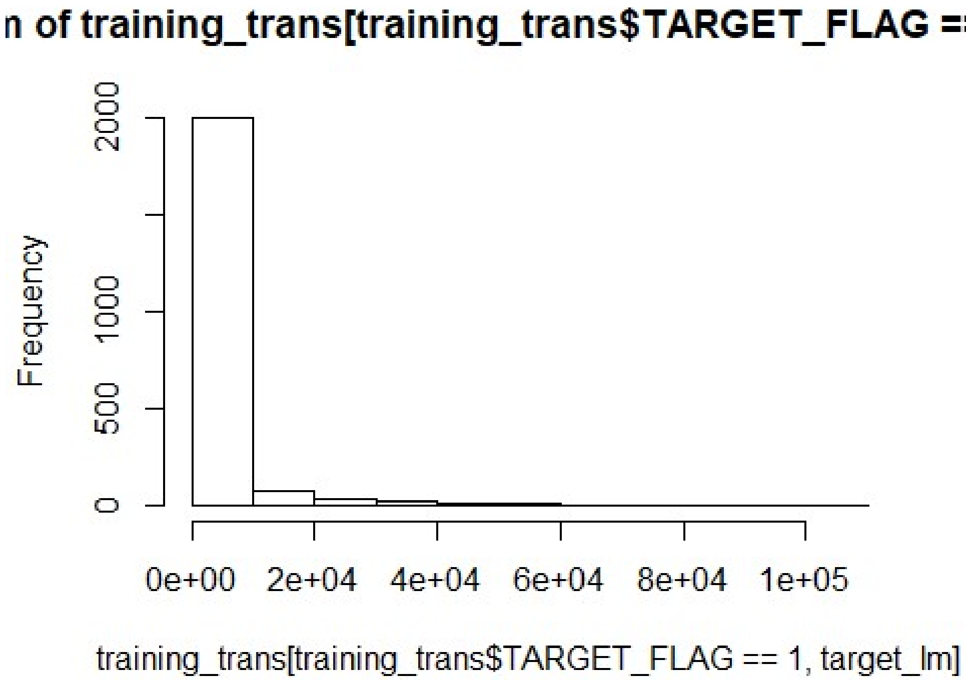
- Fill missing nummerical values with mean for: AGE, YOJ, CAR_AGE, INCOME
- Impute missing OLDCLAIM with zeros

```
# impute
impute <- function (d) {
  d[is.na(d$AGE),]$AGE <- mean(d$AGE,na.rm = TRUE)
  d[is.na(d$YOJ),]$YOJ <- mean(d$YOJ,na.rm = TRUE)
  d[is.na(d$CAR_AGE),]$CAR_AGE <- mean(d$CAR_AGE,na.rm = TRUE)
  d[is.na(d$INCOME),]$INCOME <- mean(d$INCOME,na.rm = TRUE)
  d[is.na(d$OLDCLAIM),]$OLDCLAIM <- 0
  d
}
training_trans<-impute(training_trans)
evaluation_trans<-impute(evaluation_trans)
```

Transformation Analysis

TARGET_NUM

```
hist(training_trans[training_trans$TARGET_FLAG==1,target_lm])
```



The distribution of values of the response target_lm suggest that we may benefit from a log tranformation on the response.

For better linear pattern, we should get a better linear fit. A log transformation of the target seems adequate, aside from some negative values that need to be zeroed out, it is not evident that any outliers of the predictors may skew the linear fit. With that, no further transformations seem required.

Transformations Implementation

Numerical Transformations:

- Cap AGE at 70, negative values not permitted
- Cap YOJ at 20, negative values not permitted
- Cap CAR_AGE at 20, negative values not permitted
- Cap KIDSDRIV at 3, negative values not permitted
- Cap HOMEKIDS at 4, negative values not permitted
- Cap TRAVTIME at 75, negative values not permitted
- Cap TIF at 17, negative values not permitted
- Cap CLM_FREQ at 4, negative values not permitted
- Cap MVR_PTS at 10, negative values not permitted
- Cap INCOME at 175000, negative values not permitted
- Cap BLUEBOOK at 40000, negative values not permitted
- Cap OLDCLAIM at 40000, negative values not permitted

```r
# Cap values

d<- training_trans
capColumns <- function(d){
  outputCols<- colnames(d)


  #* Cap AGE at 70, negative values not permitted
  d[d$AGE <0, 'AGE'] <- 0
  d[d$AGE >=70, 'AGE'] <- 70

  #* Cap YOJ at 20, negative values not permitted
  d[d$YOJ <0, 'YOJ'] <- 0
  d[d$YOJ >=20, 'YOJ'] <- 20

  #* Cap CAR_AGE at 20, negative values not permitted
  d[d$CAR_AGE <0, 'CAR_AGE'] <- 0
  d[d$CAR_AGE >=20, 'CAR_AGE'] <- 20

  #* Cap KIDSDRIV at 3, negative values not permitted
  d[d$KIDSDRIV <0, 'KIDSDRIV'] <- 0
  d[d$KIDSDRIV >=3, 'KIDSDRIV'] <- 3

  #* Cap HOMEKIDS at 4, negative values not permitted
  d[d$HOMEKIDS <0, 'HOMEKIDS'] <- 0
  d[d$HOMEKIDS >=4, 'HOMEKIDS'] <- 4

  #* Cap TRAVTIME at 75, negative values not permitted
  d[d$TRAVTIME <0, 'TRAVTIME'] <- 0
  d[d$TRAVTIME >=75, 'TRAVTIME'] <- 75

  #* Cap TIF at 17, negative values not permitted
  d[d$TIF <0, 'TIF'] <- 0
  d[d$TIF >=17, 'TIF'] <- 17

  #* Cap CLM_FREQ at 4, negative values not permitted
  d[d$CLM_FREQ <0, 'CLM_FREQ'] <- 0
  d[d$CLM_FREQ >=4, 'CLM_FREQ'] <- 4

  #* Cap MVR_PTS at 10, negative values not permitted
  d[d$MVR_PTS <0, 'MVR_PTS'] <- 0
  d[d$MVR_PTS >=10, 'MVR_PTS'] <- 10

  #* Cap INCOME at 175000, negative values not permitted
  d[d$INCOME <0, 'INCOME'] <- 0
  d[d$INCOME >=175000, 'INCOME'] <- 175000

  #* Cap BLUEBOOK at 40000, negative values not permitted
  d[d$BLUEBOOK <0, 'BLUEBOOK'] <- 0
```

```
  d[d$BLUEBOOK >=40000, 'BLUEBOOK'] <- 40000

  #* Cap OLDCLAIM at 40000, negative values not permitted
  d[d$OLDCLAIM <0, 'OLDCLAIM'] <- 0
  d[d$OLDCLAIM >=40000, 'OLDCLAIM'] <- 40000

  d

}


training_trans <- capColumns(training_trans)
evaluation_trans <- capColumns(evaluation_trans)
```

Final summary

```
summary <- describe(training_trans[,c(target,inputs)])[,c("n","mean","sd","median","min","max")]
summary$completeness <- summary$n/nrow(training_trans)
summary$cv <- 100*summary$sd/summary$mean

kable(summary)
```

	n	mean	sd	median	min	max	completeness	cv
TARGET_FLAG	8161	2.638157e-01	4.407276e-01	0.000000	0	1.0	1	167.05888
TARGET_AMT	8161	1.504325e+03	4.704027e+03	0.000000	0	107586.1	1	312.70025
PARENT1_BIN	8161	1.319691e-01	3.384779e-01	0.000000	0	1.0	1	256.48267
NON_HOMEOWNER_BIN	8161	3.379488e-01	4.730400e-01	0.000000	0	1.0	1	139.97387
IS_SINGLE_BIN	8161	4.003186e-01	4.899929e-01	0.000000	0	1.0	1	122.40073
IS_MALE_BIN	8161	4.639137e-01	4.987266e-01	0.000000	0	1.0	1	107.50418
ED_HS_BIN	8161	2.855042e-01	4.516819e-01	0.000000	0	1.0	1	158.20499
ED_BACHELORS_BIN	8161	2.747212e-01	4.464010e-01	0.000000	0	1.0	1	162.49237
ED_MASTERS_BIN	8161	2.031614e-01	4.023763e-01	0.000000	0	1.0	1	198.05747
ED_PHD_BIN	8161	8.920480e-02	2.850565e-01	0.000000	0	1.0	1	319.55306

JOB_BLUE_COLLAR_BIN	8161	2.236246e-01	4.166988e-01	0.000000	0	1.0	1	186.33857
JOB_CLERICAL_BIN	8161	1.557407e-01	3.626316e-01	0.000000	0	1.0	1	232.84314
JOB_PROFESSIONAL_BIN	8161	1.368705e-01	3.437316e-01	0.000000	0	1.0	1	251.13642
JOB_MANAGERIAL_BIN	8161	1.210636e-01	3.262212e-01	0.000000	0	1.0	1	269.46264
JOB_LAWYER_BIN	8161	1.023159e-01	3.030818e-01	0.000000	0	1.0	1	296.22167
JOB_STUDENT_BIN	8161	8.724420e-02	2.822099e-01	0.000000	0	1.0	1	323.47119
JOB_DOCTOR_BIN	8161	3.014340e-02	1.709922e-01	0.000000	0	1.0	1	567.26308
JOB_HOME_MAKER_BIN	8161	7.854430e-02	2.690427e-01	0.000000	0	1.0	1	342.53623
IS_COMMERCIAL_BIN	8161	3.711555e-01	4.831436e-01	0.000000	0	1.0	1	130.17282
CAR_PANEL_TRUCK_BIN	8161	8.283300e-02	2.756465e-01	0.000000	0	1.0	1	332.77383
CAR_PICKUP_BIN	8161	1.701997e-01	3.758312e-01	0.000000	0	1.0	1	220.81774
CAR_SPORTS_CAR_BIN	8161	1.111383e-01	3.143226e-01	0.000000	0	1.0	1	282.82106
CAR_VAN_BIN	8161	9.190050e-02	2.889031e-01	0.000000	0	1.0	1	314.36514
CAR_SUV_BIN	8161	2.810930e-01	4.495603e-01	0.000000	0	1.0	1	159.93295
RED_CAR_BIN	8161	2.913859e-01	4.544287e-01	0.000000	0	1.0	1	155.95427
REVOKED_BIN	8161	1.225340e-01	3.279216e-01	0.000000	0	1.0	1	267.61685
IS_URBAN_BIN	8161	7.954907e-01	4.033673e-01	1.000000	0	1.0	1	50.70672
AGE	8161	4.478517e+01	8.607250e+00	45.000000	16	70.0	1	19.21898
YOJ	8161	1.049855e+01	3.974963e+00	11.000000	0	20.0	1	37.86202
CAR_AGE	8161	8.297322e+00	5.443049e+00	8.328323	0	20.0	1	65.60007

KIDSDRIV	8161	1.705673e-01	5.083338e-01	0.000000	0	3.0	1	298.02528
HOMEKIDS	8161	7.195197e-01	1.110499e+00	0.000000	0	4.0	1	154.33896
TRAVTIME	8161	3.338684e+01	1.557003e+01	33.000000	5	75.0	1	46.63522
TIF	8161	5.334273e+00	4.090881e+00	4.000000	1	17.0	1	76.69051
CLM_FREQ	8161	7.963485e-01	1.151381e+00	0.000000	0	4.0	1	144.58254
MVR_PTS	8161	1.693420e+00	2.138207e+00	1.000000	0	10.0	1	126.26560
INCOME	8161	6.627132e+04	3.934498e+04	66367.000000	5	175000.0	1	59.36954
BLUEBOOK	8161	1.566945e+04	8.272602e+03	14440.000000	1500	40000.0	1	52.79447
OLDCLAIM	8161	3.957800e+03	8.408736e+03	0.000000	0	40000.0	1	212.45985

```
#head(training_trans)
#summary(training_trans)
```

distribution of the values for each of the variables

Here's the distribution of the values for each of the variables

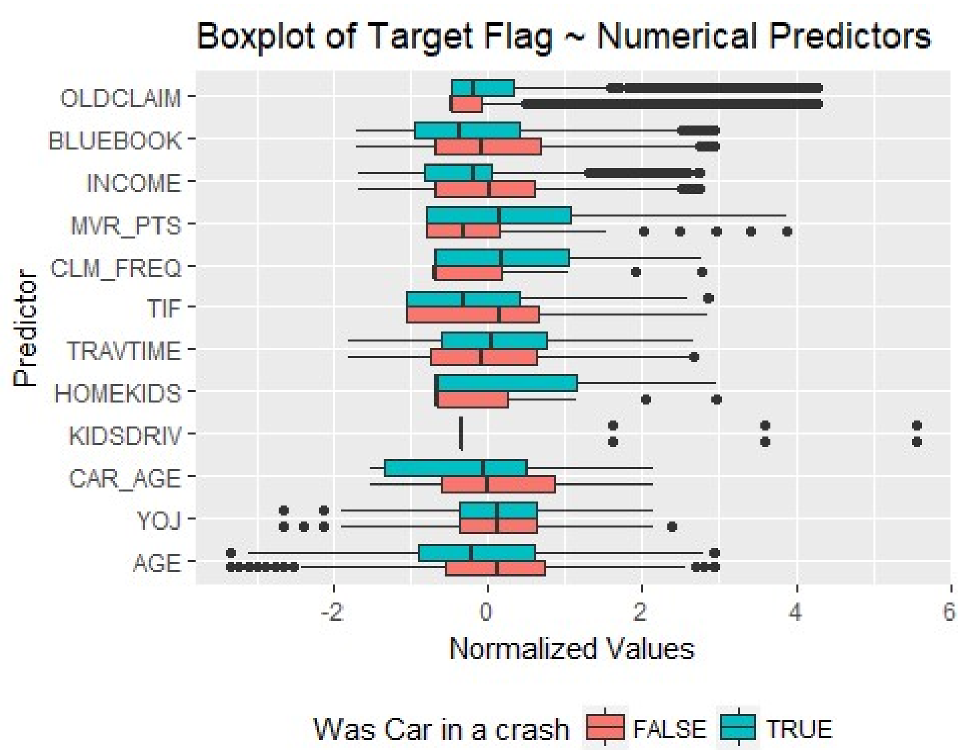
we get a view of the normalized values:

Binary target variable

```
head(data.frame(scale(training_trans[,inputs_num])))

##          AGE          YOJ    CAR_AGE  KIDSDRIV   HOMEKIDS    TRAVTIME
## 1  1.7676765  0.1261518097  1.7825814 -0.335542 -0.6479245 -1.24513858
## 2 -0.2074026  0.1261518097 -1.3406681 -0.335542 -0.6479245 -0.73133083
## 3 -1.1368516 -0.1254228265  0.3128170 -0.335542  0.2525714 -1.82317230
## 4  0.7220464  0.8808757184 -0.4220653 -0.335542 -0.6479245 -0.08907113
## 5  0.6058652  0.0001849587  1.5988609 -0.335542 -0.6479245  0.16783274
## 6 -1.2530327  0.3777264459 -0.2383447 -0.335542  0.2525714  0.81009244
##          TIF   CLM_FREQ    MVR_PTS      INCOME     BLUEBOOK    OLDCLAIM
## 1  1.3849651  1.0453982  0.6110635  0.02739048 -0.17400176  0.05984256
## 2 -1.0594962 -0.6916465 -0.7919813  0.63992102 -0.08817629 -0.47067716
## 3 -0.3261578  1.0453982  0.6110635 -1.27671500 -1.40940506  4.13049027
## 4  0.4071806 -0.6916465 -0.7919813  0.02510137 -0.02773582 -0.47067716
## 5 -1.0594962  1.0453982  0.6110635  1.23814224  0.28171941  1.81468429
## 6 -1.0594962 -0.6916465 -0.7919813  1.50031039  0.21281727 -0.47067716
```

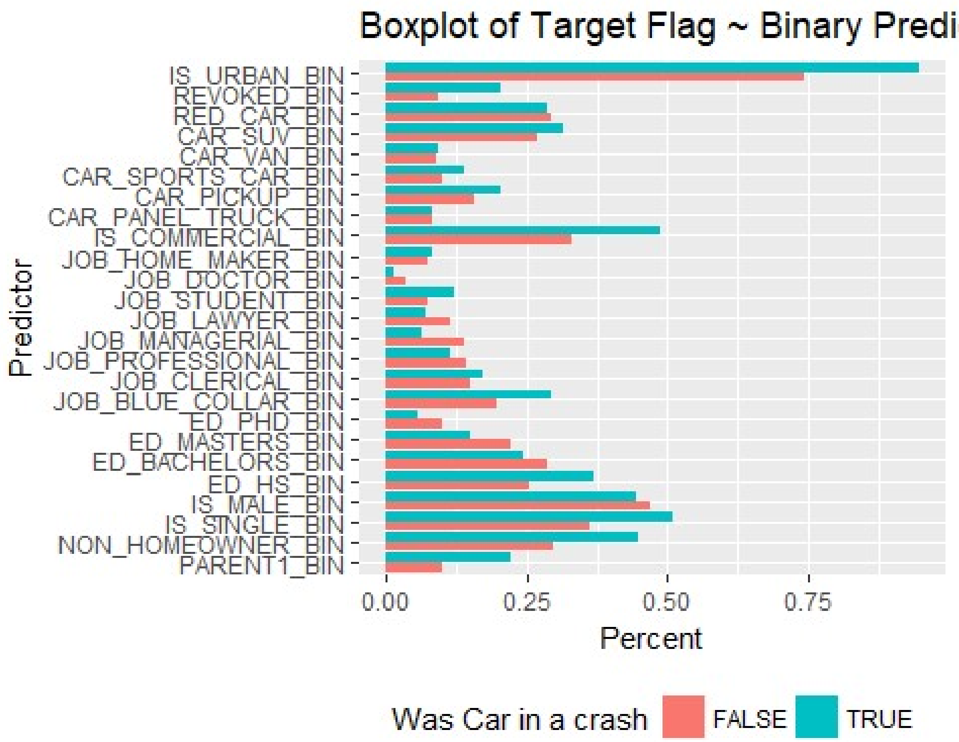
Boxplot of Target Flag vs Numerical Predictors and Target Flag vs Binary Predictors

```
require("reshape2")
require("ggplot2")
detach(package:plyr)
require("dplyr")

# Let's melt the DF so that we can plot it more easily
training_normalized <- cbind(data.frame(scale(training_trans[,inputs_num])),t
raining_trans[,c(inputs_bin,target)])
training_normalized$TARGET_FLAG <- training_normalized$TARGET_FLAG==1

ggplot(melt(training_normalized, measure.vars = inputs_num),
       aes(x=variable,y=value)
       )+
    geom_boxplot(aes(fill = factor(TARGET_FLAG)))+
  guides(fill=guide_legend(title="Was Car in a crash")) +
   theme(legend.position="bottom")+
    coord_flip()+
  labs(title="Boxplot of Target Flag ~ Numerical Predictors", y="Normalized V
alues", x="Predictor")
```




```
bin_summary <- melt(training_normalized[,c(inputs_bin,target_bin)], measure.v
ars = inputs_bin) %>%
  group_by(TARGET_FLAG,variable) %>%
  summarise(pct = sum(value)/length(value))
```

```
ggplot(bin_summary, aes(variable, pct)) +
  geom_bar(aes(fill = TARGET_FLAG), position = "dodge", stat="identity")+
  guides(fill=guide_legend(title="Was Car in a crash")) +
   theme(legend.position="bottom")+
    coord_flip()+
  labs(title="Boxplot of Target Flag ~ Binary Predictors", y="Percent", x="Pr
edictor")
```



Correlations

```
summary_positive <- describe(training_normalized[training_normalized$TARGET_F
LAG==1,c(target_bin,inputs)])[,c("mean","n")]
summary_negative <- describe(training_normalized[training_normalized$TARGET_F
LAG==0,c(target_bin,inputs)])[,c("mean","n")]
summary_by_target <- merge(summary_positive,summary_negative,by="row.names")
colnames(summary_by_target) <- c("Variable","In car crash - Avg","In car cras
h - n","NOT In car crash - Avg", "NOT In car crash - n")
summary_by_target$delta <- abs(summary_by_target[,"In car crash - Avg"]-summa
ry_by_target[,"NOT In car crash - Avg"])

kable(summary_by_target[order(-summary_by_target$delta),])
```

Variable	In car crash - Avg	In car crash - n	NOT In car crash - Avg	NOT In car crash - n	delta

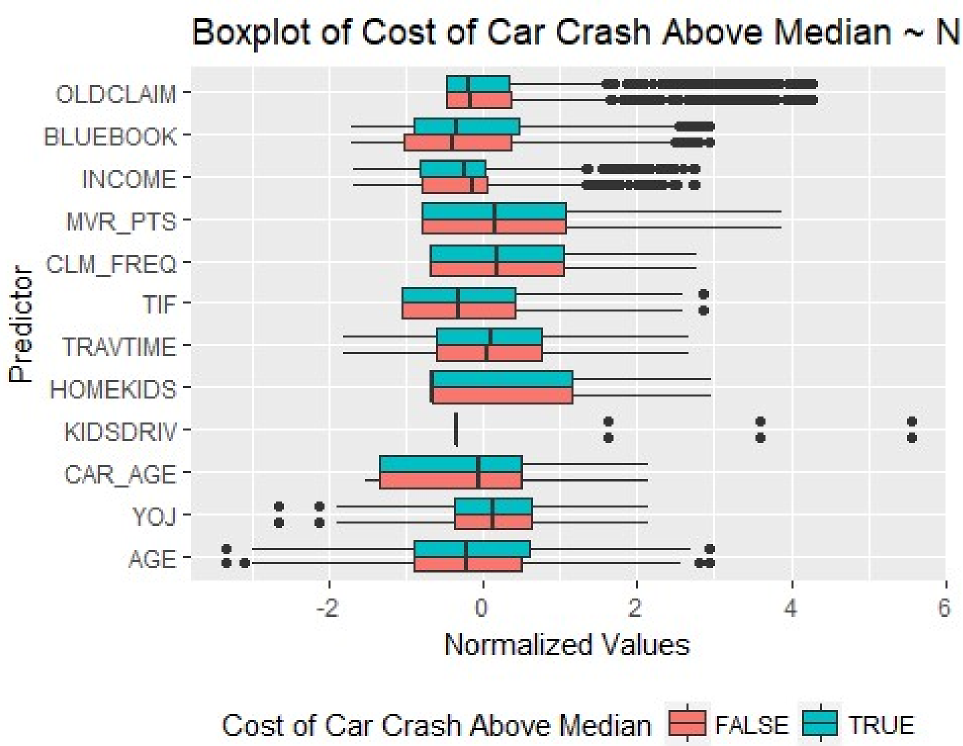
29	MVR_PTS	0.3653840	2153	-0.1309374	6008	0.4963214
9	CLM_FREQ	0.3624404	2153	-0.1298825	6008	0.4923229
31	OLDCLAIM	0.2374517	2153	-0.0850921	6008	0.3225438
15	INCOME	-0.1943105	2153	0.0696322	6008	0.2639427
14	HOMEKIDS	0.1931797	2153	-0.0692270	6008	0.2624067
2	BLUEBOOK	-0.1762139	2153	0.0631472	6008	0.2393611
28	KIDSDRIV	0.1733929	2153	-0.0621363	6008	0.2355291
1	AGE	-0.1727074	2153	0.0618906	6008	0.2345980
3	CAR_AGE	-0.1617485	2153	0.0579635	6008	0.2197120
19	IS_URBAN_BIN	0.9465862	2153	0.7413449	6008	0.2052413
36	TIF	-0.1372315	2153	0.0491777	6008	0.1864092
16	IS_COMMERCIAL_BIN	0.4862982	2153	0.3298935	6008	0.1564047
38	YOJ	-0.1142741	2153	0.0409507	6008	0.1552248
30	NON_HOMEOWNER_BIN	0.4491407	2153	0.2981025	6008	0.1510382
18	IS_SINGLE_BIN	0.5109150	2153	0.3606858	6008	0.1502292
32	PARENT1_BIN	0.2210869	2153	0.1000333	6008	0.1210536
37	TRAVTIME	0.0865733	2153	-0.0310240	6008	0.1175973
34	REVOKED_BIN	0.2057594	2153	0.0927097	6008	0.1130497
11	ED_HS_BIN	0.3683233	2153	0.2558256	6008	0.1124977
20	JOB_BLUE_COLLAR_BIN	0.2944728	2153	0.1982357	6008	0.0962371
25	JOB_MANAGERIAL_BIN	0.0636321	2153	0.1416445	6008	0.0780123
12	ED_MASTERS_BIN	0.1518811	2153	0.2215379	6008	0.0696569
27	JOB_STUDENT_BIN	0.1235485	2153	0.0742344	6008	0.0493142
5	CAR_PICKUP_BIN	0.2057594	2153	0.1574567	6008	0.0483027
7	CAR_SUV_BIN	0.3149094	2153	0.2689747	6008	0.0459347
10	ED_BACHELORS_BIN	0.2429169	2153	0.2861185	6008	0.0432016
24	JOB_LAWYER_BIN	0.0710636	2153	0.1135153	6008	0.0424517
13	ED_PHD_BIN	0.0580585	2153	0.1003662	6008	0.0423077
6	CAR_SPORTS_CAR_BIN	0.1411983	2153	0.1003662	6008	0.0408321
26	JOB_PROFESSIONAL_BIN	0.1147236	2153	0.1448069	6008	0.0300833
17	IS_MALE_BIN	0.4463539	2153	0.4702064	6008	0.0238525
22	JOB_DOCTOR_BIN	0.0134696	2153	0.0361185	6008	0.0226489
21	JOB_CLERICAL_BIN	0.1723177	2153	0.1498003	6008	0.0225174
33	RED_CAR_BIN	0.2861124	2153	0.2932756	6008	0.0071632
23	JOB_HOME_MAKER_BIN	0.0836043	2153	0.0767310	6008	0.0068732
8	CAR_VAN_BIN	0.0933581	2153	0.0913782	6008	0.0019799

