DATA621 - Insurance

Daniel Hong, Mauricio Alarcon, Maxwell Wagner

October 10, 2016

## 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 arenâ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.

1. Mean / Standard Deviation / Median
2. Bar Chart or Box Plot of the data
3. Is the data correlated to the target variable (or to other variables?)
4. Are any of the variables missing and need to be imputed âfixedâ?
5. 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/master/hw04/insurance\_training\_data.csv'),stringsAsFactors = FALSE)  
metadata <- read.csv(url('https://raw.githubusercontent.com/rmalarc/DATA621/master/hw04/insurance-metadata.csv'))  
colnames(metadata) <- c("Variable", "Name Definition", "Theoretical Effect")  
evaluation <- read.csv(url('https://raw.githubusercontent.com/rmalarc/DATA621/master/hw04/insurance-evaluation-data.csv'),stringsAsFactors = FALSE)  
  
kable(metadata)

|  |  |  |
| --- | --- | --- |
| Variable | Name Definition | Theoretical Effect |
| INDEX | Identification Variable (do not use) | None |
| TARGET\_FLAG | Was Car in a crash? 1=YES 0=NO | None |
| TARGET\_AMT | If car was in a crash, what was the cost | None |
| AGE | Age of Driver | Very young people tend to be risky. Maybe very old people also. |
| BLUEBOOK | Value of Vehicle | Unknown effect on probability of collision, but probably effect the payout if there is a crash |
| CAR\_AGE | Vehicle Age | Unknown effect on probability of collision, but probably effect the payout if there is a crash |
| CAR\_TYPE | Type of Car | Unknown effect on probability of collision, but probably effect the payout if there is a crash |
| CAR\_USE | Vehicle Use | Commercial vehicles are driven more, so might increase probability of collision |
| CLM\_FREQ | # Claims (Past 5 Years) | The more claims you filed in the past, the more you are likely to file in the future |
| EDUCATION | Max Education Level | Unknown effect, but in theory more educated people tend to drive more safely |
| HOMEKIDS | # Children at Home | Unknown effect |
| HOME\_VAL | Home Value | In theory, home owners tend to drive more responsibly |
| INCOME | Income | In theory, rich people tend to get into fewer crashes |
| JOB | Job Category | In theory, white collar jobs tend to be safer |
| KIDSDRIV | # Driving Children | When teenagers drive your car, you are more likely to get into crashes |
| MSTATUS | Marital Status | In theory, married people drive more safely |
| MVR\_PTS | Motor Vehicle Record Points | If you get lots of traffic tickets, you tend to get into more crashes |
| OLDCLAIM | Total Claims (Past 5 Years) | If your total payout over the past five years was high, this suggests future payouts will be high |
| PARENT1 | Single Parent | Unknown effect |
| RED\_CAR | A Red Car | Urban legend says that red cars (especially red sports cars) are more risky. Is that true? |
| REVOKED | License Revoked (Past 7 Years) | If your license was revoked in the past 7 years, you probably are a more risky driver. |
| SEX | Gender | Urban legend says that women have less crashes then men. Is that true? |
| TIF | Time in Force | People who have been customers for a long time are usually more safe. |
| TRAVTIME | Distance to Work | Long drives to work usually suggest greater risk |
| URBANICITY | Home/Work Area | Unknown |
| YOJ | Years on Job | People who stay at a job for a long time are usually more safe |

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)  
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

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.

1. Fix missing values (maybe with a Mean or Median value)
2. Create flags to suggest if a variable was missing
3. Transform data by putting it into buckets
4. Mathematical transforms such as log or square root (or use Box-Cox)
5. 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"

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\_MANAGERIAL, 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}  
 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  
}  
  
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)  
  
  
  
 #\* Cap numerical values to their 5/95 percentiles  
 #\* Fill missing values with mean for: AGE, YOJ, CAR\_AGE

### Data Imputations/ Capping

#### Imputations

* Fill missing nummerical values with mean for: AGE, YOJ, CAR\_AGE, INCOME
* Impute missing OLDCLAIM with zeros

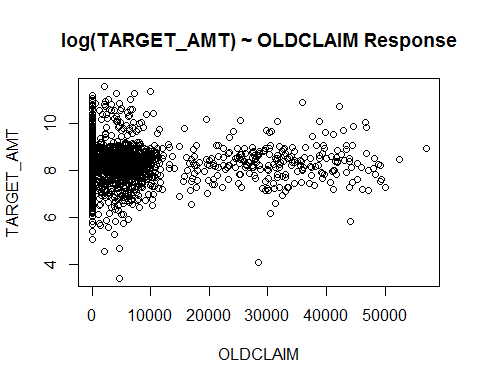
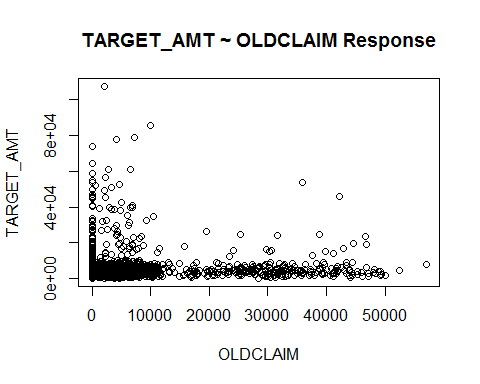
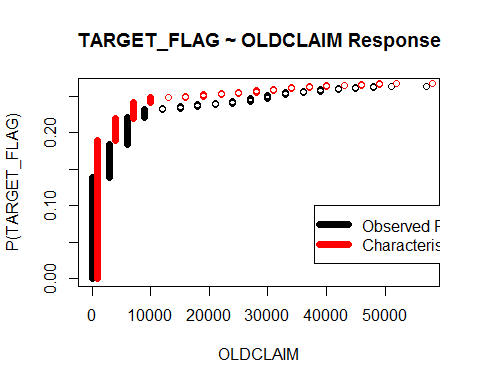
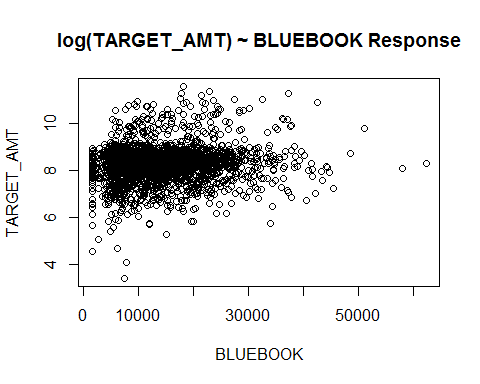
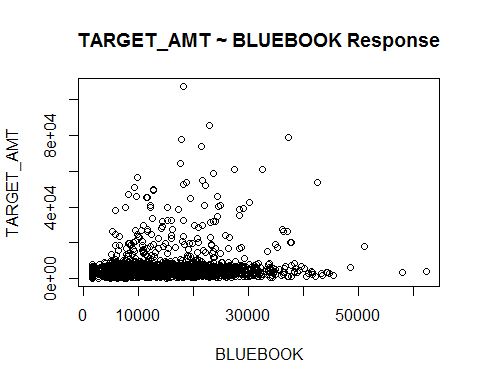
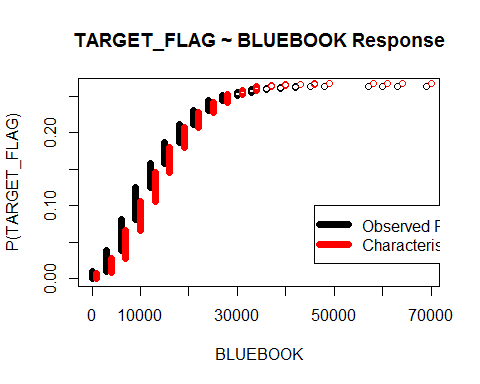
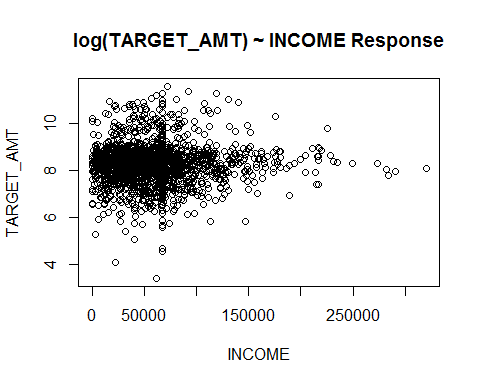
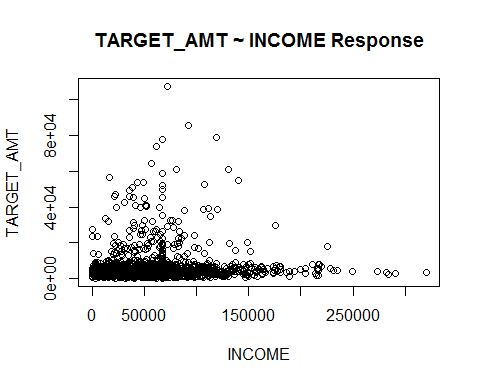
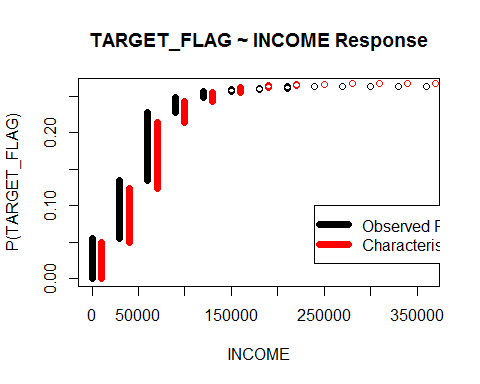
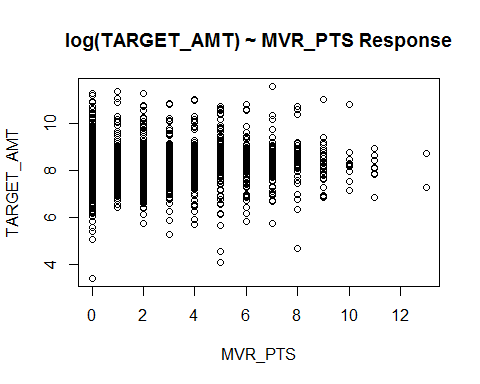
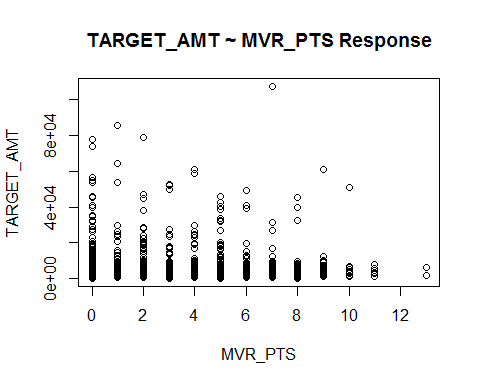
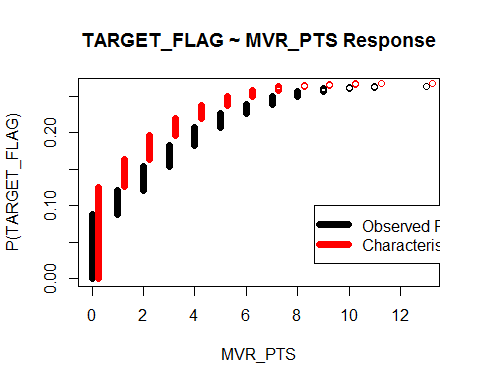
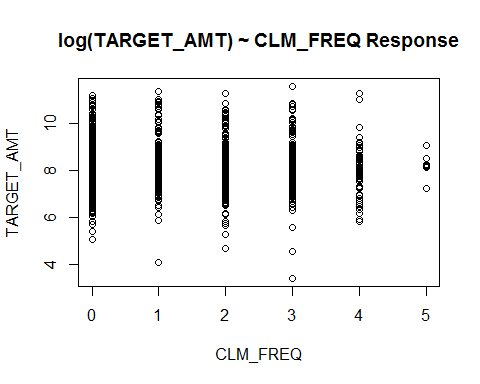
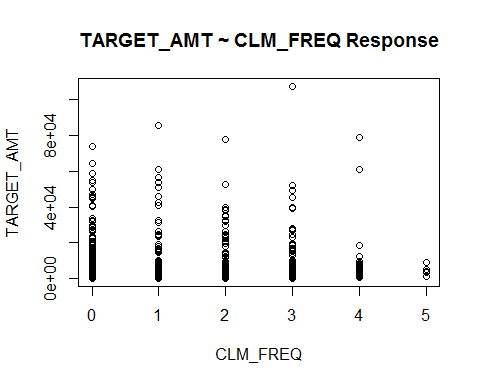
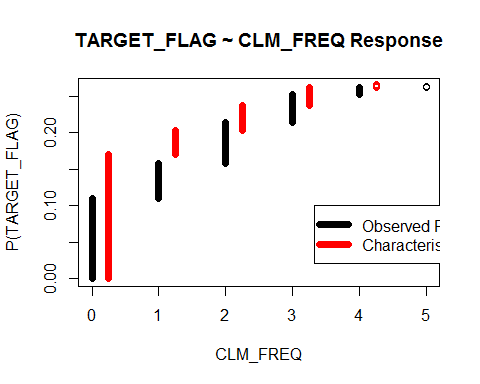
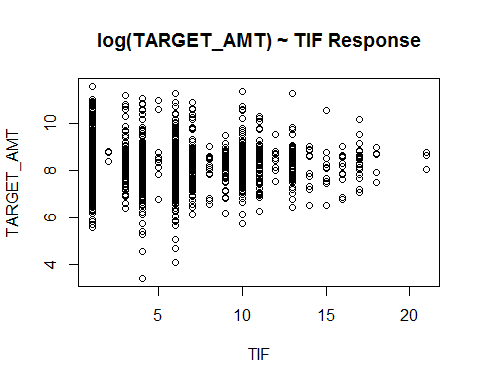
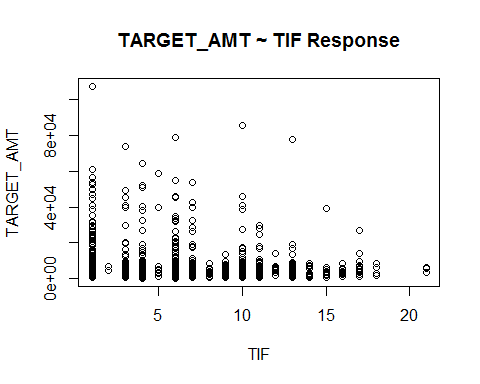
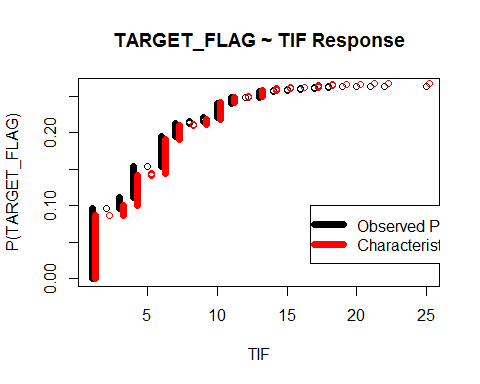
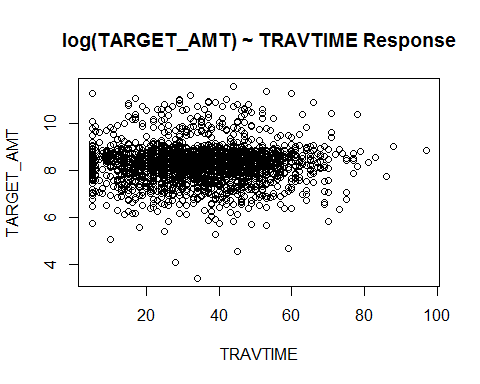
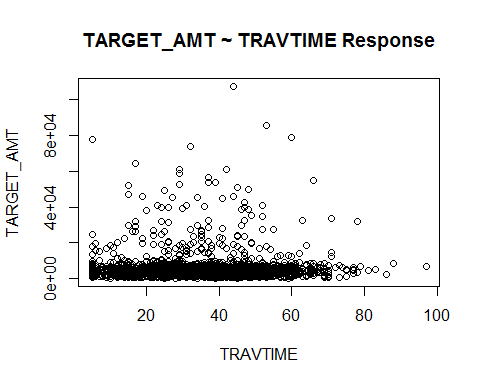
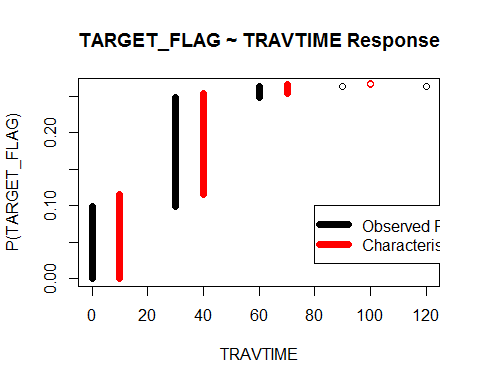
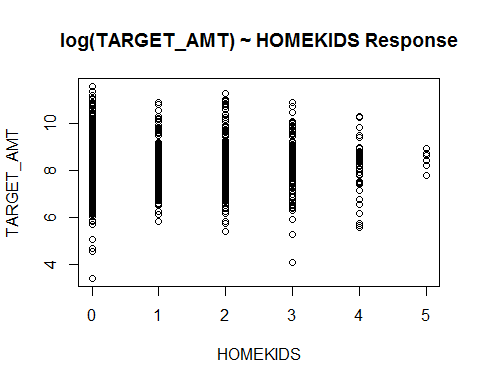
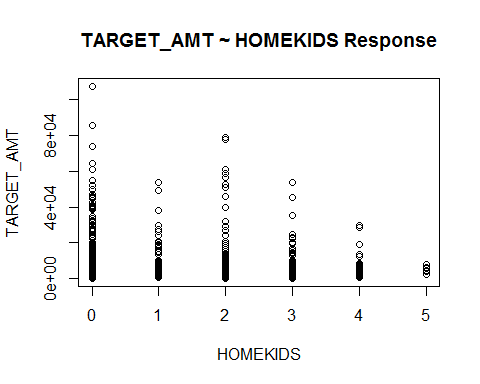
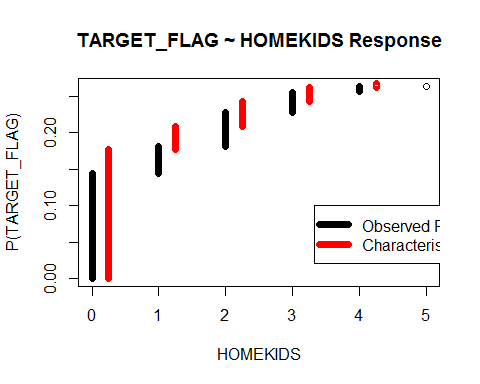
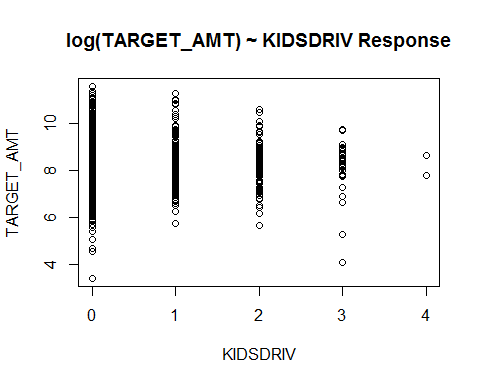
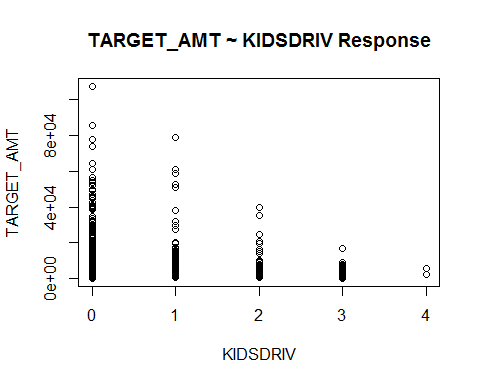
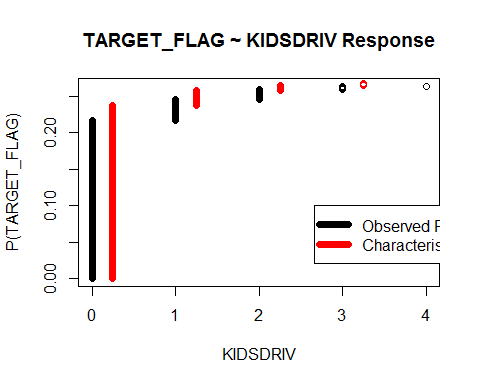
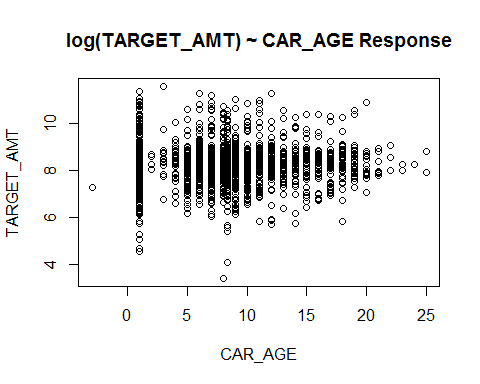
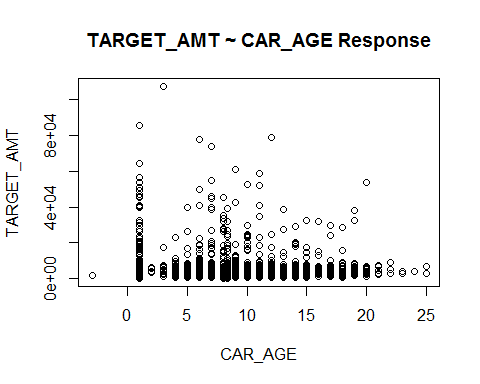
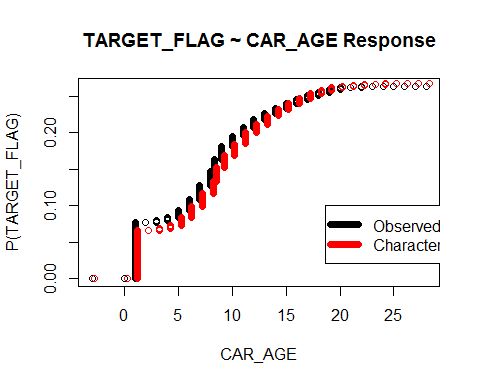
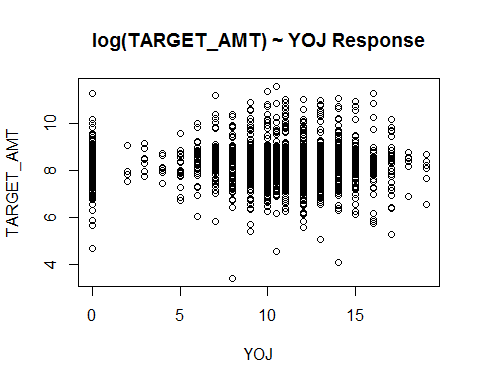
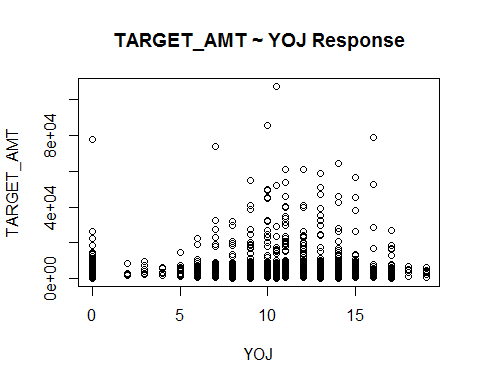
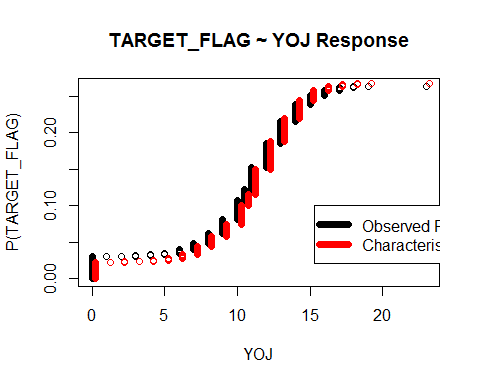
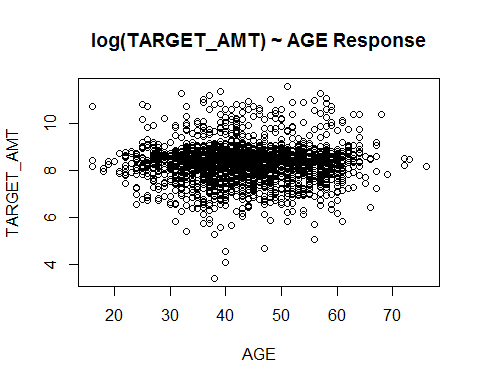
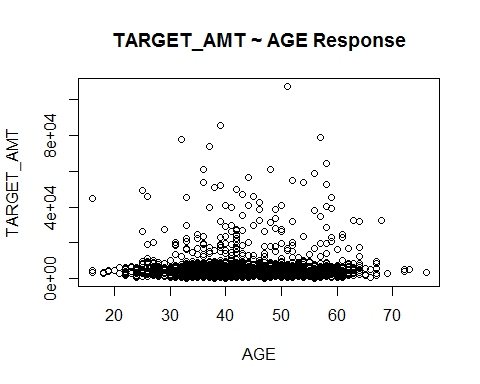
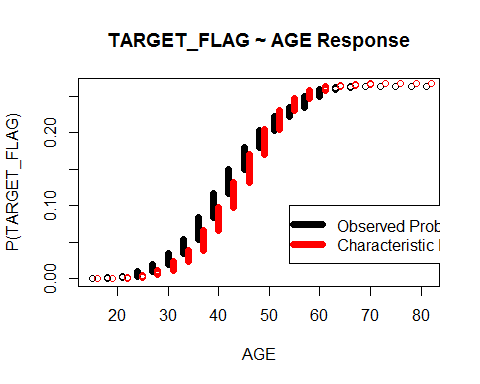
# impute  
impute <- function (d) {  
 d[is.na(d$AGE),]$AGE <- mean(d$AGE,na.rm = TRUE)  
 d[is.na(d$YOJ),]$YOJ <- mean(d$YOJ,na.rm = TRUE)  
 d[is.na(d$CAR\_AGE),]$CAR\_AGE <- mean(d$CAR\_AGE,na.rm = TRUE)  
 d[is.na(d$INCOME),]$INCOME <- mean(d$INCOME,na.rm = TRUE)  
 d[is.na(d$OLDCLAIM),]$OLDCLAIM <- 0  
 d  
}  
training\_trans<-impute(training\_trans)  
evaluation\_trans<-impute(evaluation\_trans)