4	CAR_PANEL_TRUCK_BIN	0.0826753	2153	0.0828895	6008	0.0002141
35	TARGET_FLAG	NaN	2153	NaN	6008	NaN

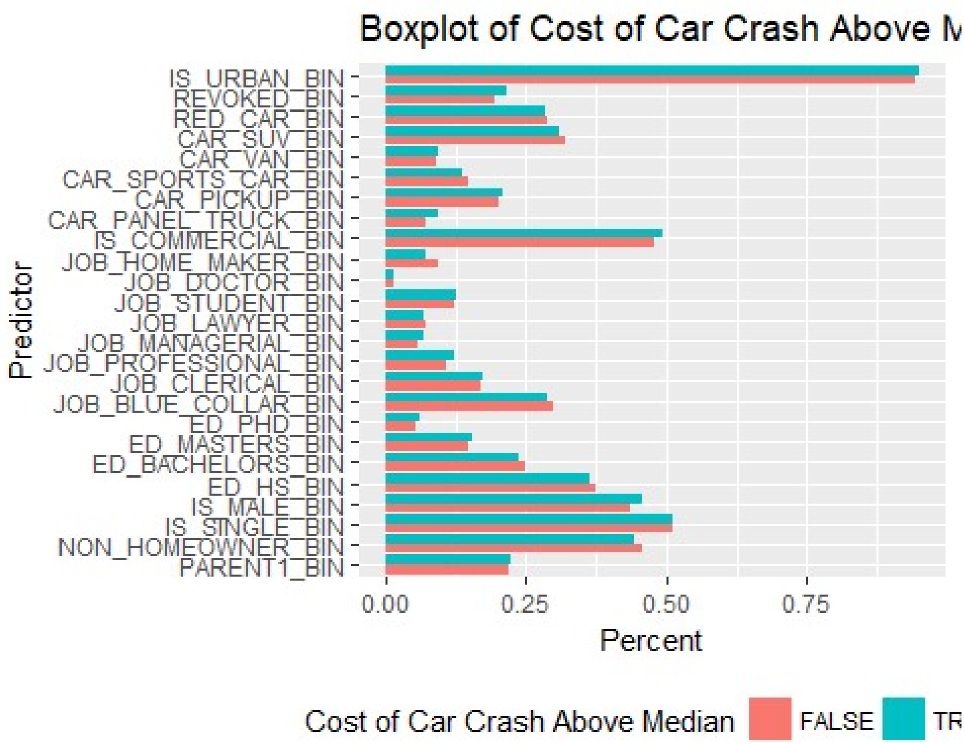
Numerical target variable - Cost of Car Crash

For our descriptive stats & intuitive understanding, let's discretize the car crash into Above / Below median cost

```
# Let's melt the DF so that we can plot it more easily
training_normalized<-training_normalized[training_normalized$TARGET_FLAG,]

training_normalized$TARGET_FLAG <- training_normalized$TARGET_AMT > median(tr
aining_normalized$TARGET_AMT)

ggplot(melt(training_normalized, measure.vars = inputs_num),
       aes(x=variable,y=value)
       )+
    geom_boxplot(aes(fill = factor(TARGET_FLAG)))+
  guides(fill=guide_legend(title="Cost of Car Crash Above Median")) +
   theme(legend.position="bottom")+
    coord_flip()+
  labs(title="Boxplot of Cost of Car Crash Above Median ~ Numerical Predictor
s", y="Normalized Values", x="Predictor")
```



```
bin_summary <- melt(training_normalized[,c(inputs_bin,target_bin)], measure.v
ars = inputs_bin) %>%
  group_by(TARGET_FLAG,variable) %>%
  summarise(pct = sum(value)/length(value))

ggplot(bin_summary, aes(variable, pct)) +
  geom_bar(aes(fill = TARGET_FLAG), position = "dodge", stat="identity")+
  guides(fill=guide_legend(title="Cost of Car Crash Above Median")) +
   theme(legend.position="bottom")+
    coord_flip()+
  labs(title="Boxplot of Cost of Car Crash Above Median ~ Binary Predictors",
y="Percent", x="Predictor")
```



correlations

```
summary_positive <- describe(training_normalized[training_normalized$TARGET_F
LAG==1,c(target_bin,inputs)])[,c("mean","n")]

## Warning in FUN(newX[, i], ...): no non-missing arguments to min; returning
## Inf

## Warning in FUN(newX[, i], ...): no non-missing arguments to max; returning
## -Inf

summary_negative <- describe(training_normalized[training_normalized$TARGET_F
LAG==0,c(target_bin,inputs)])[,c("mean","n")]
```

```
## Warning in FUN(newX[, i], ...): no non-missing arguments to min; returning
## Inf

## Warning in FUN(newX[, i], ...): no non-missing arguments to max; returning
## -Inf

summary_by_target <- merge(summary_positive,summary_negative,by="row.names")
colnames(summary_by_target) <- c("Variable","Above Median Cost of Crash - Avg
","Above Median Cost of Crash - n","Below Median Cost of Crash - Avg", "Below
Median Cost of Crash - n")
summary_by_target$delta <- abs(summary_by_target[,"Above Median Cost of Crash
- Avg"]-summary_by_target[,"Below Median Cost of Crash - Avg"])

kable(summary_by_target[order(-summary_by_target$delta),])
```

	Variable	Above Median Cost of Crash - Avg	Above Median Cost of Crash - n	Below Median Cost of Crash - Avg	Below Median Cost of Crash - n	delta
2	BLUEBOOK	-0.1280176	1076	-0.2243655	1077	0.0963479
9	CLM_FREQ	0.3165175	1076	0.4083206	1077	0.0918031
38	YOJ	-0.0763340	1076	-0.1521789	1077	0.0758448
29	MVR_PTS	0.3946086	1076	0.3361865	1077	0.0584221
14	HOMEKIDS	0.2182588	1076	0.1681238	1077	0.0501350
1	AGE	-0.1484935	1076	-0.1968988	1077	0.0484052
36	TIF	-0.1255575	1076	-0.1488946	1077	0.0233371
4	CAR_PANEL_TRUCK_BIN	0.0929368	1076	0.0724234	1077	0.0205134
23	JOB_HOME_MAKER_BIN	0.0734201	1076	0.0937790	1077	0.0203589
17	IS_MALE_BIN	0.4563197	1076	0.4363974	1077	0.0199223
34	REVOKED_BIN	0.2156134	1076	0.1959146	1077	0.0196988
3	CAR_AGE	-0.1712427	1076	-0.1522632	1077	0.0189796
16	IS_COMMERCIAL_BIN	0.4944238	1076	0.4781801	1077	0.0162437
26	JOB_PROFESSIONAL_BIN	0.1217472	1076	0.1077066	1077	0.0140406
15	INCOME	-0.2010833	1076	-0.1875439	1077	0.0135394
30	NON_HOMEOWNER_BIN	0.4423792	1076	0.4558960	1077	0.0135168

10	ED_BACHELORS_BIN	0.2369888	1076	0.2488394	1077	0.0118505
11	ED_HS_BIN	0.3624535	1076	0.3741876	1077	0.0117340
6	CAR_SPORTS_CAR_BIN	0.1356877	1076	0.1467038	1077	0.0110161
20	JOB_BLUE_COLLAR_BIN	0.2899628	1076	0.2989786	1077	0.0090158
7	CAR_SUV_BIN	0.3104089	1076	0.3194058	1077	0.0089968
12	ED_MASTERS_BIN	0.1561338	1076	0.1476323	1077	0.0085015
25	JOB_MANAGERIAL_BIN	0.0678439	1076	0.0594243	1077	0.0084195
13	ED_PHD_BIN	0.0622677	1076	0.0538533	1077	0.0084144
19	IS_URBAN_BIN	0.9507435	1076	0.9424327	1077	0.0083108
37	TRAVTIME	0.0900572	1076	0.0830926	1077	0.0069646
5	CAR_PICKUP_BIN	0.2091078	1076	0.2024141	1077	0.0066937
31	OLDCLAIM	0.2347958	1076	0.2401051	1077	0.0053093
28	KIDSDRIV	0.1708870	1076	0.1758964	1077	0.0050095
21	JOB_CLERICAL_BIN	0.1747212	1076	0.1699164	1077	0.0048048
32	PARENT1_BIN	0.2230483	1076	0.2191272	1077	0.0039211
27	JOB_STUDENT_BIN	0.1254647	1076	0.1216342	1077	0.0038305
8	CAR_VAN_BIN	0.0947955	1076	0.0919220	1077	0.0028735
24	JOB_LAWYER_BIN	0.0697026	1076	0.0724234	1077	0.0027208
33	RED_CAR_BIN	0.2853160	1076	0.2869081	1077	0.0015921
18	IS_SINGLE_BIN	0.5102230	1076	0.5116063	1077	0.0013833
22	JOB_DOCTOR_BIN	0.0139405	1076	0.0129991	1077	0.0009414
35	TARGET_FLAG	NaN	1076	NaN	1077	NaN

TRAINIG DATASETS

NEED TO:

- split datasets
- run models

```
library(caTools)

train_rows <- sample.split(training_trans$TARGET_FLAG, SplitRatio=0.7)
training_trans_model_bin <- training_trans[train_rows,]
training_trans_eval_bin <- training_trans[-train_rows,]
```

3. BUILD MODELS (25 Points)

Using the training data set, build at least two different multiple linear regression models and three different binary logistic regression models, using different variables (or the same variables with different transformations). You may select the variables manually, use an approach such as Forward or Stepwise, use a different approach such as trees, or use a combination of techniques. Describe the techniques you used. If you manually selected a variable for inclusion into the model or exclusion into the model, indicate why this was done.

Be sure to explain how you can make inferences from the model, as well as discuss other relevant model output. Discuss the coefficients in the models, do they make sense? Are you keeping the model even though it is counter intuitive? Why? The boss needs to know.

MODEL 1.

MLR Full model, all variables, flag + amt

The flag one looks okay here, the amt one doesn't seem to work so well.

```
training_target_amt <- training_trans[training_trans$TARGET_FLAG==1,]
target_amt_model_all <- glm(TARGET_AMT~.,data=training_target_amt[,c(inputs,t
arget_lm)])
predict1 <- round(predict(target_amt_model_all, training_trans_eval_bin, type
= 'response'), 4)
summary(target_amt_model_all)

##
## Call:
## glm(formula = TARGET_AMT ~ ., data = training_target_amt[, c(inputs,
##     target_lm)])
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
##  -9358   -3202   -1509     480   99501
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)            1.678e+03  2.005e+03   0.837   0.4026
## PARENT1_BIN            2.820e+02  5.885e+02   0.479   0.6318
## NON_HOMEOWNER_BIN     -5.013e+02  4.423e+02  -1.133   0.2572
## IS_SINGLE_BIN          8.154e+02  5.011e+02   1.627   0.1038
## IS_MALE_BIN            1.422e+03  6.550e+02   2.171   0.0301 *
## ED_HS_BIN             -4.171e+02  5.139e+02  -0.812   0.4171
## ED_BACHELORS_BIN       2.283e+02  6.429e+02   0.355   0.7225
## ED_MASTERS_BIN         1.170e+03  1.085e+03   1.078   0.2811
## ED_PHD_BIN             2.335e+03  1.300e+03   1.796   0.0727 .
## JOB_BLUE_COLLAR_BIN    5.893e+02  1.144e+03   0.515   0.6064
## JOB_CLERICAL_BIN       3.944e+02  1.201e+03   0.328   0.7427
## JOB_PROFESSIONAL_BIN   1.118e+03  1.127e+03   0.992   0.3213
## JOB_MANAGERIAL_BIN    -7.462e+02  1.065e+03  -0.700   0.4837
## JOB_LAWYER_BIN         3.325e+02  1.028e+03   0.323   0.7464
## JOB_STUDENT_BIN        4.467e+02  1.276e+03   0.350   0.7264
## JOB_DOCTOR_BIN        -2.142e+03  1.765e+03  -1.213   0.2251
## JOB_HOME_MAKER_BIN     1.733e+02  1.231e+03   0.141   0.8880
```

```
## IS_COMMERCIAL_BIN     4.244e+02  5.220e+02   0.813   0.4163
## CAR_PANEL_TRUCK_BIN  -6.872e+02  9.559e+02  -0.719   0.4722
## CAR_PICKUP_BIN       -5.801e+01  5.970e+02  -0.097   0.9226
## CAR_SPORTS_CAR_BIN    1.092e+03  7.498e+02   1.457   0.1453
## CAR_VAN_BIN           1.796e+01  7.715e+02   0.023   0.9814
## CAR_SUV_BIN           9.234e+02  6.662e+02   1.386   0.1658
## RED_CAR_BIN          -1.832e+02  4.965e+02  -0.369   0.7121
## REVOKED_BIN          -1.120e+03  5.205e+02  -2.151   0.0316 *
## IS_URBAN_BIN          8.840e+01  7.557e+02   0.117   0.9069
## AGE                   2.137e+01  2.132e+01   1.003   0.3161
## YOJ                  -5.061e-02  5.097e+01  -0.001   0.9992
## CAR_AGE              -9.720e+01  4.428e+01  -2.195   0.0283 *
## KIDSDRIV             -1.843e+02  3.181e+02  -0.579   0.5624
## HOMEKIDS              2.322e+02  2.095e+02   1.108   0.2680
## TRAVTIME              1.142e-01  1.115e+01   0.010   0.9918
## TIF                  -1.550e+01  4.281e+01  -0.362   0.7173
## CLM_FREQ             -1.192e+02  1.608e+02  -0.741   0.4587
## MVR_PTS               1.194e+02  6.930e+01   1.723   0.0850 .
## INCOME               -5.203e-03  6.745e-03  -0.771   0.4405
## BLUEBOOK              1.296e-01  3.090e-02   4.195 2.84e-05 ***
## OLDCLAIM              2.640e-02  2.392e-02   1.103   0.2700
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 59123876)
##
##     Null deviance: 1.2903e+11  on 2152  degrees of freedom
## Residual deviance: 1.2505e+11  on 2115  degrees of freedom
## AIC: 44678
##
## Number of Fisher Scoring iterations: 2

model1_amt <- target_amt_model_all
```

MODEL 2.

MLR Full model with log transformation on amt, all variables, amt only

```
training_target_amt$TARGET_AMT <- log(training_target_amt$TARGET_AMT)
target_amt_model_all <- glm(TARGET_AMT~.,data=training_target_amt[,c(inputs,t
arget_lm)])
predict2 <- round(predict(target_amt_model_all, training_trans_eval_bin, type
= 'response'), 4)
summary(target_amt_model_all)

##
## Call:
## glm(formula = TARGET_AMT ~ ., data = training_target_amt[, c(inputs,
##     target_lm)])
##
## Deviance Residuals:
```

```
##     Min       1Q   Median       3Q      Max  
## -4.6590  -0.4065   0.0362   0.4114   3.2775  
## 
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)    
## (Intercept)            7.885e+00  2.108e-01  37.402  < 2e-16 ***
## PARENT1_BIN            2.580e-02  6.187e-02   0.417 0.676662    
## NON_HOMEOWNER_BIN     -2.968e-02  4.650e-02  -0.638 0.523386    
## IS_SINGLE_BIN          9.331e-02  5.268e-02   1.771 0.076690 .  
## IS_MALE_BIN            9.370e-02  6.886e-02   1.361 0.173723    
## ED_HS_BIN              7.738e-03  5.403e-02   0.143 0.886132    
## ED_BACHELORS_BIN      -2.683e-02  6.759e-02  -0.397 0.691487    
## ED_MASTERS_BIN         1.560e-01  1.141e-01   1.368 0.171603    
## ED_PHD_BIN             2.553e-01  1.367e-01   1.868 0.061936 .  
## JOB_BLUE_COLLAR_BIN    6.405e-02  1.203e-01   0.533 0.594336    
## JOB_CLERICAL_BIN       5.322e-02  1.263e-01   0.422 0.673421    
## JOB_PROFESSIONAL_BIN  1.089e-01  1.185e-01   0.919 0.358127    
## JOB_MANAGERIAL_BIN     2.147e-02  1.120e-01   0.192 0.847998    
## JOB_LAWYER_BIN        -1.084e-02  1.081e-01  -0.100 0.920110    
## JOB_STUDENT_BIN        4.543e-02  1.342e-01   0.339 0.734959    
## JOB_DOCTOR_BIN        -2.927e-02  1.855e-01  -0.158 0.874673    
## JOB_HOME_MAKER_BIN    -3.033e-02  1.294e-01  -0.234 0.814712    
## IS_COMMERCIAL_BIN      1.415e-02  5.488e-02   0.258 0.796551    
## CAR_PANEL_TRUCK_BIN   -2.814e-03  1.005e-01  -0.028 0.977664    
## CAR_PICKUP_BIN         2.678e-02  6.277e-02   0.427 0.669627    
## CAR_SPORTS_CAR_BIN     5.738e-02  7.882e-02   0.728 0.466746    
## CAR_VAN_BIN           -1.563e-02  8.110e-02  -0.193 0.847171    
## CAR_SUV_BIN            9.287e-02  7.003e-02   1.326 0.184978    
## RED_CAR_BIN            2.248e-02  5.220e-02   0.431 0.666720    
## REVOKED_BIN           -9.881e-02  5.472e-02  -1.806 0.071098 .  
## IS_URBAN_BIN           5.631e-02  7.945e-02   0.709 0.478602    
## AGE                    2.270e-03  2.241e-03   1.013 0.311169    
## YOJ                   -4.977e-03  5.358e-03  -0.929 0.353098    
## CAR_AGE               -2.420e-03  4.655e-03  -0.520 0.603255    
## KIDSDRIV              -3.476e-02  3.344e-02  -1.039 0.298764    
## HOMEKIDS               2.626e-02  2.203e-02   1.192 0.233437    
## TRAVTIME              -3.735e-04  1.172e-03  -0.319 0.750069    
## TIF                   -2.080e-03  4.501e-03  -0.462 0.644061    
## CLM_FREQ              -3.830e-02  1.691e-02  -2.265 0.023610 *  
## MVR_PTS                1.547e-02  7.285e-03   2.124 0.033815 *  
## INCOME                -1.353e-06  7.091e-07  -1.908 0.056496 .  
## BLUEBOOK               1.256e-05  3.248e-06   3.865 0.000114 ***
## OLDCLAIM               4.957e-06  2.515e-06   1.971 0.048871 *  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for gaussian family taken to be 0.6534742)
## 
##     Null deviance: 1420.9  on 2152  degrees of freedom
## Residual deviance: 1382.1  on 2115  degrees of freedom
```

```
## AIC: 5233.6
##
## Number of Fisher Scoring iterations: 2
```

```
model2_amt <- target_amt_model_all
```

Model 3.

Manually remove variables from model 1 that weren't significant for flag. And try a version for amt that only has a few variables.

```
inputs_manual_amt <- inputs[c(4,24,28,36)]
training_target_amt <- training_trans[training_trans$TARGET_FLAG==1,]
target_amt_model_all <- glm(TARGET_AMT~.,data=training_target_amt[,c(inputs_m
anual_amt,target_lm)])
predict3 <- round(predict(target_amt_model_all, training_trans_eval_bin, type
= 'response'), 4)
summary(target_amt_model_all)
```

```
##
## Call:
## glm(formula = TARGET_AMT ~ ., data = training_target_amt[, c(inputs_manual
_amt,
##     target_lm)])
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
##  -7862   -3157   -1586     406  100731
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 4273.32491  411.40245  10.387  < 2e-16 ***
## IS_MALE_BIN  620.02474  334.50432   1.854   0.0639 .
## REVOKED_BIN -682.52623  409.37892  -1.667   0.0956 .
## CAR_AGE      -48.79218   31.70237  -1.539   0.1239
## BLUEBOOK       0.11641    0.02079   5.601 2.41e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 58959010)
##
##     Null deviance: 1.2903e+11  on 2152  degrees of freedom
## Residual deviance: 1.2664e+11  on 2148  degrees of freedom
## AIC: 44639
##
## Number of Fisher Scoring iterations: 2
```

```
model3_amt = target_amt_model_all
```

Model 4.

Binary Logistic Regression Baseline with all variables.

```
training_target_flag <- training_trans_model_bin
target_flag_model_all <- glm(TARGET_FLAG~.,data=training_target_flag[,c(input
s,target_bin)],family = binomial(link = "logit"))
predict4 <- round(predict(target_flag_model_all, training_trans_eval_bin, typ
e = 'response'), 4)
summary(target_flag_model_all)

##
## Call:
## glm(formula = TARGET_FLAG ~ ., family = binomial(link = "logit"),
##     data = training_target_flag[, c(inputs, target_bin)])
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4028  -0.7232  -0.4090   0.6381   3.0812
##
## Coefficients:
##                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -4.514e+00  4.088e-01 -11.042  < 2e-16 ***
## PARENT1_BIN           3.931e-01  1.322e-01   2.974 0.002940 **
## NON_HOMEOWNER_BIN     1.486e-01  9.130e-02   1.628 0.103556
## IS_SINGLE_BIN         5.007e-01  9.859e-02   5.079 3.79e-07 ***
## IS_MALE_BIN           1.599e-01  1.320e-01   1.211 0.225749
## ED_HS_BIN             8.110e-02  1.127e-01   0.720 0.471687
## ED_BACHELORS_BIN     -3.976e-01  1.368e-01  -2.907 0.003653 **
## ED_MASTERS_BIN       -4.090e-01  2.143e-01  -1.909 0.056319 .
## ED_PHD_BIN           -2.769e-01  2.563e-01  -1.080 0.279965
## JOB_BLUE_COLLAR_BIN   4.228e-01  2.231e-01   1.895 0.058135 .
## JOB_CLERICAL_BIN      4.906e-01  2.355e-01   2.084 0.037181 *
## JOB_PROFESSIONAL_BIN  2.389e-01  2.147e-01   1.113 0.265786
## JOB_MANAGERIAL_BIN   -4.588e-01  2.081e-01  -2.205 0.027462 *
## JOB_LAWYER_BIN        3.250e-01  2.006e-01   1.620 0.105163
## JOB_STUDENT_BIN       3.954e-01  2.558e-01   1.545 0.122241
## JOB_DOCTOR_BIN       -4.221e-01  3.198e-01  -1.320 0.186856
## JOB_HOME_MAKER_BIN    5.916e-01  2.475e-01   2.390 0.016838 *
## IS_COMMERCIAL_BIN     6.896e-01  1.088e-01   6.340 2.30e-10 ***
## CAR_PANEL_TRUCK_BIN   5.050e-01  1.927e-01   2.620 0.008788 **
## CAR_PICKUP_BIN        6.027e-01  1.178e-01   5.117 3.10e-07 ***
## CAR_SPORTS_CAR_BIN    9.947e-01  1.533e-01   6.489 8.63e-11 ***
## CAR_VAN_BIN           5.877e-01  1.503e-01   3.912 9.17e-05 ***
## CAR_SUV_BIN           7.034e-01  1.310e-01   5.370 7.87e-08 ***
## RED_CAR_BIN           1.638e-02  1.020e-01   0.161 0.872377
## REVOKED_BIN           8.819e-01  1.088e-01   8.104 5.33e-16 ***
## IS_URBAN_BIN          2.315e+00  1.337e-01  17.315  < 2e-16 ***
## AGE                   1.089e-03  4.824e-03   0.226 0.821447
## YOJ                  -1.674e-02  1.040e-02  -1.609 0.107689
```

```
## CAR_AGE               4.027e-03  9.055e-03   0.445 0.656502
## KIDSDRIV              3.742e-01  7.325e-02   5.108 3.26e-07 ***
## HOMEKIDS              4.658e-02  4.435e-02   1.050 0.293500
## TRAVTIME              1.591e-02  2.300e-03   6.916 4.65e-12 ***
## TIF                  -5.046e-02  8.772e-03  -5.752 8.81e-09 ***
## CLM_FREQ              2.065e-01  3.443e-02   5.999 1.99e-09 ***
## MVR_PTS               1.075e-01  1.636e-02   6.573 4.93e-11 ***
## INCOME               -3.555e-06  1.271e-06  -2.798 0.005147 **
## BLUEBOOK             -2.271e-05  6.260e-06  -3.627 0.000286 ***
## OLDCLAIM             -1.451e-05  4.966e-06  -2.922 0.003479 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 6592.6  on 5712  degrees of freedom
## Residual deviance: 5161.6  on 5675  degrees of freedom
## AIC: 5237.6
##
## Number of Fisher Scoring iterations: 5
```

```
model4_flag = target_flag_model_all
```