#### Transformation Analysis Of Numerical Features

##### **TARGET\_FLAG**

Let's see how the TARGET\_FLAG and TARGET\_NUM respond to each of the numerical features

y\_characteristic <- cumsum(rbinom(nrow(training\_trans),1,sum(training\_trans$TARGET\_FLAG)/length(training\_trans$TARGET\_FLAG)))/length(training\_trans$TARGET\_FLAG)  
numResponse <- function(col,d){  
# par(mfrow=c(4,1))  
 data <- d[order(d[,col]),c(col,target\_bin)]  
 x <- data[,col]  
 decimal\_base <- 0.25  
 decimal\_digits<- log10(sum(range(x)\*(c(-1,1))))  
 if (decimal\_digits >1.8) {  
 decimal\_base <- 10^floor(decimal\_digits-1)  
 x<- (x%/%(3\*decimal\_base)\*(3\*decimal\_base))  
 }  
 y <- cumsum(data[,target\_bin])/length(x)#sum(data[,target\_bin])  
# y\_inverse <- (sum(data[,target\_bin])-cumsum(data[,target\_bin]))/sum(data[,target\_bin])  
 plot(y~x, main=paste0("TARGET\_FLAG ~ ",col," Response"),xlab=col, ylab="P(TARGET\_FLAG)")  
 points(x+decimal\_base,y\_characteristic,col="red") # adding inverse so it gives a better visual appreciation of what's going on  
 legend(max(x)\*2/3, 0.1, c("Observed Probability", "Characteristic Binomial"), col = c("black","red"),  
 lty=c(1,1),lwd=7,merge = TRUE)  
  
 data <- d[d$TARGET\_FLAG==1,]  
 data <- data[order(data[,col]),c(col,target\_lm)]  
 x <- data[,col]  
 y <- data[,target\_lm]  
 plot(y~x, main=paste0("TARGET\_AMT ~ ",col," Response"),xlab=col, ylab="TARGET\_AMT")  
  
 x <- data[,col]  
 y <- log(data[,target\_lm])  
 plot(y~x, main=paste0("log(TARGET\_AMT) ~ ",col," Response"),xlab=col, ylab="TARGET\_AMT")  
 c(nrow(data),nrow(d))  
}  
  
  
sapply(inputs\_num,function(x){numResponse(x,training\_trans)})



## AGE YOJ CAR\_AGE KIDSDRIV HOMEKIDS TRAVTIME TIF CLM\_FREQ MVR\_PTS  
## [1,] 2153 2153 2153 2153 2153 2153 2153 2153 2153  
## [2,] 8161 8161 8161 8161 8161 8161 8161 8161 8161  
## INCOME BLUEBOOK OLDCLAIM  
## [1,] 2153 2153 2153  
## [2,] 8161 8161 8161

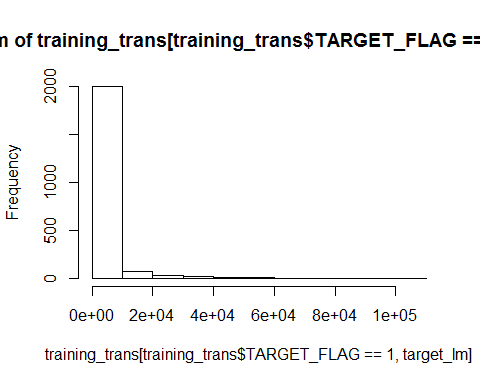
Based on the charts above, we should discretize any variables which don't give a clear sigmoid function. See how beautiful TARGET\_FLAG ~ AGE looks...., if we don't get a sigmoid off the bat, we're never going to get a good logistic regression.

The discretization can be tricky. I would discretize in regions where the bars are about the same size (somewhat homogenous). For instance CAR\_AGE, I would do CAR\_0, CAR\_1, CAR\_2\_4, CAR\_5\_15, CAR\_15\_MORE. This is an interesting variable as it appears to be pseudo-sigmoid, with the exception where the car is 1, perhaps a lot of people have accidents past the first year

I'd break HOMEKIDS into: HK\_0, HK\_1\_3, HK\_4\_MORE

##### **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.

I now see a 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

# 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)

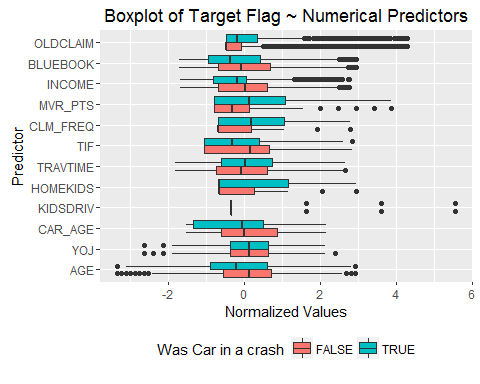
### How are the input values distributed?, do we need to do something about them?

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

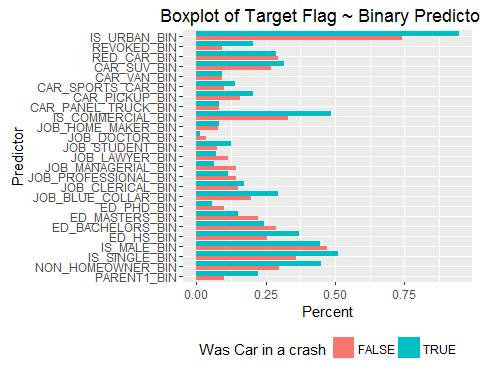
Let's get a view of the normalized values:

# Binary target variable

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])),training\_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 Values", x="Predictor")



bin\_summary <- melt(training\_normalized[,c(inputs\_bin,target\_bin)], measure.vars = 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="Predictor")



possible correlations

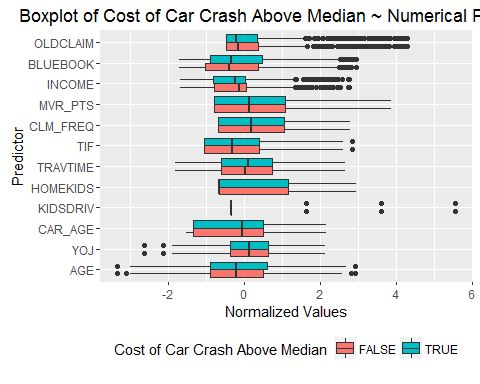
summary\_positive <- describe(training\_normalized[training\_normalized$TARGET\_FLAG==1,c(target\_bin,inputs)])[,c("mean","n")]  
summary\_negative <- describe(training\_normalized[training\_normalized$TARGET\_FLAG==0,c(target\_bin,inputs)])[,c("mean","n")]  
summary\_by\_target <- merge(summary\_positive,summary\_negative,by="row.names")  
colnames(summary\_by\_target) <- c("Variable","In car crash - Avg","In car crash - n","NOT In car crash - Avg", "NOT In car crash - n")  
summary\_by\_target$delta <- abs(summary\_by\_target[,"In car crash - Avg"]-summary\_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 |

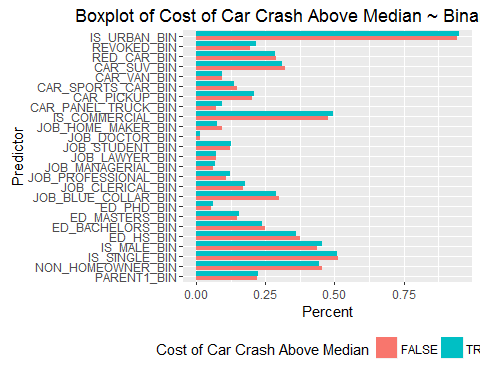
# Numerical target variable - Cost of Car Crash

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

#require("reshape2")  
#require("ggplot2")  
#detach(package:plyr)  
#require("dplyr")  
  
# 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(training\_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 Predictors", y="Normalized Values", x="Predictor")



bin\_summary <- melt(training\_normalized[,c(inputs\_bin,target\_bin)], measure.vars = 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")



possible correlations

summary\_positive <- describe(training\_normalized[training\_normalized$TARGET\_FLAG==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\_FLAG==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,target\_lm)])  
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,target\_lm)])  
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\_manual\_amt,target\_lm)])  
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(inputs,target\_bin)],family = binomial(link = "logit"))  
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   
## -2.5276 -0.7166 -0.4023 0.6484 3.1402   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -3.945e+00 4.059e-01 -9.719 < 2e-16 \*\*\*  
## PARENT1\_BIN 3.978e-01 1.309e-01 3.040 0.002369 \*\*   
## NON\_HOMEOWNER\_BIN 2.472e-01 9.245e-02 2.674 0.007499 \*\*   
## IS\_SINGLE\_BIN 4.756e-01 1.003e-01 4.740 2.13e-06 \*\*\*  
## IS\_MALE\_BIN 9.523e-02 1.346e-01 0.708 0.479178   
## ED\_HS\_BIN -9.008e-02 1.129e-01 -0.798 0.425078   
## ED\_BACHELORS\_BIN -4.925e-01 1.375e-01 -3.581 0.000343 \*\*\*  
## ED\_MASTERS\_BIN -4.700e-01 2.127e-01 -2.209 0.027144 \*   
## ED\_PHD\_BIN -3.839e-01 2.570e-01 -1.494 0.135266   
## JOB\_BLUE\_COLLAR\_BIN 1.566e-01 2.205e-01 0.710 0.477551   
## JOB\_CLERICAL\_BIN 3.571e-01 2.329e-01 1.533 0.125279   
## JOB\_PROFESSIONAL\_BIN 2.016e-01 2.112e-01 0.955 0.339812   
## JOB\_MANAGERIAL\_BIN -5.176e-01 2.036e-01 -2.543 0.010999 \*   
## JOB\_LAWYER\_BIN 1.592e-01 2.038e-01 0.781 0.434660   
## JOB\_STUDENT\_BIN 5.917e-02 2.532e-01 0.234 0.815247   
## JOB\_DOCTOR\_BIN -6.051e-01 3.469e-01 -1.744 0.081128 .   
## JOB\_HOME\_MAKER\_BIN 2.657e-01 2.427e-01 1.095 0.273640   
## IS\_COMMERCIAL\_BIN 8.213e-01 1.097e-01 7.489 6.94e-14 \*\*\*  
## CAR\_PANEL\_TRUCK\_BIN 5.916e-01 1.918e-01 3.084 0.002044 \*\*   
## CAR\_PICKUP\_BIN 5.491e-01 1.216e-01 4.516 6.31e-06 \*\*\*  
## CAR\_SPORTS\_CAR\_BIN 1.009e+00 1.552e-01 6.501 7.96e-11 \*\*\*  
## CAR\_VAN\_BIN 4.797e-01 1.552e-01 3.091 0.001996 \*\*   
## CAR\_SUV\_BIN 7.750e-01 1.339e-01 5.787 7.17e-09 \*\*\*  
## RED\_CAR\_BIN -2.916e-02 1.038e-01 -0.281 0.778736   
## REVOKED\_BIN 8.206e-01 1.090e-01 7.531 5.02e-14 \*\*\*  
## IS\_URBAN\_BIN 2.383e+00 1.362e-01 17.496 < 2e-16 \*\*\*  
## AGE -9.111e-04 4.802e-03 -0.190 0.849503   
## YOJ -3.451e-02 1.041e-02 -3.316 0.000912 \*\*\*  
## CAR\_AGE -8.470e-04 9.121e-03 -0.093 0.926016   
## KIDSDRIV 4.154e-01 7.211e-02 5.760 8.41e-09 \*\*\*  
## HOMEKIDS 7.147e-02 4.509e-02 1.585 0.112955   
## TRAVTIME 1.394e-02 2.304e-03 6.052 1.43e-09 \*\*\*  
## TIF -5.434e-02 8.822e-03 -6.159 7.30e-10 \*\*\*  
## CLM\_FREQ 1.863e-01 3.469e-02 5.370 7.88e-08 \*\*\*  
## MVR\_PTS 1.070e-01 1.653e-02 6.476 9.41e-11 \*\*\*  
## INCOME -3.922e-06 1.287e-06 -3.047 0.002309 \*\*   
## BLUEBOOK -2.364e-05 6.454e-06 -3.662 0.000250 \*\*\*  
## OLDCLAIM -1.296e-05 4.894e-06 -2.648 0.008088 \*\*   
## ---  
## 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: 5142.9 on 5675 degrees of freedom  
## AIC: 5218.9  
##   
## 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(inputs\_manual\_flag,target\_bin)],family = binomial(link = "logit"))  
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   
## -2.5428 -0.7208 -0.4095 0.6551 3.1344   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -3.713e+00 2.341e-01 -15.864 < 2e-16 \*\*\*  
## PARENT1\_BIN 5.322e-01 1.114e-01 4.779 1.77e-06 \*\*\*  
## NON\_HOMEOWNER\_BIN 2.279e-01 8.627e-02 2.642 0.008251 \*\*   
## IS\_SINGLE\_BIN 4.170e-01 9.338e-02 4.466 7.98e-06 \*\*\*  
## ED\_BACHELORS\_BIN -3.109e-01 8.455e-02 -3.677 0.000236 \*\*\*  
## ED\_MASTERS\_BIN -2.869e-01 1.019e-01 -2.814 0.004888 \*\*   
## JOB\_CLERICAL\_BIN 2.706e-01 1.090e-01 2.484 0.013006 \*   
## JOB\_MANAGERIAL\_BIN -6.470e-01 1.267e-01 -5.107 3.27e-07 \*\*\*  
## JOB\_HOME\_MAKER\_BIN 1.090e-01 1.446e-01 0.753 0.451203   
## IS\_COMMERCIAL\_BIN 8.579e-01 8.981e-02 9.553 < 2e-16 \*\*\*  
## CAR\_PANEL\_TRUCK\_BIN 6.029e-01 1.684e-01 3.579 0.000344 \*\*\*  
## CAR\_PICKUP\_BIN 5.205e-01 1.183e-01 4.401 1.08e-05 \*\*\*  
## CAR\_SPORTS\_CAR\_BIN 9.440e-01 1.270e-01 7.431 1.08e-13 \*\*\*  
## CAR\_VAN\_BIN 4.858e-01 1.464e-01 3.318 0.000908 \*\*\*  
## CAR\_SUV\_BIN 7.166e-01 1.025e-01 6.994 2.67e-12 \*\*\*  
## REVOKED\_BIN 8.311e-01 1.083e-01 7.674 1.67e-14 \*\*\*  
## IS\_URBAN\_BIN 2.363e+00 1.360e-01 17.373 < 2e-16 \*\*\*  
## YOJ -3.511e-02 9.465e-03 -3.710 0.000208 \*\*\*  
## KIDSDRIV 4.791e-01 6.469e-02 7.407 1.30e-13 \*\*\*  
## TRAVTIME 1.393e-02 2.290e-03 6.080 1.20e-09 \*\*\*  
## TIF -5.418e-02 8.798e-03 -6.158 7.38e-10 \*\*\*  
## CLM\_FREQ 1.829e-01 3.455e-02 5.296 1.18e-07 \*\*\*  
## MVR\_PTS 1.069e-01 1.647e-02 6.490 8.58e-11 \*\*\*  
## INCOME -6.064e-06 1.093e-06 -5.548 2.88e-08 \*\*\*  
## BLUEBOOK -2.646e-05 5.731e-06 -4.617 3.89e-06 \*\*\*  
## OLDCLAIM -1.249e-05 4.874e-06 -2.564 0.010356 \*   
## ---  
## 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: 5166.1 on 5687 degrees of freedom  
## AIC: 5218.1  
##   
## 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)  
predict2 <- 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\_MANAGERIAL\_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 + CAR\_AGE + KIDSDRIV + TRAVTIME +   
## TIF + CLM\_FREQ + MVR\_PTS + INCOME + BLUEBOOK + OLDCLAIM,   
## family = binomial(link = "probit"), data = training\_target\_flag[,   
## c(inputs, target\_bin)])  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.4745 -0.7405 -0.4142 0.6657 3.2999   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.965e+00 1.211e-01 -16.219 < 2e-16 \*\*\*  
## PARENT1\_BIN 2.566e-01 6.544e-02 3.922 8.80e-05 \*\*\*  
## NON\_HOMEOWNER\_BIN 1.408e-01 4.962e-02 2.838 0.004535 \*\*   
## IS\_SINGLE\_BIN 2.946e-01 5.372e-02 5.483 4.18e-08 \*\*\*  
## ED\_BACHELORS\_BIN -2.118e-01 5.136e-02 -4.124 3.72e-05 \*\*\*  
## ED\_MASTERS\_BIN -2.193e-01 6.789e-02 -3.230 0.001239 \*\*   
## JOB\_MANAGERIAL\_BIN -4.936e-01 7.129e-02 -6.924 4.39e-12 \*\*\*  
## JOB\_DOCTOR\_BIN -4.962e-01 1.517e-01 -3.271 0.001073 \*\*   
## IS\_COMMERCIAL\_BIN 4.060e-01 4.958e-02 8.189 2.64e-16 \*\*\*  
## CAR\_PANEL\_TRUCK\_BIN 3.153e-01 9.795e-02 3.219 0.001284 \*\*   
## CAR\_PICKUP\_BIN 2.393e-01 6.705e-02 3.569 0.000358 \*\*\*  
## CAR\_SPORTS\_CAR\_BIN 5.644e-01 7.216e-02 7.822 5.21e-15 \*\*\*  
## CAR\_VAN\_BIN 3.439e-01 8.172e-02 4.208 2.58e-05 \*\*\*  
## CAR\_SUV\_BIN 3.842e-01 5.741e-02 6.693 2.19e-11 \*\*\*  
## REVOKED\_BIN 5.320e-01 6.381e-02 8.338 < 2e-16 \*\*\*  
## IS\_URBAN\_BIN 1.212e+00 6.735e-02 17.993 < 2e-16 \*\*\*  
## YOJ -1.075e-02 5.141e-03 -2.090 0.036592 \*   
## CAR\_AGE -8.145e-03 4.732e-03 -1.721 0.085225 .   
## KIDSDRIV 2.328e-01 3.856e-02 6.037 1.57e-09 \*\*\*  
## TRAVTIME 7.429e-03 1.310e-03 5.671 1.42e-08 \*\*\*  
## TIF -3.358e-02 4.980e-03 -6.744 1.54e-11 \*\*\*  
## CLM\_FREQ 1.098e-01 2.047e-02 5.362 8.21e-08 \*\*\*  
## MVR\_PTS 7.099e-02 9.527e-03 7.452 9.23e-14 \*\*\*  
## INCOME -2.598e-06 6.418e-07 -4.048 5.18e-05 \*\*\*  
## BLUEBOOK -1.604e-05 3.245e-06 -4.943 7.70e-07 \*\*\*  
## OLDCLAIM -7.924e-06 2.904e-06 -2.728 0.006363 \*\*   
## ---  
## 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: 5175.4 on 5687 degrees of freedom  
## AIC: 5227.4  
##   
## 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\_flag\_model2), upper=formula(stepwise\_flag\_model)), direction = "forward", trace = 0)  
predict3 <- round(predict(forward, training\_trans\_eval\_bin ,type = 'response'), 4)  
summary(forward)

##   
## Call:  
## glm(formula = TARGET\_FLAG ~ IS\_URBAN\_BIN + MVR\_PTS + PARENT1\_BIN +   
## JOB\_MANAGERIAL\_BIN + IS\_COMMERCIAL\_BIN + BLUEBOOK + REVOKED\_BIN +   
## TIF + NON\_HOMEOWNER\_BIN + INCOME + TRAVTIME + KIDSDRIV +   
## CAR\_SPORTS\_CAR\_BIN + CAR\_SUV\_BIN + IS\_SINGLE\_BIN + CLM\_FREQ +   
## CAR\_AGE + ED\_BACHELORS\_BIN + OLDCLAIM + JOB\_DOCTOR\_BIN +   
## ED\_MASTERS\_BIN + CAR\_VAN\_BIN + CAR\_PICKUP\_BIN + CAR\_PANEL\_TRUCK\_BIN +   
## YOJ, family = binomial(link = "probit"), data = training\_target\_flag[,   
## c(inputs, target\_bin)])  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.4745 -0.7405 -0.4142 0.6657 3.2999   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.965e+00 1.211e-01 -16.219 < 2e-16 \*\*\*  
## IS\_URBAN\_BIN 1.212e+00 6.735e-02 17.993 < 2e-16 \*\*\*  
## MVR\_PTS 7.099e-02 9.527e-03 7.452 9.23e-14 \*\*\*  
## PARENT1\_BIN 2.566e-01 6.544e-02 3.922 8.80e-05 \*\*\*  
## JOB\_MANAGERIAL\_BIN -4.936e-01 7.129e-02 -6.924 4.39e-12 \*\*\*  
## IS\_COMMERCIAL\_BIN 4.060e-01 4.958e-02 8.189 2.64e-16 \*\*\*  
## BLUEBOOK -1.604e-05 3.245e-06 -4.943 7.70e-07 \*\*\*  
## REVOKED\_BIN 5.320e-01 6.381e-02 8.338 < 2e-16 \*\*\*  
## TIF -3.358e-02 4.980e-03 -6.744 1.54e-11 \*\*\*  
## NON\_HOMEOWNER\_BIN 1.408e-01 4.962e-02 2.838 0.004535 \*\*   
## INCOME -2.598e-06 6.418e-07 -4.048 5.18e-05 \*\*\*  
## TRAVTIME 7.429e-03 1.310e-03 5.671 1.42e-08 \*\*\*  
## KIDSDRIV 2.328e-01 3.856e-02 6.037 1.57e-09 \*\*\*  
## CAR\_SPORTS\_CAR\_BIN 5.644e-01 7.216e-02 7.822 5.21e-15 \*\*\*  
## CAR\_SUV\_BIN 3.842e-01 5.741e-02 6.693 2.19e-11 \*\*\*  
## IS\_SINGLE\_BIN 2.946e-01 5.372e-02 5.483 4.18e-08 \*\*\*  
## CLM\_FREQ 1.098e-01 2.047e-02 5.362 8.21e-08 \*\*\*  
## CAR\_AGE -8.145e-03 4.732e-03 -1.721 0.085225 .   
## ED\_BACHELORS\_BIN -2.118e-01 5.136e-02 -4.124 3.72e-05 \*\*\*  
## OLDCLAIM -7.924e-06 2.904e-06 -2.728 0.006363 \*\*   
## JOB\_DOCTOR\_BIN -4.962e-01 1.517e-01 -3.271 0.001073 \*\*   
## ED\_MASTERS\_BIN -2.193e-01 6.789e-02 -3.230 0.001239 \*\*   
## CAR\_VAN\_BIN 3.439e-01 8.172e-02 4.208 2.58e-05 \*\*\*  
## CAR\_PICKUP\_BIN 2.393e-01 6.705e-02 3.569 0.000358 \*\*\*  
## CAR\_PANEL\_TRUCK\_BIN 3.153e-01 9.795e-02 3.219 0.001284 \*\*   
## YOJ -1.075e-02 5.141e-03 -2.090 0.036592 \*   
## ---  
## 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: 5175.4 on 5687 degrees of freedom  
## AIC: 5227.4  
##   
## Number of Fisher Scoring iterations: 5

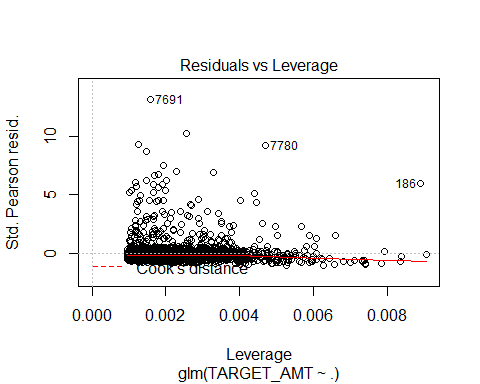
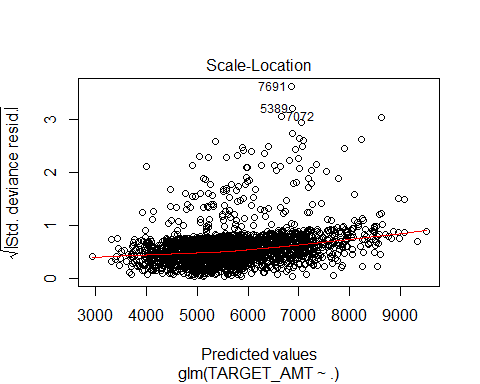
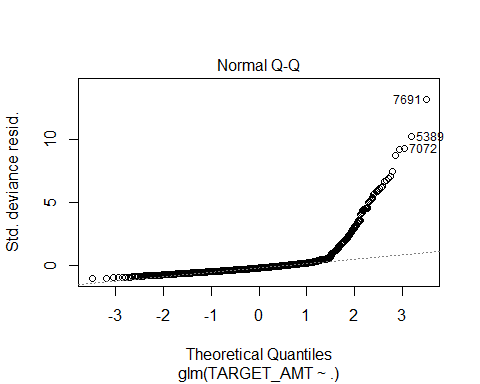
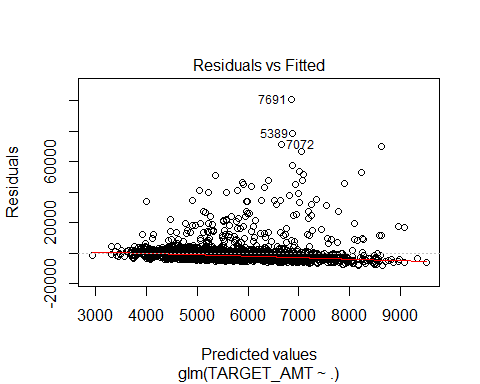
model7\_flag <- forward

## 4. SELECT MODELS (25 Points)

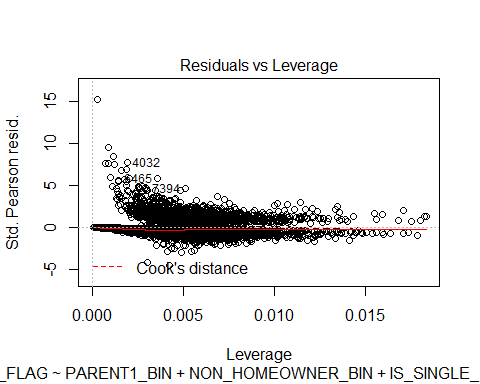
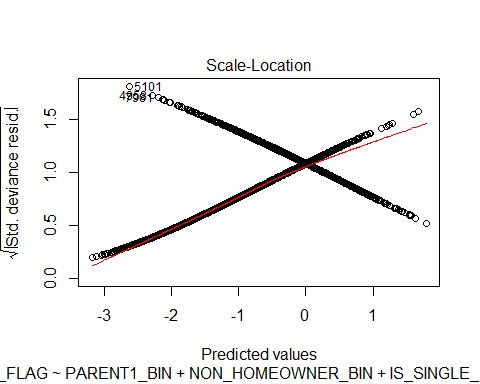
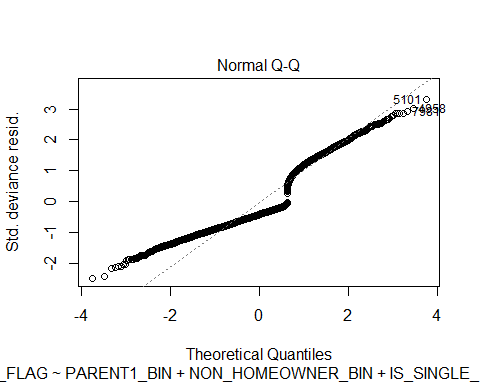
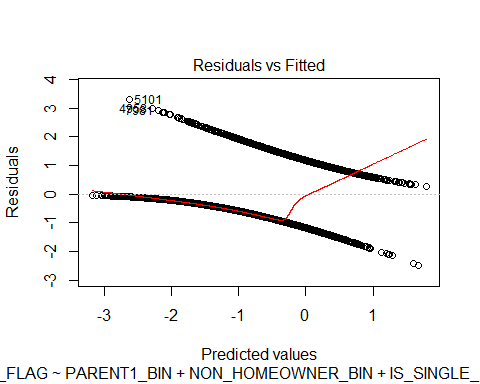
Decide on the criteria for selecting the best multiple linear regression model and the best binary logistic regression model. Will you select models with slightly worse performance if it makes more sense or is more parsimonious? Discuss why you selected your models.