Model 5.

```
inputs_manual_flag <- inputs[-c(4,5,8,9,11,13,14,15,23,26,28,30)]
target_flag_model_all <- glm(TARGET_FLAG~.,data=training_target_flag[,c(input
s_manual_flag,target_bin)],family = binomial(link = "logit"))
predict5 <- round(predict(target_flag_model_all, training_trans_eval_bin, typ
e = 'response'), 4)
summary(target_flag_model_all)
```

```
##
## Call:
## glm(formula = TARGET_FLAG ~ ., family = binomial(link = "logit"),
##     data = training_target_flag[, c(inputs_manual_flag, target_bin)])
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4553  -0.7297  -0.4179   0.6514   3.0654
##
## Coefficients:
##                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -3.720e+00  2.340e-01 -15.902  < 2e-16 ***
## PARENT1_BIN           4.843e-01  1.130e-01   4.287 1.81e-05 ***
## NON_HOMEOWNER_BIN     1.586e-01  8.568e-02   1.851 0.064218 .
## IS_SINGLE_BIN         4.355e-01  9.207e-02   4.731 2.24e-06 ***
## ED_BACHELORS_BIN     -3.435e-01  8.457e-02  -4.062 4.87e-05 ***
## ED_MASTERS_BIN       -3.554e-01  1.018e-01  -3.493 0.000477 ***
## JOB_CLERICAL_BIN      1.774e-01  1.078e-01   1.646 0.099799 .
## JOB_MANAGERIAL_BIN   -7.184e-01  1.296e-01  -5.544 2.96e-08 ***
```

```
## JOB_HOME_MAKER_BIN   2.292e-01  1.463e-01   1.566 0.117324    
## IS_COMMERCIAL_BIN    8.078e-01  8.885e-02   9.091  < 2e-16 ***
## CAR_PANEL_TRUCK_BIN  4.510e-01  1.710e-01   2.638 0.008339 ** 
## CAR_PICKUP_BIN       5.221e-01  1.149e-01   4.544 5.51e-06 ***
## CAR_SPORTS_CAR_BIN   8.700e-01  1.265e-01   6.880 5.97e-12 ***
## CAR_VAN_BIN          5.465e-01  1.421e-01   3.844 0.000121 ***
## CAR_SUV_BIN          5.794e-01  1.012e-01   5.725 1.04e-08 ***
## REVOKED_BIN          8.875e-01  1.083e-01   8.197 2.45e-16 ***
## IS_URBAN_BIN         2.282e+00  1.334e-01  17.107  < 2e-16 ***
## YOJ                 -1.962e-02  9.361e-03  -2.096 0.036087 *  
## KIDSDRIV             4.170e-01  6.572e-02   6.346 2.22e-10 ***
## TRAVTIME             1.605e-02  2.286e-03   7.022 2.19e-12 ***
## TIF                 -5.051e-02  8.740e-03  -5.780 7.49e-09 ***
## CLM_FREQ             2.000e-01  3.422e-02   5.843 5.13e-09 ***
## MVR_PTS              1.079e-01  1.627e-02   6.630 3.36e-11 ***
## INCOME              -6.091e-06  1.080e-06  -5.638 1.72e-08 ***
## BLUEBOOK            -2.767e-05  5.585e-06  -4.954 7.27e-07 ***
## OLDCLAIM            -1.393e-05  4.938e-06  -2.821 0.004780 ** 
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 6592.6  on 5712  degrees of freedom
## Residual deviance: 5191.5  on 5687  degrees of freedom
## AIC: 5243.5
## 
## Number of Fisher Scoring iterations: 5
```

```
model5_flag = target_flag_model_all
```

Model 6.

```
stepwise_flag_model <- glm(TARGET_FLAG~.,data=training_target_flag[,c(inputs,
target_bin)], family = binomial(link = "probit"))

backward <- step(stepwise_flag_model, trace = 0)
predict6 <- round(predict(backward,training_trans_eval_bin , type = 'response
'), 4)
summary(backward)
```

```
## 
## Call:
## glm(formula = TARGET_FLAG ~ PARENT1_BIN + NON_HOMEOWNER_BIN + 
##     IS_SINGLE_BIN + ED_BACHELORS_BIN + ED_MASTERS_BIN + JOB_BLUE_COLLAR_BI
N + 
##     JOB_CLERICAL_BIN + JOB_MANAGERIAL_BIN + JOB_STUDENT_BIN + 
##     JOB_DOCTOR_BIN + IS_COMMERCIAL_BIN + CAR_PANEL_TRUCK_BIN + 
##     CAR_PICKUP_BIN + CAR_SPORTS_CAR_BIN + CAR_VAN_BIN + CAR_SUV_BIN + 
##     REVOKED_BIN + IS_URBAN_BIN + YOJ + KIDSDRIV + HOMEKIDS + 
##     TRAVTIME + TIF + CLM_FREQ + MVR_PTS + INCOME + BLUEBOOK + 
```

```
##     OLDCLAIM, family = binomial(link = "probit"), data = training_target_f
lag[,
##     c(inputs, target_bin)])
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2596  -0.7424  -0.4143   0.7025   3.4294
##
## Coefficients:
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -2.300e+00  1.344e-01 -17.112  < 2e-16 ***
## PARENT1_BIN           1.810e-01  7.610e-02   2.378 0.017390 *
## NON_HOMEOWNER_BIN     9.678e-02  5.388e-02   1.796 0.072474 .
## IS_SINGLE_BIN         3.238e-01  5.770e-02   5.612 2.00e-08 ***
## ED_BACHELORS_BIN     -1.398e-01  5.060e-02  -2.763 0.005729 **
## ED_MASTERS_BIN       -1.204e-01  6.570e-02  -1.833 0.066793 .
## JOB_BLUE_COLLAR_BIN   1.456e-01  6.610e-02   2.202 0.027648 *
## JOB_CLERICAL_BIN      1.917e-01  7.120e-02   2.693 0.007080 **
## JOB_MANAGERIAL_BIN   -4.053e-01  7.194e-02  -5.634 1.76e-08 ***
## JOB_STUDENT_BIN       1.884e-01  9.144e-02   2.061 0.039342 *
## JOB_DOCTOR_BIN       -3.418e-01  1.451e-01  -2.356 0.018465 *
## IS_COMMERCIAL_BIN     3.891e-01  5.737e-02   6.782 1.18e-11 ***
## CAR_PANEL_TRUCK_BIN   3.558e-01  1.017e-01   3.501 0.000464 ***
## CAR_PICKUP_BIN        2.885e-01  6.780e-02   4.256 2.08e-05 ***
## CAR_SPORTS_CAR_BIN    5.802e-01  7.260e-02   7.992 1.33e-15 ***
## CAR_VAN_BIN           3.746e-01  8.187e-02   4.576 4.74e-06 ***
## CAR_SUV_BIN           4.172e-01  5.785e-02   7.211 5.57e-13 ***
## REVOKED_BIN           4.351e-01  6.445e-02   6.752 1.46e-11 ***
## IS_URBAN_BIN          1.301e+00  6.881e-02  18.905  < 2e-16 ***
## YOJ                  -1.687e-02  5.480e-03  -3.078 0.002084 **
## KIDSDRIV              1.973e-01  4.164e-02   4.739 2.14e-06 ***
## HOMEKIDS              5.136e-02  2.362e-02   2.175 0.029662 *
## TRAVTIME              9.415e-03  1.309e-03   7.192 6.40e-13 ***
## TIF                  -2.882e-02  5.024e-03  -5.736 9.69e-09 ***
## CLM_FREQ              9.809e-02  2.008e-02   4.884 1.04e-06 ***
## MVR_PTS               6.384e-02  9.644e-03   6.619 3.61e-11 ***
## INCOME               -2.673e-06  6.391e-07  -4.182 2.89e-05 ***
## BLUEBOOK             -1.355e-05  3.219e-06  -4.209 2.57e-05 ***
## OLDCLAIM             -5.693e-06  2.939e-06  -1.937 0.052708 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 6592.6  on 5712  degrees of freedom
## Residual deviance: 5196.5  on 5684  degrees of freedom
## AIC: 5254.5
##
## Number of Fisher Scoring iterations: 5
```

```
model6_flag <- backward
```

Model 7.

```
stepwise_flag_model2 <- glm(TARGET_FLAG~1,data=training_target_flag[,c(inputs
,target_bin)], family = binomial(link = "probit"))

forward <- step(stepwise_flag_model2, scope = list(lower=formula(stepwise_fla
g_model2), upper=formula(stepwise_flag_model)), direction = "forward", trace
= 0)
predict7 <- round(predict(forward, training_trans_eval_bin ,type = 'response'
), 4)
summary(forward)

##
## Call:
## glm(formula = TARGET_FLAG ~ IS_URBAN_BIN + MVR_PTS + INCOME +
##     IS_COMMERCIAL_BIN + PARENT1_BIN + JOB_MANAGERIAL_BIN + REVOKED_BIN +
##     TRAVTIME + BLUEBOOK + IS_SINGLE_BIN + KIDSDRIV + TIF + CAR_SPORTS_CAR_
BIN +
##     CAR_SUV_BIN + CLM_FREQ + YOJ + JOB_CLERICAL_BIN + JOB_STUDENT_BIN +
##     CAR_VAN_BIN + CAR_PICKUP_BIN + CAR_PANEL_TRUCK_BIN + JOB_BLUE_COLLAR_B
IN +
##     HOMEKIDS + ED_BACHELORS_BIN + OLDCLAIM + JOB_DOCTOR_BIN +
##     ED_MASTERS_BIN + NON_HOMEOWNER_BIN, family = binomial(link = "probit")
,
##     data = training_target_flag[, c(inputs, target_bin)])
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2596  -0.7424  -0.4143   0.7025   3.4294
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -2.300e+00  1.344e-01 -17.112  < 2e-16 ***
## IS_URBAN_BIN         1.301e+00  6.881e-02  18.905  < 2e-16 ***
## MVR_PTS              6.384e-02  9.644e-03   6.619 3.61e-11 ***
## INCOME              -2.673e-06  6.391e-07  -4.182 2.89e-05 ***
## IS_COMMERCIAL_BIN    3.891e-01  5.737e-02   6.782 1.18e-11 ***
## PARENT1_BIN          1.810e-01  7.610e-02   2.378 0.017390 *
## JOB_MANAGERIAL_BIN  -4.053e-01  7.194e-02  -5.634 1.76e-08 ***
## REVOKED_BIN          4.351e-01  6.445e-02   6.752 1.46e-11 ***
## TRAVTIME             9.415e-03  1.309e-03   7.192 6.40e-13 ***
## BLUEBOOK            -1.355e-05  3.219e-06  -4.209 2.57e-05 ***
## IS_SINGLE_BIN        3.238e-01  5.770e-02   5.612 2.00e-08 ***
## KIDSDRIV             1.973e-01  4.164e-02   4.739 2.14e-06 ***
## TIF                 -2.882e-02  5.024e-03  -5.736 9.69e-09 ***
## CAR_SPORTS_CAR_BIN   5.802e-01  7.260e-02   7.992 1.33e-15 ***
## CAR_SUV_BIN          4.172e-01  5.785e-02   7.211 5.57e-13 ***
## CLM_FREQ             9.809e-02  2.008e-02   4.884 1.04e-06 ***
## YOJ                 -1.687e-02  5.480e-03  -3.078 0.002084 **
```