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.

plot(model3\_amt)



plot(model6\_flag)



m3 <- summary(model3\_amt)  
m6 <- summary(model6\_flag)

# let's use this helper function that will return all the rates for future calculations  
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)  
}  
  
#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  
## 1 862 5584 423 1291

accuracy(d)

## [1] 0.789951

classification\_error\_rate(d)

## [1] 0.210049

precision\_c(d)

## [1] 0.6708171

sensitivity\_c(d)

## [1] 0.4003716

specificity\_c(d)

## [1] 0.9295822

f1\_score(d)

## [1] 0.5014543

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,predict2)  
plot(d\_roc, main = "ROC with pROC")

##   
## Call:  
## roc.default(response = training\_trans\_eval\_bin$TARGET\_FLAG, predictor = predict2)  
##   
## Data: predict2 in 6007 controls (training\_trans\_eval\_bin$TARGET\_FLAG 0) < 2153 cases (training\_trans\_eval\_bin$TARGET\_FLAG 1).  
## Area under the curve: 0.8103

#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  
## 1 862 5584 423 1291

accuracy(d)

## [1] 0.789951

classification\_error\_rate(d)

## [1] 0.210049

precision\_c(d)

## [1] 0.6708171

sensitivity\_c(d)

## [1] 0.4003716

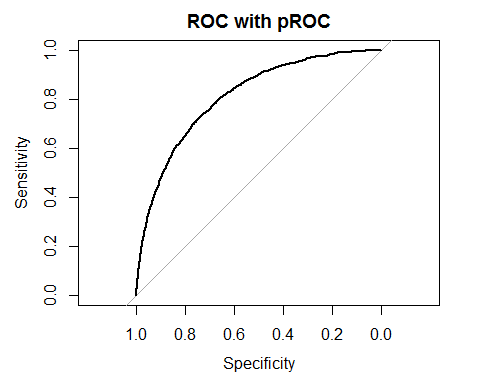
specificity\_c(d)

## [1] 0.9295822

f1\_score(d)

## [1] 0.5014543

require("pROC")  
d\_roc <- roc(training\_trans\_eval\_bin$TARGET\_FLAG,predict3)  
plot(d\_roc, main = "ROC with pROC")



##   
## Call:  
## roc.default(response = training\_trans\_eval\_bin$TARGET\_FLAG, predictor = predict3)  
##   
## Data: predict3 in 6007 controls (training\_trans\_eval\_bin$TARGET\_FLAG 0) < 2153 cases (training\_trans\_eval\_bin$TARGET\_FLAG 1).  
## Area under the curve: 0.8103

round(predict(forward, evaluation\_trans ,type = 'response'), 4)

## 1 2 3 4 5 6 7 8 9 10   
## 0.1430 0.2408 0.1554 0.3084 0.1747 0.2055 0.4545 0.3907 0.0337 0.1391   
## 11 12 13 14 15 16 17 18 19 20   
## 0.0235 0.6150 0.8263 0.0979 0.0355 0.6060 0.6776 0.1971 0.5817 0.3935   
## 21 22 23 24 25 26 27 28 29 30   
## 0.1452 0.3324 0.0372 0.3956 0.2839 0.3852 0.4844 0.4677 0.0718 0.2071   
## 31 32 33 34 35 36 37 38 39 40   
## 0.1129 0.4341 0.0983 0.2252 0.2365 0.0441 0.1521 0.1629 0.0895 0.5505   
## 41 42 43 44 45 46 47 48 49 50   
## 0.2497 0.5677 0.0189 0.5689 0.0016 0.2436 0.0479 0.3580 0.0112 0.5935   
## 51 52 53 54 55 56 57 58 59 60   
## 0.1248 0.3701 0.7394 0.0539 0.3307 0.2469 0.4368 0.4383 0.0901 0.5481   
## 61 62 63 64 65 66 67 68 69 70   
## 0.0108 0.0469 0.3836 0.1207 0.0663 0.2877 0.7869 0.5763 0.1945 0.0600   
## 71 72 73 74 75 76 77 78 79 80   
## 0.0189 0.2424 0.6990 0.3272 0.6421 0.2426 0.3371 0.2787 0.1518 0.0542   
## 81 82 83 84 85 86 87 88 89 90   
## 0.5273 0.4276 0.3063 0.0490 0.4370 0.5451 0.3412 0.2689 0.0322 0.7837   
## 91 92 93 94 95 96 97 98 99 100   
## 0.0884 0.1232 0.0580 0.1024 0.0382 0.1195 0.0054 0.3670 0.3766 0.1319   
## 101 102 103 104 105 106 107 108 109 110   
## 0.3737 0.5535 0.5771 0.6968 0.0931 0.0903 0.2186 0.0939 0.4576 0.2206   
## 111 112 113 114 115 116 117 118 119 120   
## 0.7034 0.0343 0.2452 0.1652 0.6101 0.0563 0.0132 0.6622 0.5314 0.1461   
## 121 122 123 124 125 126 127 128 129 130   
## 0.3000 0.7745 0.6126 0.4914 0.2572 0.4494 0.3705 0.2819 0.2308 0.1082   
## 131 132 133 134 135 136 137 138 139 140   
## 0.1105 0.1233 0.2074 0.0657 0.1060 0.2284 0.7488 0.5755 0.0467 0.2020   
## 141 142 143 144 145 146 147 148 149 150   
## 0.0522 0.7631 0.1501 0.1025 0.0910 0.5665 0.1504 0.1397 0.3782 0.0325   
## 151 152 153 154 155 156 157 158 159 160   
## 0.6029 0.3582 0.7840 0.4505 0.3382 0.4574 0.0369 0.1444 0.5061 0.5099   
## 161 162 163 164 165 166 167 168 169 170   
## 0.2256 0.1572 0.2903 0.1346 0.6603 0.2671 0.0498 0.1175 0.3817 0.1234   
## 171 172 173 174 175 176 177 178 179 180   
## 0.0726 0.4387 0.1954 0.8536 0.3085 0.3249 0.4431 0.5347 0.7311 0.6584   
## 181 182 183 184 185 186 187 188 189 190   
## 0.5525 0.2270 0.1247 0.2338 0.2719 0.3807 0.0590 0.0737 0.1927 0.2744   
## 191 192 193 194 195 196 197 198 199 200   
## 0.4869 0.6555 0.4738 0.3727 0.0975 0.5542 0.4832 0.2191 0.3255 0.1526   
## 201 202 203 204 205 206 207 208 209 210   
## 0.1946 0.3326 0.0628 0.0543 0.1042 0.0154 0.7884 0.0254 0.0721 0.0664   
## 211 212 213 214 215 216 217 218 219 220   
## 0.0219 0.2250 0.5929 0.5126 0.0777 0.3052 0.4128 0.0694 0.1744 0.0441   
## 221 222 223 224 225 226 227 228 229 230   
## 0.0605 0.0445 0.4658 0.3890 0.1177 0.2801 0.6174 0.5391 0.3494 0.1430   
## 231 232 233 234 235 236 237 238 239 240   
## 0.0299 0.0577 0.4763 0.1393 0.0082 0.2242 0.1113 0.0247 0.0183 0.6621   
## 241 242 243 244 245 246 247 248 249 250   
## 0.0330 0.1360 0.5991 0.2763 0.1728 0.4548 0.2974 0.4103 0.0575 0.5342   
## 251 252 253 254 255 256 257 258 259 260   
## 0.4965 0.6191 0.0597 0.2841 0.4532 0.3334 0.1115 0.2922 0.5650 0.0816   
## 261 262 263 264 265 266 267 268 269 270   
## 0.0896 0.0936 0.2029 0.1457 0.2013 0.0823 0.1049 0.1005 0.8686 0.4272   
## 271 272 273 274 275 276 277 278 279 280   
## 0.5018 0.0551 0.2897 0.6455 0.1330 0.3786 0.6109 0.2114 0.2631 0.2652   
## 281 282 283 284 285 286 287 288 289 290   
## 0.2404 0.3098 0.3065 0.3516 0.2590 0.4706 0.0158 0.3384 0.4966 0.7564   
## 291 292 293 294 295 296 297 298 299 300   
## 0.3924 0.2184 0.0884 0.4279 0.0316 0.4059 0.1816 0.5611 0.1151 0.3048   
## 301 302 303 304 305 306 307 308 309 310   
## 0.2751 0.0699 0.2897 0.4693 0.5333 0.3689 0.1334 0.4626 0.0158 0.3340   
## 311 312 313 314 315 316 317 318 319 320   
## 0.7009 0.3232 0.0485 0.8912 0.2312 0.1282 0.2789 0.1028 0.6173 0.0146   
## 321 322 323 324 325 326 327 328 329 330   
## 0.2585 0.5258 0.1612 0.2530 0.7551 0.3859 0.6116 0.1888 0.4399 0.2680   
## 331 332 333 334 335 336 337 338 339 340   
## 0.0702 0.0406 0.7861 0.1282 0.2671 0.3581 0.1944 0.5753 0.1822 0.0581   
## 341 342 343 344 345 346 347 348 349 350   
## 0.5261 0.5131 0.3832 0.6488 0.0318 0.0401 0.1357 0.0711 0.0696 0.2616   
## 351 352 353 354 355 356 357 358 359 360   
## 0.0373 0.3678 0.8198 0.7277 0.0832 0.6403 0.3252 0.2565 0.0221 0.0772   
## 361 362 363 364 365 366 367 368 369 370   
## 0.6454 0.2470 0.1348 0.5447 0.1410 0.3484 0.1213 0.3605 0.0712 0.1888   
## 371 372 373 374 375 376 377 378 379 380   
## 0.1588 0.3328 0.4686 0.2160 0.1610 0.5915 0.0088 0.2582 0.0267 0.1222   
## 381 382 383 384 385 386 387 388 389 390   
## 0.0795 0.2769 0.3364 0.2999 0.0407 0.4662 0.3261 0.1962 0.0991 0.5968   
## 391 392 393 394 395 396 397 398 399 400   
## 0.3038 0.0760 0.1953 0.1753 0.0519 0.3616 0.0680 0.3278 0.3113 0.5099   
## 401 402 403 404 405 406 407 408 409 410   
## 0.3579 0.2666 0.0885 0.0786 0.0784 0.2418 0.0160 0.3476 0.0070 0.1175   
## 411 412 413 414 415 416 417 418 419 420   
## 0.0741 0.7754 0.2449 0.0160 0.5556 0.2797 0.0125 0.3547 0.0024 0.1000   
## 421 422 423 424 425 426 427 428 429 430   
## 0.5740 0.6441 0.7742 0.0183 0.2571 0.4754 0.1012 0.2674 0.4316 0.3955   
## 431 432 433 434 435 436 437 438 439 440   
## 0.3926 0.0316 0.0104 0.0162 0.2999 0.7107 0.0528 0.0289 0.1916 0.1595   
## 441 442 443 444 445 446 447 448 449 450   
## 0.0136 0.1701 0.0872 0.2324 0.0483 0.0199 0.1590 0.1236 0.6121 0.7268   
## 451 452 453 454 455 456 457 458 459 460   
## 0.0692 0.1624 0.3783 0.1026 0.4464 0.5221 0.3056 0.7859 0.0446 0.2882   
## 461 462 463 464 465 466 467 468 469 470   
## 0.0406 0.0085 0.0262 0.3182 0.0321 0.1192 0.7721 0.5901 0.2078 0.3533   
## 471 472 473 474 475 476 477 478 479 480   
## 0.0782 0.7534 0.0830 0.0654 0.1864 0.0279 0.6082 0.8742 0.0495 0.3641   
## 481 482 483 484 485 486 487 488 489 490   
## 0.0523 0.2533 0.2366 0.3363 0.7898 0.5667 0.1702 0.4302 0.1773 0.7122   
## 491 492 493 494 495 496 497 498 499 500   
## 0.5248 0.1016 0.1222 0.2880 0.1100 0.5894 0.2678 0.0637 0.4518 0.3712   
## 501 502 503 504 505 506 507 508 509 510   
## 0.2486 0.3469 0.7681 0.1859 0.5973 0.0604 0.4543 0.2501 0.2218 0.2599   
## 511 512 513 514 515 516 517 518 519 520   
## 0.0708 0.1850 0.3102 0.0189 0.1667 0.0804 0.5986 0.4903 0.3726 0.5151   
## 521 522 523 524 525 526 527 528 529 530   
## 0.1892 0.1405 0.2572 0.2936 0.1126 0.0387 0.0489 0.0335 0.0022 0.1237   
## 531 532 533 534 535 536 537 538 539 540   
## 0.0063 0.2245 0.2170 0.3292 0.0428 0.2253 0.2936 0.0220 0.2174 0.0226   
## 541 542 543 544 545 546 547 548 549 550   
## 0.0341 0.0855 0.3869 0.3687 0.0080 0.2720 0.0700 0.8007 0.4770 0.0527   
## 551 552 553 554 555 556 557 558 559 560   
## 0.1290 0.0875 0.0601 0.4407 0.0673 0.1007 0.3716 0.0629 0.3884 0.2743   
## 561 562 563 564 565 566 567 568 569 570   
## 0.0729 0.2619 0.3348 0.2648 0.5041 0.1066 0.9578 0.3322 0.4563 0.6306   
## 571 572 573 574 575 576 577 578 579 580   
## 0.1154 0.2406 0.2405 0.0705 0.1093 0.0013 0.1821 0.1018 0.5105 0.1799   
## 581 582 583 584 585 586 587 588 589 590   
## 0.0518 0.5442 0.0853 0.8050 0.0319 0.0625 0.0527 0.1482 0.7200 0.5141   
## 591 592 593 594 595 596 597 598 599 600   
## 0.4108 0.1427 0.0032 0.3103 0.5661 0.5086 0.5166 0.2475 0.2717 0.5873   
## 601 602 603 604 605 606 607 608 609 610   
## 0.5468 0.2806 0.1924 0.1817 0.3632 0.3935 0.5422 0.1271 0.1702 0.1040   
## 611 612 613 614 615 616 617 618 619 620   
## 0.1550 0.3143 0.0112 0.0233 0.3212 0.0225 0.0464 0.4452 0.2716 0.5469   
## 621 622 623 624 625 626 627 628 629 630   
## 0.2248 0.1867 0.1526 0.2890 0.2240 0.7080 0.5229 0.0928 0.0257 0.5525   
## 631 632 633 634 635 636 637 638 639 640   
## 0.0179 0.3041 0.2053 0.1835 0.3172 0.2175 0.0849 0.5266 0.2286 0.2149   
## 641 642 643 644 645 646 647 648 649 650   
## 0.2079 0.3179 0.0501 0.0540 0.2126 0.4852 0.4393 0.3477 0.2622 0.4370   
## 651 652 653 654 655 656 657 658 659 660   
## 0.0271 0.1010 0.7274 0.0023 0.1430 0.2112 0.3092 0.0217 0.1863 0.0693   
## 661 662 663 664 665 666 667 668 669 670   
## 0.3995 0.5040 0.3998 0.2174 0.2209 0.4110 0.4407 0.0744 0.3338 0.0709   
## 671 672 673 674 675 676 677 678 679 680   
## 0.2370 0.5756 0.6235 0.0633 0.3780 0.2391 0.1724 0.3562 0.0224 0.1039   
## 681 682 683 684 685 686 687 688 689 690   
## 0.5813 0.1710 0.3466 0.0506 0.3419 0.1052 0.0135 0.2494 0.2689 0.2626   
## 691 692 693 694 695 696 697 698 699 700   
## 0.1943 0.0778 0.0253 0.2629 0.0823 0.4242 0.0987 0.3286 0.3376 0.2662   
## 701 702 703 704 705 706 707 708 709 710   
## 0.1710 0.0336 0.3592 0.0385 0.0688 0.1180 0.2974 0.7890 0.1571 0.1072   
## 711 712 713 714 715 716 717 718 719 720   
## 0.1380 0.0772 0.4417 0.2156 0.0366 0.0229 0.0279 0.0611 0.0358 0.1416   
## 721 722 723 724 725 726 727 728 729 730   
## 0.5870 0.0471 0.0335 0.0292 0.0256 0.1638 0.0633 0.0533 0.0922 0.1545   
## 731 732 733 734 735 736 737 738 739 740   
## 0.5338 0.4886 0.3918 0.0600 0.2841 0.4828 0.0365 0.0692 0.2889 0.2775   
## 741 742 743 744 745 746 747 748 749 750   
## 0.6398 0.2534 0.6762 0.1134 0.3160 0.3460 0.6361 0.1613 0.1050 0.3010   
## 751 752 753 754 755 756 757 758 759 760   
## 0.1242 0.2808 0.5163 0.6209 0.0456 0.3736 0.3386 0.2947 0.1886 0.3601   
## 761 762 763 764 765 766 767 768 769 770   
## 0.2068 0.7437 0.4498 0.4511 0.5324 0.6138 0.3993 0.1450 0.4550 0.0483   
## 771 772 773 774 775 776 777 778 779 780   
## 0.4299 0.4739 0.3419 0.5430 0.0487 0.3933 0.1224 0.3421 0.1267 0.0171   
## 781 782 783 784 785 786 787 788 789 790   
## 0.3248 0.8160 0.0421 0.2991 0.1800 0.4250 0.0844 0.2199 0.3406 0.0750   
## 791 792 793 794 795 796 797 798 799 800   
## 0.0653 0.2016 0.3024 0.2372 0.1284 0.3419 0.1085 0.6473 0.6260 0.4255   
## 801 802 803 804 805 806 807 808 809 810   
## 0.1160 0.0340 0.0448 0.3658 0.0270 0.1571 0.1466 0.2228 0.0445 0.2772   
## 811 812 813 814 815 816 817 818 819 820   
## 0.3834 0.0555 0.0666 0.0241 0.2389 0.2826 0.0260 0.5323 0.5557 0.3963   
## 821 822 823 824 825 826 827 828 829 830   
## 0.5315 0.2049 0.5348 0.4978 0.5487 0.2860 0.2766 0.0411 0.0075 0.1810   
## 831 832 833 834 835 836 837 838 839 840   
## 0.0801 0.0803 0.4546 0.0281 0.0568 0.1328 0.2795 0.1782 0.0382 0.2365   
## 841 842 843 844 845 846 847 848 849 850   
## 0.0919 0.1171 0.2306 0.0834 0.0291 0.0730 0.4195 0.3100 0.6641 0.6656   
## 851 852 853 854 855 856 857 858 859 860   
## 0.5505 0.1435 0.0966 0.3006 0.4581 0.0022 0.0617 0.1416 0.7776 0.1725   
## 861 862 863 864 865 866 867 868 869 870   
## 0.1446 0.6227 0.3227 0.0664 0.2648 0.1328 0.6821 0.1263 0.0192 0.6399   
## 871 872 873 874 875 876 877 878 879 880   
## 0.0747 0.5750 0.1402 0.8680 0.4745 0.2569 0.0479 0.1590 0.0132 0.4399   
## 881 882 883 884 885 886 887 888 889 890   
## 0.4513 0.2050 0.0903 0.0586 0.5550 0.3717 0.6973 0.0339 0.0564 0.1651   
## 891 892 893 894 895 896 897 898 899 900   
## 0.4212 0.1521 0.0404 0.0179 0.1009 0.3170 0.2764 0.0446 0.4709 0.3197   
## 901 902 903 904 905 906 907 908 909 910   
## 0.0915 0.1247 0.6586 0.0015 0.0357 0.1659 0.5478 0.0769 0.4219 0.4001   
## 911 912 913 914 915 916 917 918 919 920   
## 0.6390 0.1168 0.0232 0.2104 0.0571 0.4159 0.7274 0.5153 0.1978 0.3502   
## 921 922 923 924 925 926 927 928 929 930   
## 0.2147 0.1178 0.1300 0.0994 0.0430 0.0372 0.1824 0.0262 0.6802 0.0101   
## 931 932 933 934 935 936 937 938 939 940   
## 0.2204 0.8796 0.0310 0.4310 0.3220 0.3756 0.0991 0.3962 0.4244 0.3182   
## 941 942 943 944 945 946 947 948 949 950   
## 0.7498 0.0821 0.5058 0.2126 0.4794 0.1281 0.2629 0.1927 0.1039 0.5024   
## 951 952 953 954 955 956 957 958 959 960   
## 0.0491 0.2427 0.3530 0.5110 0.0504 0.4599 0.3156 0.2484 0.3848 0.0322   
## 961 962 963 964 965 966 967 968 969 970   
## 0.4203 0.2854 0.3744 0.1020 0.0751 0.5826 0.2455 0.3022 0.0871 0.1845   
## 971 972 973 974 975 976 977 978 979 980   
## 0.6235 0.0740 0.0413 0.4529 0.0082 0.3319 0.3756 0.0714 0.1297 0.3018   
## 981 982 983 984 985 986 987 988 989 990   
## 0.1166 0.0684 0.7002 0.4657 0.5783 0.5599 0.3767 0.0457 0.3078 0.5469   
## 991 992 993 994 995 996 997 998 999 1000   
## 0.1322 0.0103 0.2389 0.0653 0.3774 0.2811 0.2133 0.0342 0.2495 0.2078   
## 1001 1002 1003 1004 1005 1006 1007 1008 1009 1010   
## 0.3709 0.6611 0.5774 0.1615 0.4690 0.0269 0.1305 0.4395 0.3360 0.1203   
## 1011 1012 1013 1014 1015 1016 1017 1018 1019 1020   
## 0.3559 0.0815 0.1295 0.0439 0.0463 0.4011 0.1425 0.4654 0.3217 0.0192   
## 1021 1022 1023 1024 1025 1026 1027 1028 1029 1030   
## 0.1737 0.0414 0.5380 0.4057 0.5199 0.6944 0.4891 0.0462 0.2366 0.0215   
## 1031 1032 1033 1034 1035 1036 1037 1038 1039 1040   
## 0.0984 0.0580 0.2386 0.1568 0.2949 0.0518 0.0450 0.0984 0.1217 0.2527   
## 1041 1042 1043 1044 1045 1046 1047 1048 1049 1050   
## 0.2532 0.4774 0.5236 0.0312 0.7635 0.0455 0.3247 0.4689 0.4560 0.3751   
## 1051 1052 1053 1054 1055 1056 1057 1058 1059 1060   
## 0.5070 0.5309 0.7184 0.0124 0.4060 0.0900 0.2553 0.0173 0.4869 0.6108   
## 1061 1062 1063 1064 1065 1066 1067 1068 1069 1070   
## 0.2399 0.7656 0.4326 0.0590 0.0603 0.1218 0.1229 0.1616 0.0298 0.4905   
## 1071 1072 1073 1074 1075 1076 1077 1078 1079 1080   
## 0.3298 0.1675 0.3010 0.7252 0.1880 0.0104 0.0190 0.0494 0.6690 0.0258   
## 1081 1082 1083 1084 1085 1086 1087 1088 1089 1090   
## 0.7205 0.5531 0.0441 0.3829 0.4946 0.8082 0.0279 0.2683 0.1889 0.0090   
## 1091 1092 1093 1094 1095 1096 1097 1098 1099 1100   
## 0.1340 0.0196 0.1487 0.4050 0.2256 0.1430 0.1994 0.0891 0.3815 0.7995   
## 1101 1102 1103 1104 1105 1106 1107 1108 1109 1110   
## 0.0109 0.1383 0.5823 0.3430 0.3333 0.3396 0.0621 0.2348 0.1205 0.4150   
## 1111 1112 1113 1114 1115 1116 1117 1118 