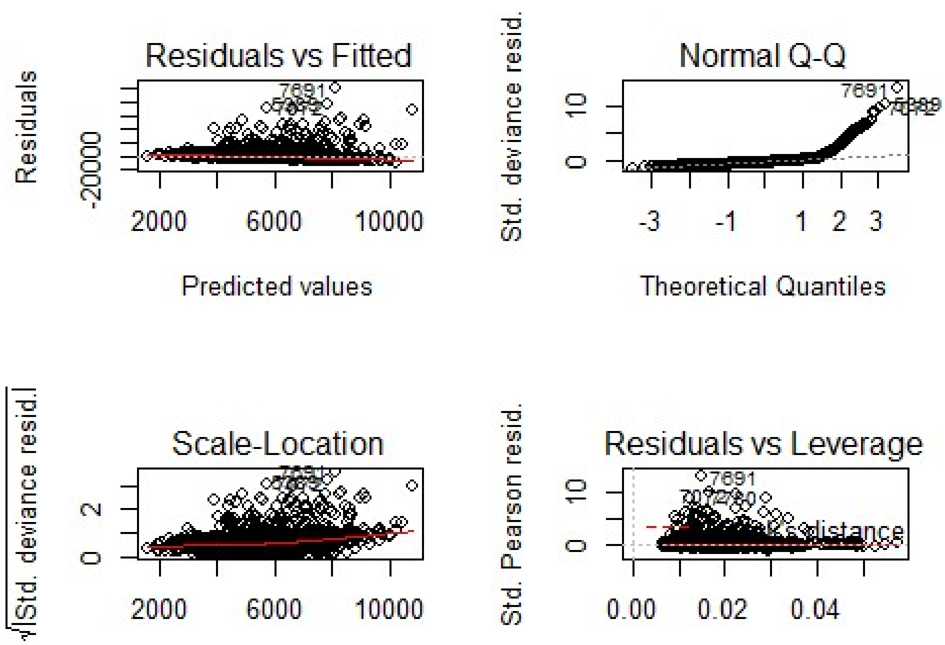
```
## JOB_CLERICAL_BIN     1.917e-01  7.120e-02   2.693 0.007080 **
## JOB_STUDENT_BIN      1.884e-01  9.144e-02   2.061 0.039342 *
## CAR_VAN_BIN          3.746e-01  8.187e-02   4.576 4.74e-06 ***
## CAR_PICKUP_BIN       2.885e-01  6.780e-02   4.256 2.08e-05 ***
## CAR_PANEL_TRUCK_BIN  3.558e-01  1.017e-01   3.501 0.000464 ***
## JOB_BLUE_COLLAR_BIN  1.456e-01  6.610e-02   2.202 0.027648 *
## HOMEKIDS             5.136e-02  2.362e-02   2.175 0.029662 *
## ED_BACHELORS_BIN    -1.398e-01  5.060e-02  -2.763 0.005729 **
## OLDCLAIM            -5.693e-06  2.939e-06  -1.937 0.052708 .
## JOB_DOCTOR_BIN      -3.418e-01  1.451e-01  -2.356 0.018465 *
## ED_MASTERS_BIN      -1.204e-01  6.570e-02  -1.833 0.066793 .
## NON_HOMEOWNER_BIN    9.678e-02  5.388e-02   1.796 0.072474 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 6592.6  on 5712  degrees of freedom
## Residual deviance: 5196.5  on 5684  degrees of freedom
## AIC: 5254.5
##
## Number of Fisher Scoring iterations: 5

model7_flag <- forward
```

4. SELECT MODELS (25 Points)

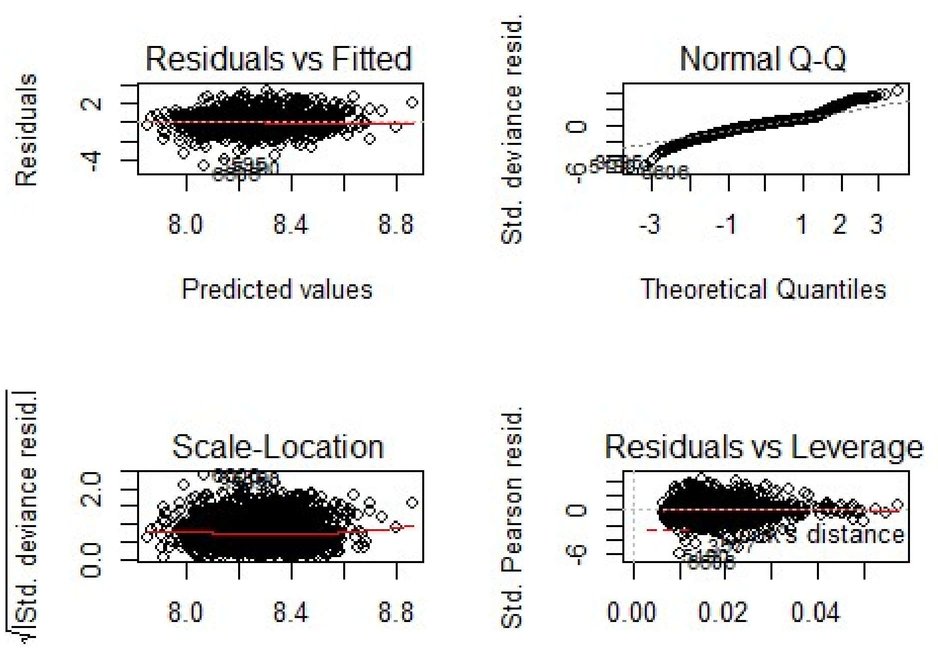
Decide on the criteria for selecting the best multiple linear regression model and the best binary logistic regression model. Will you select models with slightly worse performance if it makes more sense or is more parsimonious? Discuss why you selected your models.

For the multiple linear regression model, will you use a metric such as Adjusted R2, RMSE, etc.? Be sure to explain how you can make inferences from the model, discuss multi-collinearity issues (if any), and discuss other relevant model output. Using the training data set, evaluate the multiple linear regression model based on (a) mean squared error, (b) R2, (c) F-statistic, and (d) residual plots. For the binary logistic regression model, will you use a metric such as log likelihood, AIC, ROC curve, etc.? Using the training data set, evaluate the binary logistic regression model based on (a) accuracy, (b) classification error rate, (c) precision, (d) sensitivity, (e) specificity, (f) F1 score, (g) AUC, and (h) confusion matrix. Make predictions using the evaluation data set.

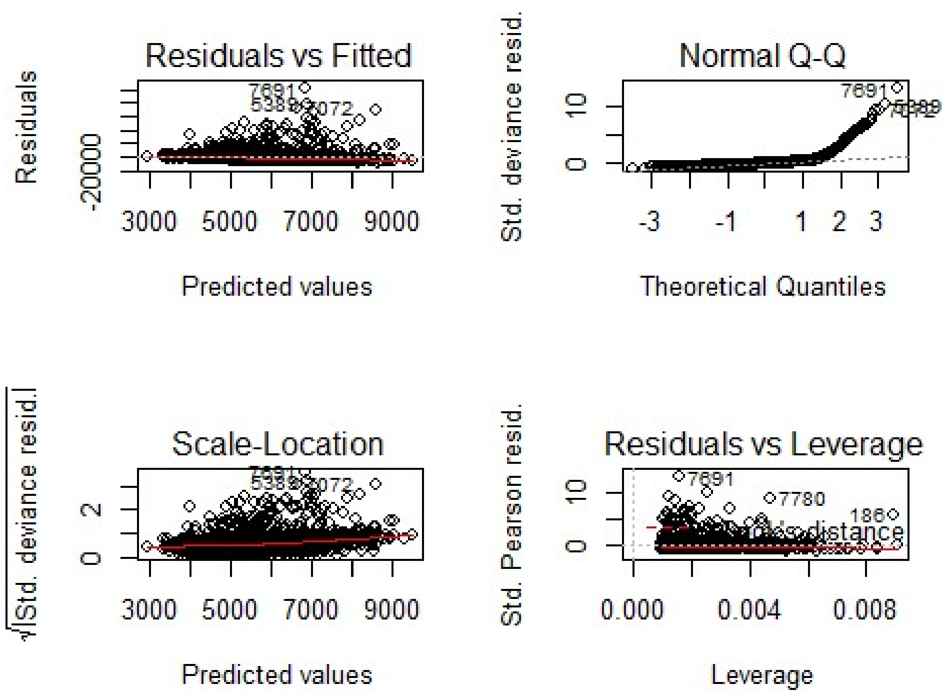
```
par(mfrow=c(2,2))
plot(model1_amt)
```