1119 1120   
## 0.6340 0.3354 0.1798 0.2318 0.2205 0.0010 0.3050 0.8188 0.4911 0.1679   
## 1121 1122 1123 1124 1125 1126 1127 1128 1129 1130   
## 0.3899 0.5270 0.0218 0.1278 0.0681 0.3142 0.6899 0.0391 0.0231 0.1865   
## 1131 1132 1133 1134 1135 1136 1137 1138 1139 1140   
## 0.1480 0.0674 0.7337 0.0517 0.5931 0.0464 0.1397 0.2192 0.0833 0.3005   
## 1141 1142 1143 1144 1145 1146 1147 1148 1149 1150   
## 0.0940 0.2046 0.3259 0.3691 0.5801 0.0644 0.2136 0.7191 0.1466 0.4890   
## 1151 1152 1153 1154 1155 1156 1157 1158 1159 1160   
## 0.4675 0.7829 0.1912 0.3341 0.7214 0.3046 0.0353 0.2945 0.2969 0.0140   
## 1161 1162 1163 1164 1165 1166 1167 1168 1169 1170   
## 0.0945 0.0830 0.2054 0.2080 0.0514 0.4652 0.0821 0.4694 0.3106 0.0327   
## 1171 1172 1173 1174 1175 1176 1177 1178 1179 1180   
## 0.4716 0.5492 0.5128 0.5816 0.0319 0.1270 0.2977 0.0056 0.3084 0.7174   
## 1181 1182 1183 1184 1185 1186 1187 1188 1189 1190   
## 0.0633 0.8197 0.1841 0.3085 0.8662 0.0605 0.0747 0.0835 0.2667 0.2280   
## 1191 1192 1193 1194 1195 1196 1197 1198 1199 1200   
## 0.3415 0.1160 0.0157 0.5781 0.2151 0.0294 0.1123 0.0392 0.5312 0.3634   
## 1201 1202 1203 1204 1205 1206 1207 1208 1209 1210   
## 0.1375 0.1852 0.2326 0.2535 0.2551 0.0143 0.6023 0.2072 0.2007 0.4207   
## 1211 1212 1213 1214 1215 1216 1217 1218 1219 1220   
## 0.0285 0.2814 0.5806 0.0243 0.0861 0.0173 0.4884 0.3754 0.1047 0.1228   
## 1221 1222 1223 1224 1225 1226 1227 1228 1229 1230   
## 0.0427 0.2965 0.7172 0.2362 0.6488 0.0253 0.0277 0.0796 0.3967 0.6107   
## 1231 1232 1233 1234 1235 1236 1237 1238 1239 1240   
## 0.1177 0.2654 0.4293 0.6455 0.0191 0.0028 0.0140 0.7476 0.1264 0.0443   
## 1241 1242 1243 1244 1245 1246 1247 1248 1249 1250   
## 0.3446 0.0038 0.0115 0.2856 0.1483 0.6867 0.3591 0.0043 0.0332 0.0614   
## 1251 1252 1253 1254 1255 1256 1257 1258 1259 1260   
## 0.1390 0.7692 0.0159 0.3247 0.2835 0.1036 0.7374 0.0383 0.4329 0.1580   
## 1261 1262 1263 1264 1265 1266 1267 1268 1269 1270   
## 0.2279 0.3076 0.0158 0.7578 0.1076 0.2730 0.3824 0.0224 0.4093 0.1250   
## 1271 1272 1273 1274 1275 1276 1277 1278 1279 1280   
## 0.3089 0.2890 0.3195 0.3045 0.0961 0.0226 0.3618 0.1831 0.2915 0.0855   
## 1281 1282 1283 1284 1285 1286 1287 1288 1289 1290   
## 0.6537 0.3140 0.2825 0.4038 0.4485 0.2478 0.5230 0.2676 0.2193 0.2108   
## 1291 1292 1293 1294 1295 1296 1297 1298 1299 1300   
## 0.6764 0.2455 0.0033 0.4974 0.4394 0.0815 0.2818 0.2610 0.6290 0.3609   
## 1301 1302 1303 1304 1305 1306 1307 1308 1309 1310   
## 0.0657 0.1366 0.0621 0.2268 0.0107 0.0834 0.6126 0.4621 0.0504 0.7683   
## 1311 1312 1313 1314 1315 1316 1317 1318 1319 1320   
## 0.5365 0.2228 0.8402 0.2321 0.0318 0.0189 0.2280 0.0370 0.0593 0.8013   
## 1321 1322 1323 1324 1325 1326 1327 1328 1329 1330   
## 0.1690 0.4532 0.6962 0.0163 0.0652 0.5298 0.1720 0.3791 0.0119 0.2993   
## 1331 1332 1333 1334 1335 1336 1337 1338 1339 1340   
## 0.0663 0.1037 0.0923 0.0839 0.4046 0.3512 0.1622 0.0115 0.3503 0.2764   
## 1341 1342 1343 1344 1345 1346 1347 1348 1349 1350   
## 0.1054 0.5449 0.0135 0.1458 0.5506 0.0271 0.2274 0.4020 0.0180 0.2891   
## 1351 1352 1353 1354 1355 1356 1357 1358 1359 1360   
## 0.1205 0.5261 0.1795 0.2276 0.0395 0.0045 0.1367 0.4380 0.0574 0.2641   
## 1361 1362 1363 1364 1365 1366 1367 1368 1369 1370   
## 0.0043 0.0247 0.5619 0.0918 0.0864 0.2062 0.6001 0.6534 0.2874 0.4807   
## 1371 1372 1373 1374 1375 1376 1377 1378 1379 1380   
## 0.2691 0.0650 0.0248 0.2788 0.1999 0.3753 0.3511 0.3372 0.0290 0.2206   
## 1381 1382 1383 1384 1385 1386 1387 1388 1389 1390   
## 0.5451 0.6993 0.4982 0.0187 0.3367 0.0396 0.0638 0.0778 0.0310 0.4648   
## 1391 1392 1393 1394 1395 1396 1397 1398 1399 1400   
## 0.6891 0.4785 0.0598 0.4586 0.1243 0.1665 0.1662 0.4785 0.3459 0.2989   
## 1401 1402 1403 1404 1405 1406 1407 1408 1409 1410   
## 0.2506 0.1399 0.7495 0.4648 0.3517 0.3253 0.0894 0.2106 0.0580 0.6387   
## 1411 1412 1413 1414 1415 1416 1417 1418 1419 1420   
## 0.1075 0.2460 0.0454 0.1836 0.0095 0.0213 0.1132 0.0284 0.3779 0.0650   
## 1421 1422 1423 1424 1425 1426 1427 1428 1429 1430   
## 0.0121 0.8557 0.4582 0.1006 0.3921 0.8444 0.0244 0.2445 0.4643 0.0715   
## 1431 1432 1433 1434 1435 1436 1437 1438 1439 1440   
## 0.4314 0.0141 0.0612 0.2115 0.0826 0.4016 0.4277 0.5144 0.0961 0.0199   
## 1441 1442 1443 1444 1445 1446 1447 1448 1449 1450   
## 0.0640 0.6981 0.5857 0.0712 0.1994 0.5324 0.4232 0.2578 0.2151 0.0955   
## 1451 1452 1453 1454 1455 1456 1457 1458 1459 1460   
## 0.0765 0.3086 0.2618 0.0391 0.0488 0.1418 0.6547 0.0309 0.0986 0.0581   
## 1461 1462 1463 1464 1465 1466 1467 1468 1469 1470   
## 0.0035 0.1411 0.0328 0.4351 0.2324 0.4265 0.0353 0.0665 0.1923 0.3720   
## 1471 1472 1473 1474 1475 1476 1477 1478 1479 1480   
## 0.2369 0.1260 0.0851 0.2317 0.3882 0.0030 0.4598 0.1865 0.1609 0.0368   
## 1481 1482 1483 1484 1485 1486 1487 1488 1489 1490   
## 0.4677 0.0071 0.5271 0.3069 0.4297 0.2968 0.1974 0.3641 0.6678 0.1395   
## 1491 1492 1493 1494 1495 1496 1497 1498 1499 1500   
## 0.1953 0.7106 0.1122 0.0374 0.6704 0.3016 0.0917 0.0256 0.0195 0.0242   
## 1501 1502 1503 1504 1505 1506 1507 1508 1509 1510   
## 0.0196 0.4415 0.3433 0.6939 0.1094 0.4079 0.6778 0.3455 0.2426 0.2358   
## 1511 1512 1513 1514 1515 1516 1517 1518 1519 1520   
## 0.0329 0.2174 0.0844 0.2283 0.3271 0.4175 0.0307 0.4307 0.3138 0.0590   
## 1521 1522 1523 1524 1525 1526 1527 1528 1529 1530   
## 0.3342 0.0566 0.1354 0.3454 0.0126 0.3965 0.1910 0.4538 0.5191 0.6429   
## 1531 1532 1533 1534 1535 1536 1537 1538 1539 1540   
## 0.3672 0.5088 0.0421 0.0310 0.4148 0.3494 0.1653 0.2974 0.6346 0.7216   
## 1541 1542 1543 1544 1545 1546 1547 1548 1549 1550   
## 0.0497 0.2354 0.0717 0.4204 0.6028 0.2745 0.0553 0.3178 0.1194 0.1970   
## 1551 1552 1553 1554 1555 1556 1557 1558 1559 1560   
## 0.3384 0.1156 0.1302 0.6189 0.2672 0.0668 0.1776 0.0330 0.4218 0.2175   
## 1561 1562 1563 1564 1565 1566 1567 1568 1569 1570   
## 0.1472 0.6672 0.3571 0.3526 0.7621 0.2051 0.1317 0.2098 0.1942 0.3926   
## 1571 1572 1573 1574 1575 1576 1577 1578 1579 1580   
## 0.5630 0.0756 0.0310 0.4143 0.0553 0.3920 0.2808 0.0342 0.1553 0.3001   
## 1581 1582 1583 1584 1585 1586 1587 1588 1589 1590   
## 0.2604 0.0210 0.2484 0.0518 0.0941 0.0490 0.0310 0.3312 0.0105 0.4218   
## 1591 1592 1593 1594 1595 1596 1597 1598 1599 1600   
## 0.3464 0.7481 0.0773 0.1809 0.1126 0.1471 0.2825 0.2814 0.1451 0.3530   
## 1601 1602 1603 1604 1605 1606 1607 1608 1609 1610   
## 0.4344 0.3206 0.3343 0.5176 0.2312 0.7912 0.0088 0.0690 0.1666 0.7479   
## 1611 1612 1613 1614 1615 1616 1617 1618 1619 1620   
## 0.3692 0.1252 0.0664 0.4457 0.3764 0.5808 0.1727 0.5283 0.2810 0.4039   
## 1621 1622 1623 1624 1625 1626 1627 1628 1629 1630   
## 0.5627 0.0375 0.6163 0.1816 0.1780 0.2011 0.3460 0.0844 0.1260 0.8005   
## 1631 1632 1633 1634 1635 1636 1637 1638 1639 1640   
## 0.1404 0.6840 0.1577 0.3843 0.1805 0.3298 0.5645 0.2685 0.2758 0.2584   
## 1641 1642 1643 1644 1645 1646 1647 1648 1649 1650   
## 0.0489 0.3430 0.2938 0.1895 0.3302 0.0083 0.2150 0.1555 0.0252 0.5626   
## 1651 1652 1653 1654 1655 1656 1657 1658 1659 1660   
## 0.1805 0.0076 0.3530 0.0177 0.4910 0.3985 0.0012 0.0634 0.2215 0.0937   
## 1661 1662 1663 1664 1665 1666 1667 1668 1669 1670   
## 0.5232 0.6183 0.5013 0.6842 0.8781 0.0117 0.3359 0.3339 0.8350 0.1466   
## 1671 1672 1673 1674 1675 1676 1677 1678 1679 1680   
## 0.6484 0.3525 0.6366 0.1524 0.0325 0.2228 0.0463 0.0567 0.2000 0.2790   
## 1681 1682 1683 1684 1685 1686 1687 1688 1689 1690   
## 0.0822 0.8078 0.6816 0.2891 0.4683 0.2050 0.0206 0.3538 0.0073 0.0452   
## 1691 1692 1693 1694 1695 1696 1697 1698 1699 1700   
## 0.0505 0.0828 0.0323 0.3648 0.2438 0.6248 0.2254 0.5332 0.8581 0.0320   
## 1701 1702 1703 1704 1705 1706 1707 1708 1709 1710   
## 0.4886 0.0264 0.0971 0.2959 0.2471 0.1297 0.5805 0.2295 0.7839 0.0468   
## 1711 1712 1713 1714 1715 1716 1717 1718 1719 1720   
## 0.2804 0.0121 0.4186 0.1785 0.3429 0.0224 0.0712 0.0694 0.2432 0.0671   
## 1721 1722 1723 1724 1725 1726 1727 1728 1729 1730   
## 0.3032 0.3576 0.2651 0.3502 0.5023 0.0011 0.1626 0.2682 0.7330 0.4595   
## 1731 1732 1733 1734 1735 1736 1737 1738 1739 1740   
## 0.4723 0.0560 0.0224 0.3390 0.1427 0.0609 0.4079 0.1333 0.3652 0.0466   
## 1741 1742 1743 1744 1745 1746 1747 1748 1749 1750   
## 0.9090 0.2159 0.0328 0.0311 0.3391 0.1864 0.2112 0.2496 0.4573 0.2926   
## 1751 1752 1753 1754 1755 1756 1757 1758 1759 1760   
## 0.2426 0.3531 0.1537 0.6969 0.0421 0.1404 0.0790 0.4426 0.1779 0.1150   
## 1761 1762 1763 1764 1765 1766 1767 1768 1769 1770   
## 0.5609 0.0277 0.1410 0.0847 0.3568 0.7606 0.0740 0.3100 0.0944 0.1243   
## 1771 1772 1773 1774 1775 1776 1777 1778 1779 1780   
## 0.0119 0.1509 0.2154 0.7462 0.3385 0.0361 0.5472 0.4100 0.5360 0.3629   
## 1781 1782 1783 1784 1785 1786 1787 1788 1789 1790   
## 0.1776 0.0603 0.3145 0.2545 0.2638 0.0984 0.2144 0.0197 0.7641 0.3710   
## 1791 1792 1793 1794 1795 1796 1797 1798 1799 1800   
## 0.0550 0.1793 0.2670 0.2712 0.2993 0.4561 0.0246 0.2241 0.0441 0.3478   
## 1801 1802 1803 1804 1805 1806 1807 1808 1809 1810   
## 0.4140 0.1719 0.1271 0.2663 0.0872 0.3648 0.6437 0.5737 0.1622 0.2227   
## 1811 1812 1813 1814 1815 1816 1817 1818 1819 1820   
## 0.1782 0.0816 0.5156 0.3703 0.4875 0.0944 0.0381 0.1585 0.3189 0.2985   
## 1821 1822 1823 1824 1825 1826 1827 1828 1829 1830   
## 0.0267 0.1533 0.0307 0.3854 0.2902 0.7801 0.2708 0.4477 0.0040 0.3021   
## 1831 1832 1833 1834 1835 1836 1837 1838 1839 1840   
## 0.2104 0.0724 0.1643 0.0784 0.4097 0.1744 0.1447 0.8712 0.0626 0.6743   
## 1841 1842 1843 1844 1845 1846 1847 1848 1849 1850   
## 0.0657 0.1151 0.2608 0.2236 0.0315 0.0670 0.1065 0.1198 0.4055 0.1535   
## 1851 1852 1853 1854 1855 1856 1857 1858 1859 1860   
## 0.3150 0.2800 0.2817 0.0051 0.0049 0.2950 0.0563 0.1871 0.1417 0.0423   
## 1861 1862 1863 1864 1865 1866 1867 1868 1869 1870   
## 0.3623 0.0141 0.1310 0.2453 0.4676 0.1714 0.0661 0.0585 0.3985 0.4492   
## 1871 1872 1873 1874 1875 1876 1877 1878 1879 1880   
## 0.5662 0.2993 0.0446 0.0237 0.3101 0.1798 0.0197 0.7075 0.3260 0.3522   
## 1881 1882 1883 1884 1885 1886 1887 1888 1889 1890   
## 0.1434 0.0772 0.0375 0.1708 0.4828 0.3015 0.1631 0.0872 0.3386 0.3198   
## 1891 1892 1893 1894 1895 1896 1897 1898 1899 1900   
## 0.0677 0.2916 0.0252 0.0143 0.3220 0.1720 0.2124 0.2448 0.1183 0.5846   
## 1901 1902 1903 1904 1905 1906 1907 1908 1909 1910   
## 0.1754 0.3332 0.0366 0.3682 0.3006 0.0181 0.0152 0.2381 0.3298 0.7768   
## 1911 1912 1913 1914 1915 1916 1917 1918 1919 1920   
## 0.3093 0.3714 0.0568 0.2631 0.5031 0.4753 0.1575 0.0289 0.0586 0.2238   
## 1921 1922 1923 1924 1925 1926 1927 1928 1929 1930   
## 0.0228 0.4223 0.0934 0.1495 0.3146 0.0027 0.1762 0.1815 0.0109 0.1660   
## 1931 1932 1933 1934 1935 1936 1937 1938 1939 1940   
## 0.0067 0.2291 0.4628 0.5780 0.1204 0.5089 0.2707 0.0141 0.1341 0.1286   
## 1941 1942 1943 1944 1945 1946 1947 1948 1949 1950   
## 0.0557 0.0465 0.1284 0.0377 0.0115 0.3133 0.6572 0.0783 0.6828 0.0276   
## 1951 1952 1953 1954 1955 1956 1957 1958 1959 1960   
## 0.3054 0.1661 0.4903 0.4999 0.2553 0.3597 0.0888 0.1135 0.3370 0.3760   
## 1961 1962 1963 1964 1965 1966 1967 1968 1969 1970   
## 0.7061 0.3574 0.3537 0.3044 0.2306 0.0239 0.1528 0.4919 0.0136 0.0146   
## 1971 1972 1973 1974 1975 1976 1977 1978 1979 1980   
## 0.0394 0.0776 0.5644 0.6589 0.0617 0.3175 0.0061 0.3352 0.3825 0.1467   
## 1981 1982 1983 1984 1985 1986 1987 1988 1989 1990   
## 0.2068 0.4388 0.2062 0.0831 0.1463 0.2959 0.2235 0.1349 0.0933 0.3795   
## 1991 1992 1993 1994 1995 1996 1997 1998 1999 2000   
## 0.0108 0.7665 0.6278 0.5875 0.1875 0.1546 0.3245 0.7063 0.4382 0.0833   
## 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010   
## 0.5914 0.4076 0.6053 0.0170 0.5564 0.1703 0.4435 0.0210 0.3007 0.4590   
## 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020   
## 0.4555 0.0779 0.9667 0.0292 0.2187 0.4655 0.0741 0.7923 0.4565 0.0534   
## 2021 2022 2023 2024 2025 2026 2027 2028 2029 2030   
## 0.0249 0.1943 0.3698 0.2468 0.2600 0.0405 0.1540 0.1543 0.1336 0.5952   
## 2031 2032 2033 2034 2035 2036 2037 2038 2039 2040   
## 0.1676 0.1790 0.0210 0.0964 0.5919 0.4241 0.1928 0.0898 0.5628 0.1373   
## 2041 2042 2043 2044 2045 2046 2047 2048 2049 2050   
## 0.0715 0.0153 0.2275 0.3546 0.0170 0.0523 0.0468 0.4167 0.2562 0.2419   
## 2051 2052 2053 2054 2055 2056 2057 2058 2059 2060   
## 0.3165 0.2434 0.3645 0.0754 0.5757 0.0281 0.2799 0.2718 0.0470 0.1426   
## 2061 2062 2063 2064 2065 2066 2067 2068 2069 2070   
## 0.3162 0.3813 0.4583 0.2555 0.2987 0.0958 0.3827 0.8735 0.3097 0.0039   
## 2071 2072 2073 2074 2075 2076 2077 2078 2079 2080   
## 0.1621 0.2512 0.4355 0.4066 0.0145 0.0739 0.6775 0.2226 0.2432 0.9288   
## 2081 2082 2083 2084 2085 2086 2087 2088 2089 2090   
## 0.0294 0.4517 0.3027 0.0182 0.0334 0.0491 0.3014 0.5126 0.0961 0.5023   
## 2091 2092 2093 2094 2095 2096 2097 2098 2099 2100   
## 0.1202 0.0479 0.4443 0.2373 0.4369 0.6594 0.2369 0.4088 0.5918 0.5146   
## 2101 2102 2103 2104 2105 2106 2107 2108 2109 2110   
## 0.5618 0.3213 0.9130 0.1923 0.0213 0.1591 0.6088 0.0069 0.3517 0.0338   
## 2111 2112 2113 2114 2115 2116 2117 2118 2119 2120   
## 0.8041 0.0736 0.4503 0.0357 0.0884 0.0168 0.4615 0.2847 0.9211 0.2007   
## 2121 2122 2123 2124 2125 2126 2127 2128 2129 2130   
## 0.0337 0.2428 0.7234 0.2875 0.3070 0.0320 0.3820 0.1246 0.0404 0.0430   
## 2131 2132 2133 2134 2135 2136 2137 2138 2139 2140   
## 0.1745 0.2416 0.2285 0.2284 0.4215 0.0115 0.3495 0.0235 0.0044 0.2719   
## 2141   
## 0.1564

round(predict(model3\_amt, evaluation\_trans ,type = 'response'), 4)

## 1 2 3 4 5 6 7 8   
## 6963.024 7048.258 4472.240 5090.148 6639.648 7295.004 5538.837 7287.979   
## 9 10 11 12 13 14 15 16   
## 7000.632 8200.041 7086.523 5227.668 4963.666 5358.295 5872.140 5210.856   
## 17 18 19 20 21 22 23 24   
## 4048.659 6768.159 4316.762 4950.901 6634.334 4901.602 6990.862 5827.541   
## 25 26 27 28 29 30 31 32   
## 4903.829 6546.517 5034.565 6605.181 5755.365 6950.381 4293.126 6938.375   
## 33 34 35 36 37 38 39 40   
## 5650.593 6147.373 5663.399 6973.047 6500.319 6139.175 5254.789 5358.396   
## 41 42 43 44 45 46 47 48   
## 5007.892 5297.516 6145.552 6305.999 6451.568 5187.724 5538.585 5966.640   
## 49 50 51 52 53 54 55 56   
## 7036.416 4108.484 4662.550 6965.607 6068.059 5962.074 5281.613 6315.058   
## 57 58 59 60 61 62 63 64   
## 6638.081 5005.818 5522.235 4388.330 4754.366 5637.536 5898.857 6248.502   
## 65 66 67 68 69 70 71 72   
## 6164.126 3946.357 4525.486 6475.962 5643.863 7272.836 6320.678 6034.908   
## 73 74 75 76 77 78 79 80   
## 4672.268 5162.113 5453.146 5531.904 5220.574 5609.950 6343.958 5172.136   
## 81 82 83 84 85 86 87 88   
## 4843.193 6066.188 6388.299 6814.267 5986.986 6564.486 5940.267 6236.710   
## 89 90 91 92 93 94 95 96   
## 5596.235 5594.715 6754.949 6068.516 7373.811 6288.488 6753.278 8442.635   
## 97 98 99 100 101 102 103 104   
## 6179.665 7197.724 6337.431 6184.472 4540.519 6807.283 6158.862 5854.223   
## 105 106 107 108 109 110 111 112   
## 7326.285 6625.579 5886.000 5499.407 5687.188 6231.040 5474.464 4531.255   
## 113 114 115 116 117 118 119 120   
## 5159.330 4974.082 5136.757 5421.260 6255.030 5125.570 3745.376 5537.319   
## 121 122 123 124 125 126 127 128   
## 4523.260 5237.689 4492.748 5042.917 6524.147 5487.563 5668.762 5039.931   
## 129 130 131 132 133 134 135 136   
## 5362.902 5369.482 4953.189 6718.863 5126.187 5882.962 8285.477 6846.156   
## 137 138 139 140 141 142 143 144   
## 5817.114 6739.166 6450.604 4209.309 5443.225 5173.300 5600.333 5001.868   
## 145 146 147 148 149 150 151 152   
## 6292.079 4057.607 5922.849 5566.776 7779.384 6350.184 5153.764 6156.533   
## 153 154 155 156 157 158 159 160   
## 5252.461 4263.768 6376.556 4155.504 5668.055 5483.717 4730.475 5368.419   
## 161 162 163 164 165 166 167 168   
## 5942.437 4862.173 5416.351 5047.927 6857.392 5385.831 7152.575 7687.267   
## 169 170 171 172 173 174 175 176   
## 6157.809 6842.116 8182.628 7677.497 7253.501 6307.678 5265.266 3873.430   
## 177 178 179 180 181 182 183 184   
## 8623.327 5343.414 7836.588 4508.531 5838.725 8623.387 5433.254 6273.354   
## 185 186 187 188 189 190 191 192   
## 7356.046 5765.842 6381.120 4883.228 6212.869 6831.022 4876.446 4898.008   
## 193 194 195 196 197 198 199 200   
## 7757.367 4140.877 6443.873 4981.218 5862.465 6529.765 5167.328 7980.831   
## 201 202 203 204 205 206 207 208   
## 7205.782 4895.538 7175.556 6726.958 5295.887 5707.282 5473.188 7708.525   
## 209 210 211 212 213 214 215 216   
## 5693.209 5649.076 6257.056 5780.368 5061.088 4751.531 4106.399 7743.296   
## 217 218 219 220 221 222 223 224   
## 4961.277 5081.788 7206.530 8062.876 5782.392 6241.364 6291.227 4589.919   
## 225 226 227 228 229 230 231 232   
## 4580.047 4654.554 5659.104 4434.238 6972.134 6952.801 6333.078 6192.167   
## 233 234 235 236 237 238 239 240   
## 6340.466 6352.271 6664.552 5874.814 6070.996 5474.810 6117.207 5595.180   
## 241 242 243 244 245 246 247 248   
## 6130.922 6810.019 5998.212 6624.769 6127.077 5279.236 4561.728 4846.888   
## 249 250 251 252 253 254 255 256   
## 5066.200 4567.201 4595.334 4985.579 7805.249 5899.868 4031.237 5132.555   
## 257 258 259 260 261 262 263 264   
## 5625.235 5632.222 5116.967 6317.994 5245.476 7352.704 4210.210 6259.941   
## 265 266 267 268 269 270 271 272   
## 5082.407 5862.465 5837.514 6631.704 6365.732 4815.911 5390.740 6764.820   
## 273 274 275 276 277 278 279 280   
## 6379.340 5632.786 6469.736 7485.922 5866.111 7659.884 6707.624 5243.756   
## 281 282 283 284 285 286 287 288   
## 4961.386 7390.564 4837.170 5526.639 4688.465 5487.563 7394.057 4862.131   
## 289 290 291 292 293 294 295 296   
## 4019.402 5004.501 5426.069 5695.537 4564.004 5367.162 7622.582 5315.324   
## 297 298 299 300 301 302 303 304   
## 6533.561 6238.480 5307.588 3994.188 7497.612 4628.031 5489.486 5627.524   
## 305 306 307 308 309 310 311 312   
## 4659.260 5508.112 6488.311 4655.768 6278.060 6771.145 6100.655 4806.042   
## 313 314 315 316 317 318 319 320   
## 5752.947 5577.303 7297.839 5748.635 5644.022 6691.376 5007.900 6056.267   
## 321 322 323 324 325 326 327 328   
## 5067.322 5821.264 4940.887 4934.450 4902.876 5000.403 5043.829 3667.270   
## 329 330 331 332 333 334 335 336   
## 6099.493 4155.191 7223.943 5736.687 4936.282 5975.334 6326.042 6544.189   
## 337 338 339 340 341 342 343 344   
## 7272.584 5352.230 7595.553 4600.295 4873.814 6849.649 4486.109 4748.898   
## 345 346 347 348 349 350 351 352   
## 6987.876 5496.471 6319.117 6357.788 5872.943 4508.178 5322.308 6620.364   
## 353 354 355 356 357 358 359 360   
## 6278.879 5629.083 5952.103 4061.099 6393.816 6495.043 5825.366 6030.253   
## 361 362 363 364 365 366 367 368   
## 5301.354 5090.293 5385.122 5638.495 4656.080 8025.572 4583.896 4795.464   
## 369 370 371 372 373 374 375 376   
## 6643.142 6254.778 6259.941 7191.802 7354.731 6116.863 4342.162 5220.218   
## 377 378 379 380 381 382 383 384   
## 8262.802 7217.766 8105.037 6748.369 7213.931 5057.393 7639.636 5380.058   
## 385 386 387 388 389 390 391 392   
## 5800.462 4705.784 6639.497 4499.977 4924.126 4674.249 6739.056 5567.940   
## 393 394 395 396 397 398 399 400   
## 4565.979 7792.644 5716.949 5385.982 7234.976 5340.986 6707.167 4889.613   
## 401 402 403 404 405 406 407 408   
## 6444.095 6301.252 4971.401 5953.114 6574.864 4932.174 4627.273 6674.725   
## 409 410 411 412 413 414 415 416   
## 5748.531 6131.026 6358.385 4729.826 6588.275 6923.950 5188.896 4203.984   
## 417 418 419 420 421 422 423 424   
## 6252.197 4060.847 7605.422 8261.282 4331.279 6938.829 5668.055 7738.387   
## 425 426 427 428 429 430 431 432   
## 5379.301 4936.981 4692.464 5113.929 5240.970 6422.412 5255.701 6693.250   
## 433 434 435 436 437 438 439 440   
## 6147.628 5721.148 5332.887 6448.782 5985.003 8109.997 5175.831 5566.524   
## 441 442 443 444 445 446 447 448   
## 7034.797 6086.685 4581.466 4379.110 4857.971 5398.180 4809.433 6666.173   
## 449 450 451 452 453 454 455 456   
## 5802.842 5329.800 7517.860 6087.699 6712.988 6566.411 5350.957 5627.116   
## 457 458 459 460 461 462 463 464   
## 5810.991 6111.589 5828.503 4939.309 5895.110 7559.971 4768.639 5001.159   
## 465 466 467 468 469 470 471 472   
## 7639.546 5753.240 6204.263 4806.253 4696.723 6427.677 5631.918 4978.079   
## 473 474 475 476 477 478 479 480   
## 7369.580 5053.394 6910.235 5944.412 7666.817 4171.801 4961.934 5079.005   
## 481 482 483 484 485 486 487 488   
## 6151.424 4433.671 7908.461 5665.878 4170.325 5627.311 5423.243 5040.337   
## 489 490 491 492 493 494 495 496   
## 6957.863 5894.201 5043.424 6609.837 5183.473 5970.173 8139.049 4260.935   
## 497 498 499 500 501 502 503 504   
## 5803.651 5764.525 5166.517 6456.789 6713.700 4981.218 4338.272 6716.480   
## 505 506 507 508 509 510 511 512   
## 6328.170 5667.598 5296.040 6150.006 6091.709 5290.625 6338.505 3813.646   
## 513 514 515 516 517 518 519 520   
## 6625.377 5563.990 7034.655 6108.349 7242.010 4081.903 7324.067 7304.367   
## 521 522 523 524 525 526 527 528   
## 6527.234 6159.573 7674.207 5739.421 5566.371 6199.253 5843.434 6943.335   
## 529 530 531 532 533 534 535 536   
## 6572.889 7118.563 6118.525 4692.110 6376.556 6070.752 6060.822 6788.303   
## 537 538 539 540 541 542 543 544   
## 4396.125 4828.566 6405.405 5816.053 7553.038 5721.605 4819.658 4989.720   
## 545 546 547 548 549 550 551 552   
## 7712.623 4904.536 7417.643 6345.933 7042.437 4595.334 5034.059 7099.126   
## 553 554 555 556 557 558 559 560   
## 6044.070 5830.577 4720.362 5425.207 6027.772 6222.182 7261.497 5516.718   
## 561 562 563 564 565 566 567 568   
## 4838.284 4399.152 5726.262 4755.993 5140.249 6946.118 5874.925 5877.142   
## 569 570 571 572 573 574 575 576   
## 5605.102 4547.816 5462.713 5645.480 5249.727 6094.582 5240.921 6919.143   
## 577 578 579 580 581 582 583 584   
## 5504.622 6235.039 4449.939 5157.659 5580.746 5536.813 6903.807 4450.736   
## 585 586 587 588 589 590 591 592   
## 8333.409 7513.001 7368.043 5789.388 5321.851 4301.020 5093.887 5038.110   
## 593 594 595 596 597 598 599 600   
## 6898.087 4483.882 4832.209 4930.966 6233.733 5412.097 5630.803 3783.792   
## 601 602 603 604 605 606 607 608   
## 5545.367 6019.985 5637.443 6301.343 7354.071 6119.941 6959.329 4954.747   
## 609 610 611 612 613 614 615 616   
## 7535.018 8036.150 5895.768 7001.695 6104.150 6632.208 6228.813 6108.653   
## 617 618 619 620 621 622 623 624   
## 5872.537 5036.286 6772.715 6876.424 7007.625 5637.596 4917.344 5300.899   
## 625 626 627 628 629 630 631 632   
## 6373.062 7568.624 5877.851 5505.685 5823.137 5141.876 8749.053 7301.038   
## 633 634 635 636 637 638 639 640   
## 4853.164 5139.693 4440.209 7773.665 7406.002 6395.292 5472.076 7842.855   
## 641 642 643 644 645 646 647 648   
## 6518.578 7051.660 4886.923 4263.203 8917.499 5964.868 7365.720 7208.908   
## 649 650 651 652 653 654 655 656   
## 4903.838 4677.289 8392.172 5891.415 5096.719 6198.444 6041.284 5590.414   
## 657 658 659 660 661 662 663 664   
## 7074.781 6759.049 6435.226 5477.755 5990.470 3900.003 4627.475 6785.873   
## 665 666 667 668 669 670 671 672   
## 4876.446 5533.268 6612.174 5572.506 4862.477 8246.759 6522.780 3862.438   
## 673 674 675 676 677 678 679 680   
## 5075.055 5310.667 6020.330 6399.182 5865.908 5880.281 6930.075 8866.784   
## 681 682 683 684 685 686 687 688   
## 8024.662 6782.989 6653.518 5429.663 5050.609 5298.470 6446.402 5804.409   
## 689 690 691 692 693 694 695 696   
## 6465.485 8643.027 6670.734 5285.207 6500.258 5027.378 5051.320 5574.023   
## 697 698 699 700 701 702 703 704   
## 4791.820 7986.093 5542.633 6587.262 6783.901 6844.082 5330.054 4187.738   
## 705 706 707 708 709 710 711 712   
## 5772.775 6724.629 6207.916 5280.400 6154.205 4969.171 6382.832 5676.204   
## 713 714 715 716 717 718 719 720   
## 5430.877 5723.386 5117.624 6629.729 7593.071 5807.294 5445.909 4759.630   
## 721 722 723 724 725 726 727 728   
## 5148.853 5644.316 6815.787 5995.793 6577.899 4940.684 6444.531 6222.092   
## 729 730 731 732 733 734 735 736   
## 5772.370 5874.925 4106.399 6650.582 6048.019 5073.135 7720.925 5076.129   
## 737 738 739 740 741 742 743 744   
## 7991.204 6842.968 5346.807 5295.331 4744.444 4338.365 4583.693 8197.661   
## 745 746 747 748 749 750 751 752   
## 6640.762 7032.064 6232.202 6867.818 5480.540 5442.975 5425.308 6299.721   
## 753 754 755 756 757 758 759 760   
## 5200.893 5174.373 4628.840 4456.458 5942.790 5610.506 5489.335 4249.790   
## 761 762 763 764 765 766 767 768   
## 7059.850 5467.825 6835.931 6833.351 5320.342 5129.627 4119.054 8149.175   
## 769 770 771 772 773 774 775 776   
## 6515.551 4699.802 4752.593 6196.114 6276.896 5642.798 5051.066 6427.778   
## 777 778 779 780 781 782 783 784   
## 5515.554 5089.392 7559.919 5982.877 5196.229 6334.300 5391.753 6583.164   
## 785 786 787 788 789 790 791 792   
## 6001.704 7310.187 5777.889 6363.294 4190.277 6765.284 5870.968 6524.399   
## 793 794 795 796 797 798 799 800   
## 7079.842 7484.010 5789.075 5049.902 5845.559 4203.984 6102.983 5211.059   
## 801 802 803 804 805 806 807 808   
## 6973.753 5571.482 5154.270 5913.838 6367.750 4606.976 5306.821 5988.898   
## 809 810 811 812 813 814 815 816   
## 4700.106 7240.747 6768.866 4249.335 6428.233 4157.925 7031.811 5543.036   
## 817 818 819 820 821 822 823 824   
## 5960.962 6967.226 6444.185 4420.816 4533.838 4775.216 5049.252 4528.119   
## 825 826 827 828 829 830 831 832   
## 5716.240 5201.189 4672.320 8735.996 7694.351 6278.008 5899.213 5228.014   
## 833 834 835 836 837 838 839 840   
## 6408.391 5169.807 6122.774 7698.859 5604.283 7689.442 8149.175 4741.103   
## 841 842 843 844 845 846 847 848   
## 6148.639 4647.266 6024.331 4864.805 6866.708 4986.584 4994.733 7258.875   
## 849 850 851 852 853 854 855 856   
## 6369.722 5144.103 6377.830 5628.385 6391.132 5309.098 5824.756 6964.443   
## 857 858 859 860 861 862 863 864   
## 7338.129 6583.679 4838.394 5489.691 6400.850 5350.248 5356.320 5416.603   
## 865 866 867 868 869 870 871 872   
## 5337.393 4019.243 5345.339 5452.642 5460.385 5046.409 6519.389 7012.170   
## 873 874 875 876 877 878 879 880   
## 6816.963 4721.162 4053.060 4499.977 6976.082 7284.576 4600.194 4985.935   
## 881 882 883 884 885 886 887 888   
## 7272.532 6191.156 5703.790 6641.976 5054.558 5322.308 6088.307 5577.557   
## 889 890 891 892 893 894 895 896   
## 7111.578 7691.973 5009.258 6215.449 6643.091 7309.127 5795.653 6258.270   
## 897 898 899 900 901 902 903 904   
## 6101.729 6151.877 5290.420 7561.034 5734.411 6037.641 5620.833 8648.231   
## 905 906 907 908 909 910 911 912   
## 5138.831 5931.554 6205.794 6516.504 4806.700 4594.636 6128.900 7303.712   
## 913 914 915 916 917 918 919 920   
## 5579.683 4547.605 4511.618 3949.395 6813.106 4733.006 6031.719 5723.476   
## 921 922 923 924 925 926 927 928   
## 5323.979 7340.709 5920.924 5803.095 7331.752 5939.550 7268.988 6292.646   
## 929 930 931 932 933 934 935 936   
## 4537.128 6416.846 6312.984 4408.120 7789.760 6253.057 4558.184 7202.380   
## 937 938 939 940 941 942 943 944   
## 5821.115 6135.480 5959.450 5766.297 4892.289 7397.499 4567.951 5709.561   
## 945 946 947 948 949 950 951 952   
## 5892.338 7532.690 6715.724 5857.352 4511.972 5019.289 5589.601 5843.486   
## 953 954 955 956 957 958 959 960   
## 6701.349 5055.065 5452.286 7739.401 7321.173 5807.406 7516.239 6451.820   
## 961 962 963 964 965 966 967 968   
## 5540.609 8124.928 5260.609 5634.446 5617.450 6540.444 4884.343 6810.775   
## 969 970 971 972 973 974 975 976   
## 6629.981 4803.106 5264.263 8211.986 5241.326 5120.155 5729.703 4077.000   
## 977 978 979 980 981 982 983 984   
## 6850.355 6186.754 5829.566 6697.197 5799.602 5918.798 5430.227 5014.320   
## 985 986 987 988 989 990 991 992   
## 4530.193 5612.884 6154.205 4945.889 5978.270 6209.174 8194.423 6251.902   
## 993 994 995 996 997 998 999 1000   
## 8683.155 5967.692 5463.218 3748.110 7189.525 5727.476 6150.614 4889.860   
## 1001 1002 1003 1004 1005 1006 1007 1008   
## 4698.638 6470.700 5817.064 7144.935 6023.115 7142.604 5580.441 5066.756   
## 1009 1010 1011 1012 1013 1014 1015 1016   
## 5506.748 5499.864 5664.108 8100.383 5854.012 6242.224 6326.904 4189.669   
## 1017 1018 1019 1020 1021 1022 1023 1024   
## 6110.839 4437.823 3788.449 5911.055 7346.631 5660.980 7792.138 5873.499   
## 1025 1026 1027 1028 1029 1030 1031 1032   
## 6432.637 4713.468 4221.092 7397.864 4690.895 4384.879 5368.417 5547.693   
## 1033 1034 1035 1036 1037 1038 1039 1040   
## 5707.384 5939.763 5127.090 4409.539 5266.986 5895.768 6921.115 6907.450   
## 1041 1042 1043 1044 1045 1046 1047 1048   
## 