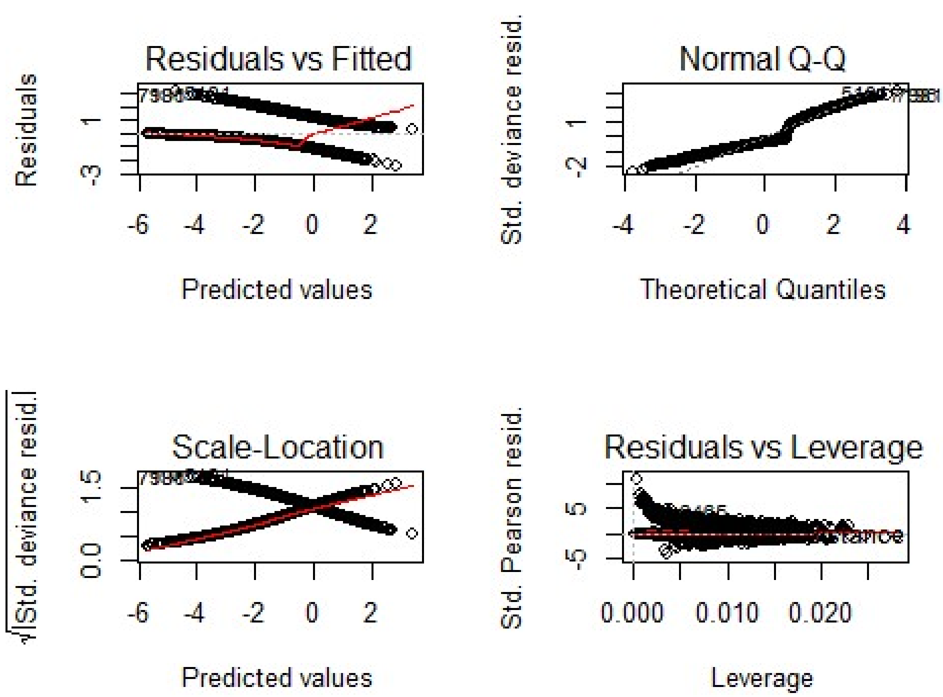

```
plot(model2_amt)
```



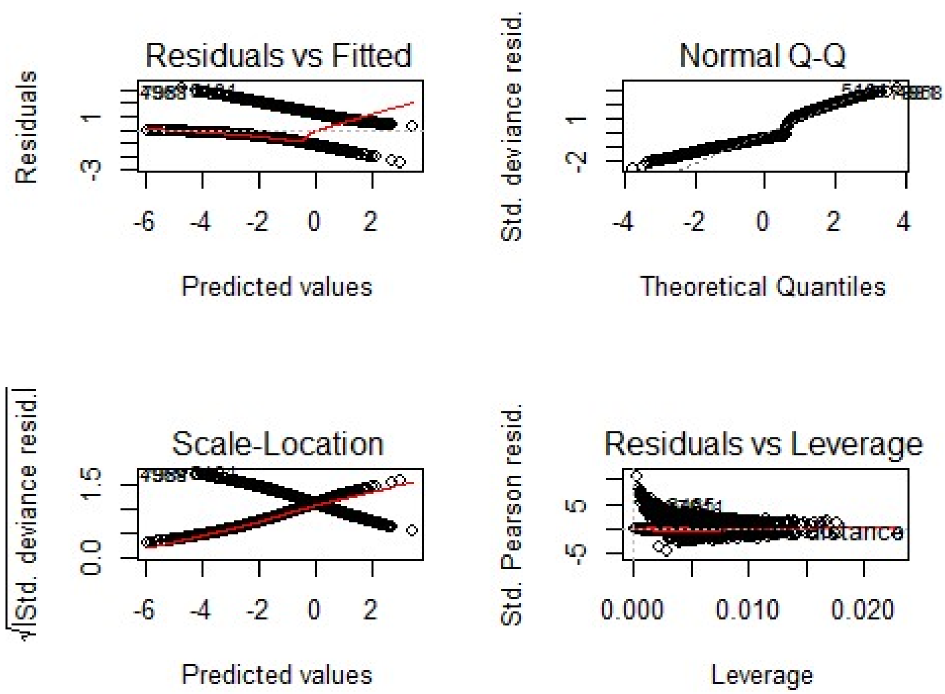

```
plot(model3_amt)
```



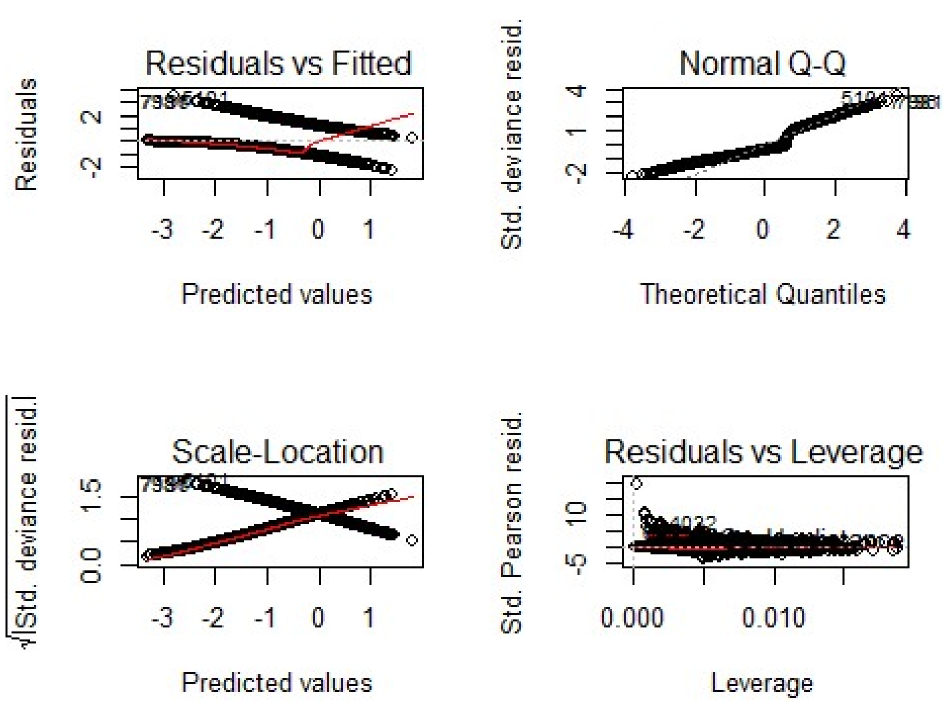

```r
plot(model4_flag)
```



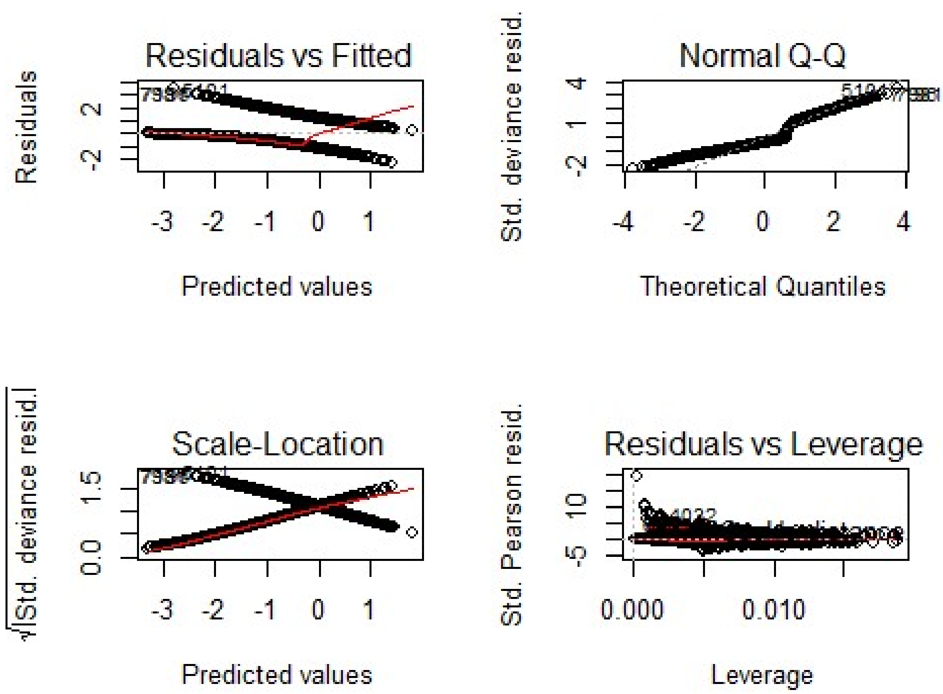

```r
plot(model5_flag)
```




```
plot(model6_flag)
```




```
plot(model7_flag)
```



Function of confusion matrix

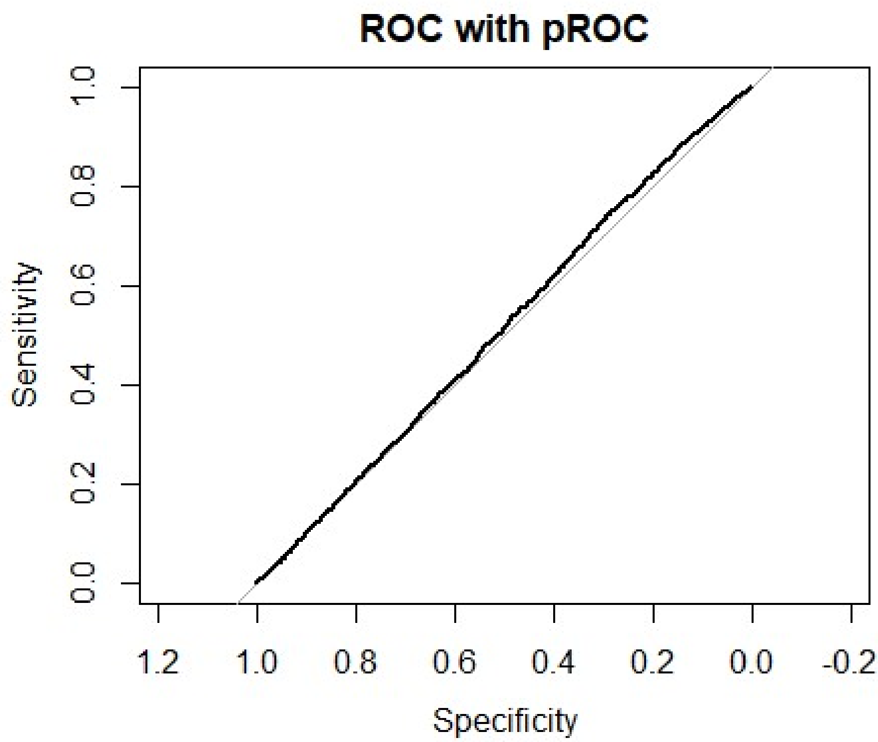
```r
# let's use this helper function that will return all the rates for future ca
lculations
confusion_matrix <- function(d){
  data.frame(tp=nrow(d[d$class==1 & d$scored.class==1,]),
             tn=nrow(d[d$class==0 & d$scored.class==0,]),
             fp=nrow(d[d$class==0 & d$scored.class==1,]),
             fn=nrow(d[d$class==1 & d$scored.class==0,])
  )
}
accuracy<-function(d){
  f <- confusion_matrix(d)
  (f$tp+f$tn)/(f$tp+f$fp+f$tn+f$fn)
}

classification_error_rate<-function(d){
  f <- confusion_matrix(d)
  (f$fp+f$fn)/(f$tp+f$fp+f$tn+f$fn)
}

precision_c<-function(d){
  f <- confusion_matrix(d)
  (f$tp)/(f$tp+f$fp)
}
```

```
sensitivity_c<-function(d){
  f <- confusion_matrix(d)
  (f$tp)/(f$tp+f$fn)
}

specificity_c<-function(d){
  f <- confusion_matrix(d)
  (f$tn)/(f$tn+f$fp)
}


f1_score<-function(d){
  p<- precision_c(d)
  s<- sensitivity_c(d)
  2*p*s/(p+s)
}
```

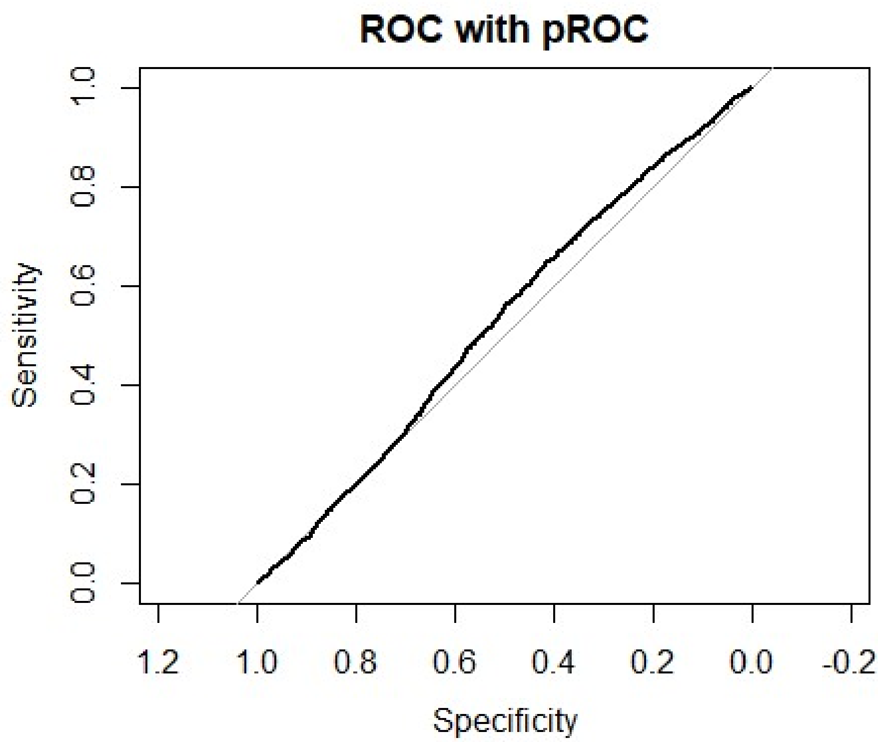
Predictions and Accuracy

```
#predict 1
d<- data.frame(class=training_trans_eval_bin$TARGET_FLAG,scored.class=ifelse(
predict1>0.5,1,0))

confusion_matrix(d)

##     tp tn   fp fn
## 1 2153  0 6007  0

Accuracy <- accuracy(d)
Error <- classification_error_rate(d)
Precision <- precision_c(d)
Sensitivity <- sensitivity_c(d)
Specificity <- specificity_c(d)
F1 <- f1_score(d)

BestFitModel1<- data.frame(Accuracy,Error,Precision,Sensitivity,Specificity,F
1)

require("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
```

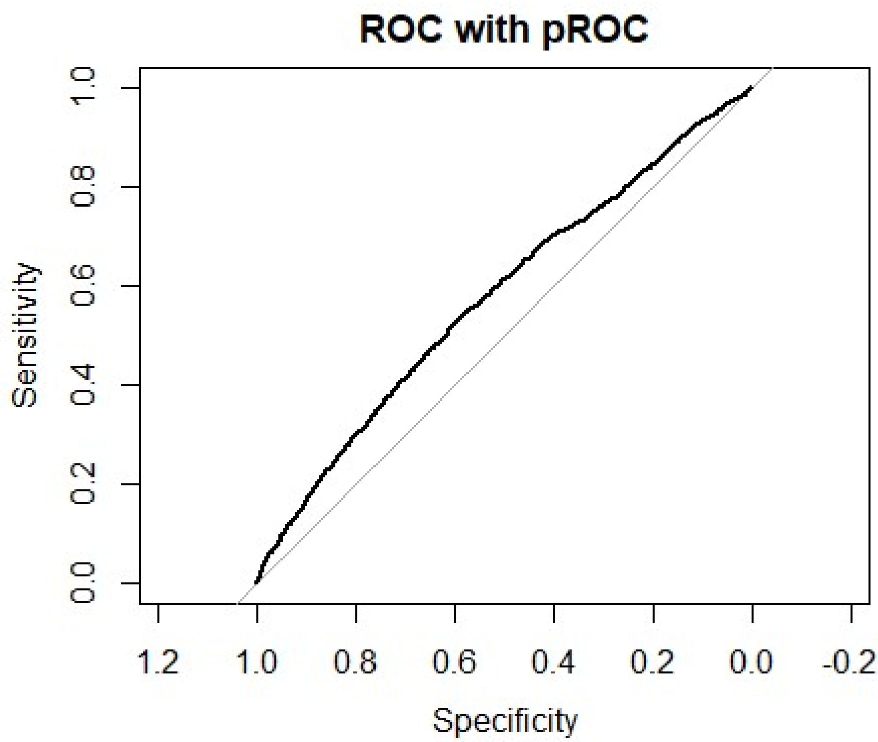
```
d_roc <- roc(training_trans_eval_bin$TARGET_FLAG,predict1)
plot(d_roc, main = "ROC with pROC")
```



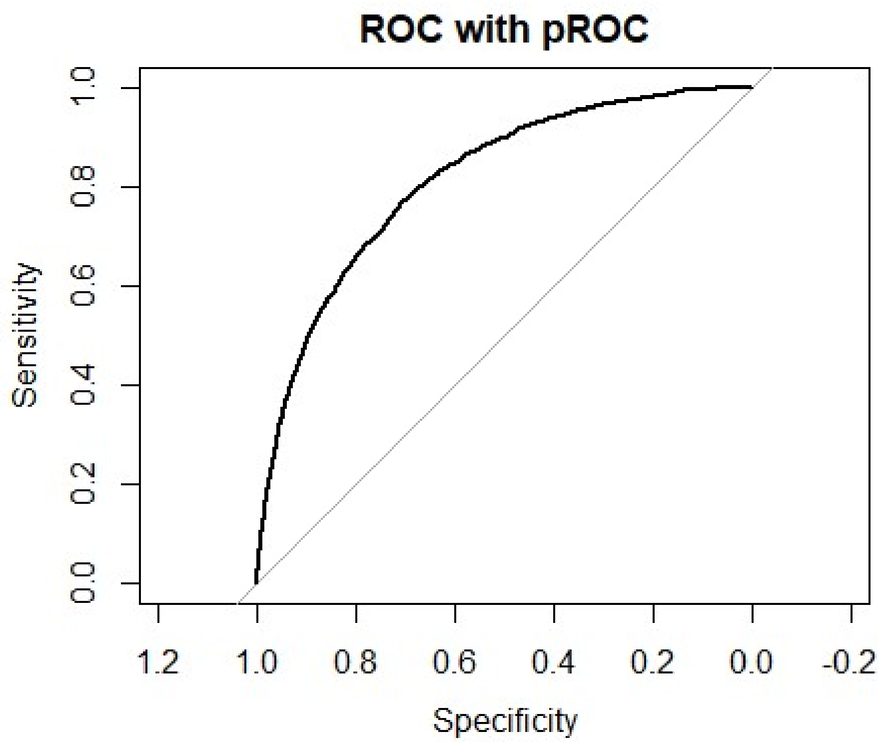

```
#predict 2
d<- data.frame(class=training_trans_eval_bin$TARGET_FLAG,scored.class=ifelse(
predict2>0.5,1,0))

confusion_matrix(d)

##      tp tn   fp fn
## 1 2153  0 6007  0

Accuracy <- accuracy(d)
Error <- classification_error_rate(d)
Precision <- precision_c(d)
Sensitivity <- sensitivity_c(d)
Specificity <- specificity_c(d)
F1 <- f1_score(d)

BestFitModel2<- data.frame(Accuracy,Error,Precision,Sensitivity,Specificity,F
1)

require("pROC")
d_roc <- roc(training_trans_eval_bin$TARGET_FLAG,predict2)
plot(d_roc, main = "ROC with pROC")
```



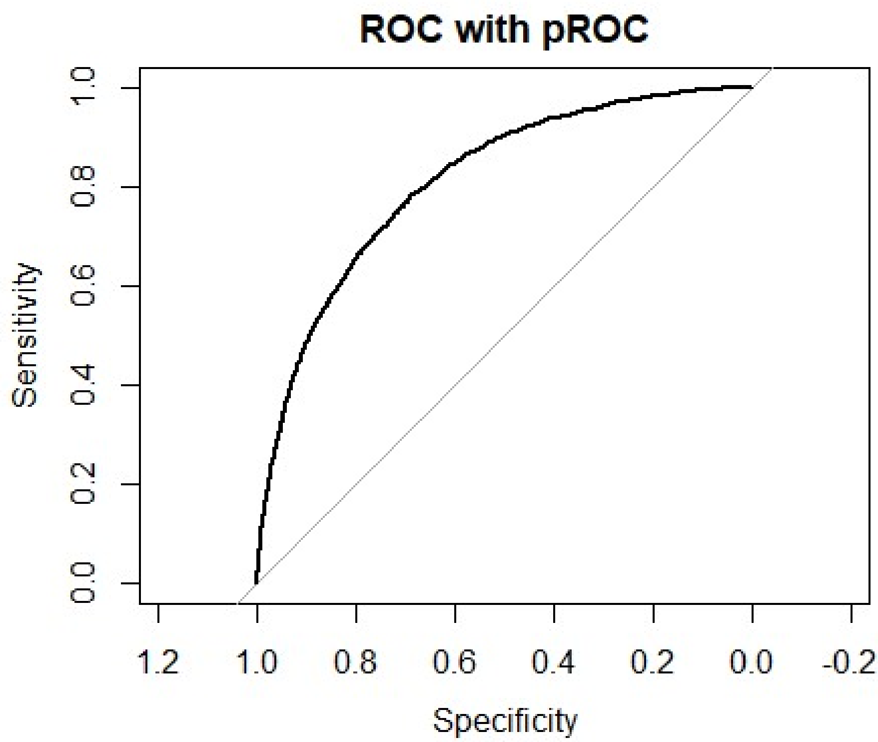

```
#predict 3
d<- data.frame(class=training_trans_eval_bin$TARGET_FLAG,scored.class=ifelse(
predict3>0.5,1,0))

confusion_matrix(d)

##      tp tn   fp fn
## 1 2153  0 6007  0

Accuracy <- accuracy(d)
Error <- classification_error_rate(d)
Precision <- precision_c(d)
Sensitivity <- sensitivity_c(d)
Specificity <- specificity_c(d)
F1 <- f1_score(d)

BestFitModel3<- data.frame(Accuracy,Error,Precision,Sensitivity,Specificity,F
1)

require("pROC")
d_roc <- roc(training_trans_eval_bin$TARGET_FLAG,predict3)
plot(d_roc, main = "ROC with pROC")
```



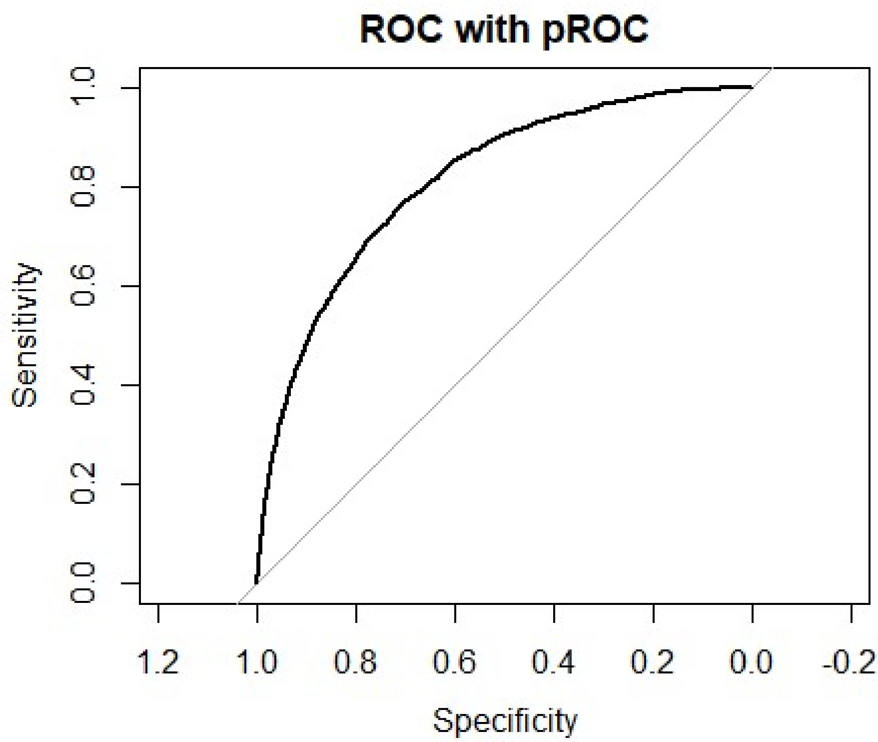

```
#predict 4
d<- data.frame(class=training_trans_eval_bin$TARGET_FLAG,scored.class=ifelse(
predict4>0.5,1,0))

confusion_matrix(d)

##    tp   tn  fp   fn
## 1 892 5568 439 1261

Accuracy <- accuracy(d)
Error <- classification_error_rate(d)
Precision <- precision_c(d)
Sensitivity <- sensitivity_c(d)
Specificity <- specificity_c(d)
F1 <- f1_score(d)

BestFitModel4<- data.frame(Accuracy,Error,Precision,Sensitivity,Specificity,F
1)

require("pROC")
d_roc <- roc(training_trans_eval_bin$TARGET_FLAG,predict4)
plot(d_roc, main = "ROC with pROC")
```



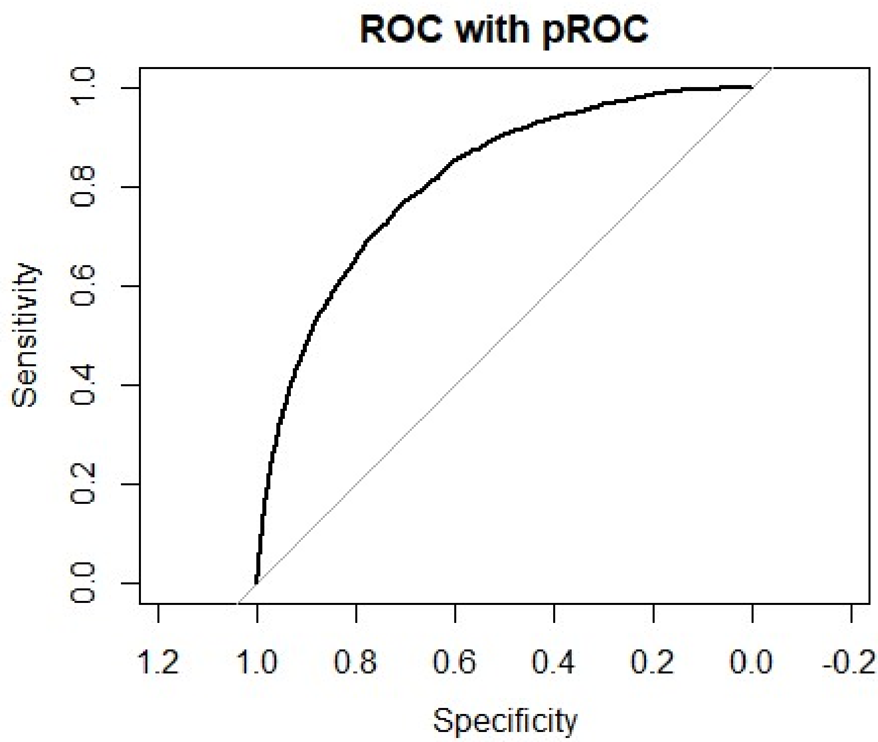

```
#predict 5
d<- data.frame(class=training_trans_eval_bin$TARGET_FLAG,scored.class=ifelse(
predict5>0.5,1,0))

confusion_matrix(d)

##    tp   tn  fp   fn
## 1 869 5587 420 1284

Accuracy <- accuracy(d)
Error <- classification_error_rate(d)
Precision <- precision_c(d)
Sensitivity <- sensitivity_c(d)
Specificity <- specificity_c(d)
F1 <- f1_score(d)

BestFitModel5<- data.frame(Accuracy,Error,Precision,Sensitivity,Specificity,F
1)

require("pROC")
d_roc <- roc(training_trans_eval_bin$TARGET_FLAG,predict5)
plot(d_roc, main = "ROC with pROC")
```




```r
#predict 6
d<- data.frame(class=training_trans_eval_bin$TARGET_FLAG,scored.class=ifelse(
predict6>0.5,1,0))

confusion_matrix(d)

##    tp   tn  fp   fn
## 1 886 5569 438 1267

Accuracy <- accuracy(d)
Error <- classification_error_rate(d)
Precision <- precision_c(d)
Sensitivity <- sensitivity_c(d)
Specificity <- specificity_c(d)
F1 <- f1_score(d)

BestFitModel6<- data.frame(Accuracy,Error,Precision,Sensitivity,Specificity,F
1)

require("pROC")
d_roc <- roc(training_trans_eval_bin$TARGET_FLAG,predict6)
plot(d_roc, main = "ROC with pROC")
```




```
#predict 7
d<- data.frame(class=training_trans_eval_bin$TARGET_FLAG,scored.class=ifelse(
predict7>0.5,1,0))

confusion_matrix(d)

##    tp   tn  fp   fn
## 1 886 5569 438 1267

Accuracy <- accuracy(d)
Error <- classification_error_rate(d)
Precision <- precision_c(d)
Sensitivity <- sensitivity_c(d)
Specificity <- specificity_c(d)
F1 <- f1_score(d)

BestFitModel7<- data.frame(Accuracy,Error,Precision,Sensitivity,Specificity,F
1)

require("pROC")
d_roc <- roc(training_trans_eval_bin$TARGET_FLAG,predict7)
plot(d_roc, main = "ROC with pROC")
```



Compare the Models to choose the best

```
CompareBestFitModel=rbind(BestFitModel1,BestFitModel2,BestFitModel3,BestFitMo
del4,BestFitModel5,BestFitModel6,BestFitModel7)
colnames(CompareBestFitModel)=c("Accuracy","Error","Precision","Sensitivity",
"Specificity","F1")
rownames(CompareBestFitModel)=c("Model1","Model2","Model3","Model4","Model5",
"Model6","Model7")
CompareBestFitModel

##         Accuracy     Error Precision Sensitivity Specificity        F1
## Model1 0.2638480 0.7361520 0.2638480   1.0000000   0.0000000 0.4175313
## Model2 0.2638480 0.7361520 0.2638480   1.0000000   0.0000000 0.4175313
## Model3 0.2638480 0.7361520 0.2638480   1.0000000   0.0000000 0.4175313
## Model4 0.7916667 0.2083333 0.6701728   0.4143056   0.9269186 0.5120551
## Model5 0.7911765 0.2088235 0.6741660   0.4036229   0.9300816 0.5049390
## Model6 0.7910539 0.2089461 0.6691843   0.4115188   0.9270851 0.5096347
## Model7 0.7910539 0.2089461 0.6691843   0.4115188   0.9270851 0.5096347
```

Conclusion

From the above analysis, we can deduce that the AUC (Area Under Curve) for all the three models are very close to 1, which indicate that the model 4 is more specificity, sensitivity and accuracy.