5109.779 3855.068 3978.961 5813.117 4065.865 5563.993 8074.112 7694.351   
## 1049 1050 1051 1052 1053 1054 1055 1056   
## 4987.493 6850.610 4735.132 6227.800 4899.729 4902.057 6569.345 6305.087   
## 1057 1058 1059 1060 1061 1062 1063 1064   
## 6353.071 5518.439 5343.972 5122.440 5244.920 4141.482 7326.485 5808.663   
## 1065 1066 1067 1068 1069 1070 1071 1072   
## 6429.449 7933.203 8264.978 7697.843 5373.026 4728.963 4301.020 5168.594   
## 1073 1074 1075 1076 1077 1078 1079 1080   
## 5806.040 6430.410 4862.477 5867.032 5894.910 5009.157 5478.664 6535.637   
## 1081 1082 1083 1084 1085 1086 1087 1088   
## 6399.332 6560.487 5570.422 5357.739 4964.920 6267.238 6699.122 4563.303   
## 1089 1090 1091 1092 1093 1094 1095 1096   
## 5324.132 4483.427 6327.611 6904.716 6293.446 4716.303 5038.622 6831.729   
## 1097 1098 1099 1100 1101 1102 1103 1104   
## 6162.962 6040.881 6080.968 3995.106 6825.150 5493.082 4183.443 5318.868   
## 1105 1106 1107 1108 1109 1110 1111 1112   
## 9129.067 4256.876 6639.801 6863.111 4510.819 4228.280 5133.870 7657.008   
## 1113 1114 1115 1116 1117 1118 1119 1120   
## 6458.147 4241.490 5080.980 8574.031 5868.236 6231.038 6332.774 4295.900   
## 1121 1122 1123 1124 1125 1126 1127 1128   
## 4585.920 3968.383 4677.632 7575.609 4095.163 5043.473 8099.468 5796.312   
## 1129 1130 1131 1132 1133 1134 1135 1136   
## 5658.540 6426.360 4662.399 7340.364 5464.138 5093.785 5694.373 7739.752   
## 1137 1138 1139 1140 1141 1142 1143 1144   
## 5555.135 4222.914 5721.504 6008.286 5947.244 5523.147 5587.640 7605.877   
## 1145 1146 1147 1148 1149 1150 1151 1152   
## 5671.091 8441.928 7345.821 6801.462 5634.558 7062.227 4433.274 5518.246   
## 1153 1154 1155 1156 1157 1158 1159 1160   
## 7284.943 5388.310 5915.558 6046.650 4448.402 6486.793 5234.541 6744.726   
## 1161 1162 1163 1164 1165 1166 1167 1168   
## 5892.379 6601.283 4766.968 5639.105 5859.885 4807.570 7941.047 4507.164   
## 1169 1170 1171 1172 1173 1174 1175 1176   
## 6676.648 5856.645 4688.624 7285.741 5816.607 6797.970 7385.505 7148.781   
## 1177 1178 1179 1180 1181 1182 1183 1184   
## 6139.638 8574.031 6029.240 4834.842 6083.195 3671.074 4580.047 7391.323   
## 1185 1186 1187 1188 1189 1190 1191 1192   
## 3698.811 5596.336 4925.948 4758.313 6842.207 6252.449 6731.362 6290.411   
## 1193 1194 1195 1196 1197 1198 1199 1200   
## 6502.334 5442.266 5663.095 5474.355 7837.642 6132.544 4182.684 6758.846   
## 1201 1202 1203 1204 1205 1206 1207 1208   
## 6696.134 4768.182 5507.507 5463.776 4451.193 5263.088 3966.460 6288.080   
## 1209 1210 1211 1212 1213 1214 1215 1216   
## 6484.223 6016.747 5714.621 5314.513 7281.237 5783.455 7708.777 5082.851   
## 1217 1218 1219 1220 1221 1222 1223 1224   
## 5984.699 4301.568 6311.163 5069.602 5830.327 5140.704 5083.064 7411.467   
## 1225 1226 1227 1228 1229 1230 1231 1232   
## 4858.639 6366.534 5143.185 6709.041 5414.636 6218.689 7180.210 4567.598   
## 1233 1234 1235 1236 1237 1238 1239 1240   
## 4475.985 4995.500 7796.643 7235.888 6310.910 5580.746 6945.918 5438.722   
## 1241 1242 1243 1244 1245 1246 1247 1248   
## 5786.451 7627.288 5753.138 4456.145 5252.510 5844.447 6719.318 6556.994   
## 1249 1250 1251 1252 1253 1254 1255 1256   
## 5364.978 6030.807 6426.006 3697.090 7041.883 7081.007 5110.346 5992.848   
## 1257 1258 1259 1260 1261 1262 1263 1264   
## 4426.656 6067.352 7517.403 7266.408 5839.993 7377.254 6799.489 4873.570   
## 1265 1266 1267 1268 1269 1270 1271 1272   
## 6795.641 5421.766 6833.707 4676.369 4814.251 6074.893 4713.821 6958.620   
## 1273 1274 1275 1276 1277 1278 1279 1280   
## 4639.267 7164.520 6440.734 6953.560 7765.516 6850.761 5299.733 6193.432   
## 1281 1282 1283 1284 1285 1286 1287 1288   
## 5765.497 6814.429 6357.928 5571.139 6066.898 4929.095 5139.498 5596.496   
## 1289 1290 1291 1292 1293 1294 1295 1296   
## 4726.424 6191.914 6413.099 5554.677 7788.645 5941.988 5947.395 8915.576   
## 1297 1298 1299 1300 1301 1302 1303 1304   
## 4911.775 7449.124 5642.697 6404.090 6199.606 4876.446 6262.469 6644.112   
## 1305 1306 1307 1308 1309 1310 1311 1312   
## 8086.260 5447.125 6304.745 5141.616 5989.972 4829.881 4470.570 4647.619   
## 1313 1314 1315 1316 1317 1318 1319 1320   
## 5477.138 5871.321 8137.227 6722.252 3472.101 6023.723 5578.672 4505.817   
## 1321 1322 1323 1324 1325 1326 1327 1328   
## 5967.642 6048.472 4106.399 6845.143 6585.045 6951.545 6800.199 4783.822   
## 1329 1330 1331 1332 1333 1334 1335 1336   
## 7210.784 4758.313 7538.307 4778.711 5895.061 6411.478 4624.944 4075.373   
## 1337 1338 1339 1340 1341 1342 1343 1344   
## 5173.908 5260.004 4946.294 5923.912 5827.593 5125.570 7959.065 5940.512   
## 1345 1346 1347 1348 1349 1350 1351 1352   
## 7348.513 6126.016 7206.733 5192.077 5383.098 5196.179 5906.196 5308.339   
## 1353 1354 1355 1356 1357 1358 1359 1360   
## 3929.613 6247.245 4863.337 5223.965 5892.226 6098.439 7083.436 7301.948   
## 1361 1362 1363 1364 1365 1366 1367 1368   
## 7141.996 5955.393 6444.380 6310.910 6311.264 7249.502 6146.875 4147.355   
## 1369 1370 1371 1372 1373 1374 1375 1376   
## 7447.607 5761.440 4971.609 5972.450 5600.445 5722.263 7140.224 5078.397   
## 1377 1378 1379 1380 1381 1382 1383 1384   
## 6376.302 6860.430 7519.985 6058.294 7037.074 6065.024 5444.794 5014.888   
## 1385 1386 1387 1388 1389 1390 1391 1392   
## 5418.384 5416.858 5543.036 5340.025 7054.587 4609.203 5250.283 5631.816   
## 1393 1394 1395 1396 1397 1398 1399 1400   
## 4934.653 4624.438 6031.264 6438.661 7198.483 6099.491 6618.392 5782.696   
## 1401 1402 1403 1404 1405 1406 1407 1408   
## 5639.711 7813.245 5648.265 4444.663 4980.306 7407.114 5575.078 4841.169   
## 1409 1410 1411 1412 1413 1414 1415 1416   
## 5270.684 5465.039 5970.275 6081.473 6665.716 8749.511 6761.882 5467.277   
## 1417 1418 1419 1420 1421 1422 1423 1424   
## 5870.666 5996.694 4837.778 6946.575 6603.258 5537.673 4987.493 7053.988   
## 1425 1426 1427 1428 1429 1430 1431 1432   
## 5375.513 5886.455 5484.830 7920.194 4057.607 8038.175 5014.632 6504.255   
## 1433 1434 1435 1436 1437 1438 1439 1440   
## 5789.580 4486.109 6774.336 4698.942 4043.898 4222.114 6340.011 5060.834   
## 1441 1442 1443 1444 1445 1446 1447 1448   
## 5285.673 5592.538 5273.566 6190.498 8693.126 4888.087 7510.328 5439.733   
## 1449 1450 1451 1452 1453 1454 1455 1456   
## 5821.173 4130.695 6118.018 5354.551 5301.264 7616.354 6145.703 7927.229   
## 1457 1458 1459 1460 1461 1462 1463 1464   
## 5663.550 6551.987 5854.316 4658.502 5238.441 7263.269 6651.199 4353.913   
## 1465 1466 1467 1468 1469 1470 1471 1472   
## 4677.632 6069.741 6118.574 7268.128 6437.141 5442.214 6616.365 4528.077   
## 1473 1474 1475 1476 1477 1478 1479 1480   
## 6177.641 4260.977 5887.317 6462.349 4795.768 7911.235 6257.358 6243.796   
## 1481 1482 1483 1484 1485 1486 1487 1488   
## 4754.618 5997.302 3960.023 6468.632 6102.276 6509.015 4928.832 5090.646   
## 1489 1490 1491 1492 1493 1494 1495 1496   
## 5414.426 5971.844 5869.812 3603.453 5437.053 5548.857 5566.776 7255.322   
## 1497 1498 1499 1500 1501 1502 1503 1504   
## 7365.813 4883.228 7280.832 7112.490 4336.087 5509.734 5921.540 5282.728   
## 1505 1506 1507 1508 1509 1510 1511 1512   
## 7172.619 5414.275 5564.448 5495.106 8671.615 7207.086 5910.902 7536.029   
## 1513 1514 1515 1516 1517 1518 1519 1520   
## 6858.962 7736.514 8097.951 8004.010 5650.542 4462.985 5217.586 5667.598   
## 1521 1522 1523 1524 1525 1526 1527 1528   
## 4610.013 5768.171 6073.729 5851.988 4400.166 4953.229 4114.600 5902.098   
## 1529 1530 1531 1532 1533 1534 1535 1536   
## 5585.917 5190.307 6311.526 7334.634 4365.302 5535.446 5276.907 4794.957   
## 1537 1538 1539 1540 1541 1542 1543 1544   
## 6219.703 6665.118 5570.876 5910.195 6184.127 4573.369 4959.759 6626.601   
## 1545 1546 1547 1548 1549 1550 1551 1552   
## 5477.845 5288.398 7437.189 6069.681 3618.477 5501.585 6894.592 6525.311   
## 1553 1554 1555 1556 1557 1558 1559 1560   
## 5911.613 4992.859 5876.890 5757.948 8015.651 7645.459 5652.922 4197.667   
## 1561 1562 1563 1564 1565 1566 1567 1568   
## 5363.864 7026.503 5472.835 6209.529 6709.243 5508.783 5234.593 7112.742   
## 1569 1570 1571 1572 1573 1574 1575 1576   
## 7821.241 4855.897 5379.351 8620.292 6175.565 7148.778 5856.341 5001.717   
## 1577 1578 1579 1580 1581 1582 1583 1584   
## 7796.137 7374.116 5394.232 8349.707 7098.822 5264.658 5586.476 8521.998   
## 1585 1586 1587 1588 1589 1590 1591 1592   
## 5098.694 6781.074 6944.502 4642.568 7377.353 6891.100 5054.052 5122.229   
## 1593 1594 1595 1596 1597 1598 1599 1600   
## 5115.348 7687.276 5705.661 6047.359 4677.278 4787.871 6661.870 4812.328   
## 1601 1602 1603 1604 1605 1606 1607 1608   
## 5794.896 5986.115 5473.292 5298.936 6303.214 6945.014 4639.673 6034.209   
## 1609 1610 1611 1612 1613 1614 1615 1616   
## 6846.865 4869.875 6074.337 6428.850 6616.417 6864.226 7498.777 5300.551   
## 1617 1618 1619 1620 1621 1622 1623 1624   
## 4399.152 5674.230 5945.170 5725.098 5314.310 5228.014 5540.913 5385.081   
## 1625 1626 1627 1628 1629 1630 1631 1632   
## 5405.214 7042.590 5016.142 5712.292 6926.736 5369.581 6070.388 4883.078   
## 1633 1634 1635 1636 1637 1638 1639 1640   
## 4723.389 4746.266 3851.920 5165.858 5543.493 5466.661 4539.820 5973.312   
## 1641 1642 1643 1644 1645 1646 1647 1648   
## 6931.754 4607.442 4972.968 7015.207 5432.901 6057.127 8674.500 5524.007   
## 1649 1650 1651 1652 1653 1654 1655 1656   
## 6204.873 4295.394 5661.424 6625.881 6102.983 6286.562 6068.059 7678.053   
## 1657 1658 1659 1660 1661 1662 1663 1664   
## 7041.982 5955.190 4901.906 5969.565 5046.409 7144.881 4322.226 6406.822   
## 1665 1666 1667 1668 1669 1670 1671 1672   
## 5673.177 8996.559 5274.376 5570.674 4735.241 4629.702 5168.087 6812.668   
## 1673 1674 1675 1676 1677 1678 1679 1680   
## 5651.807 6923.446 5125.217 6229.874 5724.947 7407.925 6842.207 6616.275   
## 1681 1682 1683 1684 1685 1686 1687 1688   
## 6023.773 4057.607 5019.634 5992.848 6816.294 4777.445 6513.773 6531.131   
## 1689 1690 1691 1692 1693 1694 1695 1696   
## 5187.321 5525.122 5438.875 7761.780 6951.536 7332.358 4550.996 4106.399   
## 1697 1698 1699 1700 1701 1702 1703 1704   
## 4177.532 5484.385 4048.050 9061.953 4680.982 5559.841 5019.177 3813.646   
## 1705 1706 1707 1708 1709 1710 1711 1712   
## 6752.823 7209.820 5744.798 4912.789 4908.595 6429.348 5716.037 6867.110   
## 1713 1714 1715 1716 1717 1718 1719 1720   
## 4824.567 6183.916 5411.947 7974.654 9193.752 5738.520 5713.711 5361.382   
## 1721 1722 1723 1724 1725 1726 1727 1728   
## 5562.169 5888.783 6175.159 6973.855 4886.671 6712.382 4546.137 6051.967   
## 1729 1730 1731 1732 1733 1734 1735 1736   
## 6008.689 4518.806 4023.697 6596.525 5704.650 5454.724 4900.803 6172.072   
## 1737 1738 1739 1740 1741 1742 1743 1744   
## 5984.242 6210.896 6168.084 7302.545 4261.744 4214.411 5055.572 5265.115   
## 1745 1746 1747 1748 1749 1750 1751 1752   
## 4853.164 6718.252 6806.576 5460.840 4892.955 5300.897 4895.891 7040.221   
## 1753 1754 1755 1756 1757 1758 1759 1760   
## 6783.646 4719.795 6336.267 9257.121 7048.970 4786.211 5748.542 6213.121   
## 1761 1762 1763 1764 1765 1766 1767 1768   
## 6709.496 5327.219 5056.541 6432.432 5162.932 4339.891 5001.008 6266.419   
## 1769 1770 1771 1772 1773 1774 1775 1776   
## 4898.312 3667.270 8263.611 4995.188 5739.421 6222.595 5743.672 5889.191   
## 1777 1778 1779 1780 1781 1782 1783 1784   
## 4984.710 5166.315 5004.862 6338.138 7565.690 7270.102 6702.511 3252.494   
## 1785 1786 1787 1788 1789 1790 1791 1792   
## 6015.673 4435.090 6531.090 6520.553 3596.773 6415.732 5379.968 4776.230   
## 1793 1794 1795 1796 1797 1798 1799 1800   
## 5063.619 4556.310 6195.407 5254.789 5312.995 5480.934 6838.361 4669.080   
## 1801 1802 1803 1804 1805 1806 1807 1808   
## 6339.759 5712.909 5891.569 5904.626 5377.833 5430.726 4808.370 5792.667   
## 1809 1810 1811 1812 1813 1814 1815 1816   
## 5493.334 7180.919 4155.191 5010.677 5391.458 5023.127 4048.659 4743.483   
## 1817 1818 1819 1820 1821 1822 1823 1824   
## 4433.671 5303.337 5052.937 4929.087 5609.950 6241.425 5217.344 5351.412   
## 1825 1826 1827 1828 1829 1830 1831 1832   
## 5946.384 5015.586 6008.842 4906.410 6824.038 6797.112 6007.982 4284.259   
## 1833 1834 1835 1836 1837 1838 1839 1840   
## 5852.544 4996.910 5043.333 6938.577 6645.468 5311.396 5687.999 5661.323   
## 1841 1842 1843 1844 1845 1846 1847 1848   
## 5123.749 4854.276 7667.578 6294.155 7865.935 7004.933 6134.061 6338.088   
## 1849 1850 1851 1852 1853 1854 1855 1856   
## 5146.525 6606.345 4994.075 4482.463 7719.507 6199.609 6795.087 5494.600   
## 1857 1858 1859 1860 1861 1862 1863 1864   
## 5943.650 6458.958 4576.829 5186.105 6009.762 4638.813 5337.494 4902.613   
## 1865 1866 1867 1868 1869 1870 1871 1872   
## 5065.035 5391.247 4887.682 5161.658 6483.402 4770.055 5917.180 6079.248   
## 1873 1874 1875 1876 1877 1878 1879 1880   
## 4561.676 8369.700 5738.714 4202.162 5429.357 5064.022 7266.408 7500.398   
## 1881 1882 1883 1884 1885 1886 1887 1888   
## 5601.144 6881.535 5821.318 5743.176 3744.414 4644.784 4930.106 6430.309   
## 1889 1890 1891 1892 1893 1894 1895 1896   
## 6272.593 8817.992 5751.114 5301.962 5428.244 6472.013 6440.836 5780.370   
## 1897 1898 1899 1900 1901 1902 1903 1904   
## 4783.225 6834.312 8083.170 4384.837 7860.367 7583.709 4441.973 4888.087   
## 1905 1906 1907 1908 1909 1910 1911 1912   
## 4879.939 7221.713 7161.079 5844.193 7163.912 4997.820 5249.981 5675.040   
## 1913 1914 1915 1916 1917 1918 1919 1920   
## 5767.310 6386.982 5254.646 4137.883 4852.810 5808.458 5005.665 4779.976   
## 1921 1922 1923 1924 1925 1926 1927 1928   
## 5931.856 3743.149 5795.047 5143.851 4413.850 5874.715 5252.512 5232.520   
## 1929 1930 1931 1932 1933 1934 1935 1936   
## 5535.700 4911.625 5800.564 6322.297 5610.210 7567.460 7878.842 4846.433   
## 1937 1938 1939 1940 1941 1942 1943 1944   
## 4727.084 6688.694 7200.759 6468.827 8094.814 6497.676 6613.379 6187.715   
## 1945 1946 1947 1948 1949 1950 1951 1952   
## 4329.863 4746.773 3667.270 6384.856 3716.626 5445.605 6666.072 6366.129   
## 1953 1954 1955 1956 1957 1958 1959 1960   
## 5253.625 6738.651 6256.700 6064.367 5157.618 6968.390 5253.271 5503.365   
## 1961 1962 1963 1964 1965 1966 1967 1968   
## 5885.496 6756.973 4817.988 4872.195 5532.510 8189.260 3914.471 5449.199   
## 1969 1970 1971 1972 1973 1974 1975 1976   
## 4486.109 5041.904 5656.565 4727.287 4450.029 5010.683 6530.019 6163.518   
## 1977 1978 1979 1980 1981 1982 1983 1984   
## 5422.779 7405.952 5648.972 7796.695 5375.809 5364.976 7974.958 7254.109   
## 1985 1986 1987 1988 1989 1990 1991 1992   
## 5202.403 6278.718 6216.361 5681.012 6621.024 5557.006 7533.246 4738.733   
## 1993 1994 1995 1996 1997 1998 1999 2000   
## 6403.329 6547.024 5885.748 6549.303 6724.125 6019.623 5372.377 4544.822   
## 2001 2002 2003 2004 2005 2006 2007 2008   
## 5578.721 5138.123 5254.334 6193.079 5867.273 5140.958 8004.475 7508.042   
## 2009 2010 2011 2012 2013 2014 2015 2016   
## 5422.374 4618.921 5554.677 5701.412 4008.815 6013.600 7581.430 6420.791   
## 2017 2018 2019 2020 2021 2022 2023 2024   
## 5727.223 3991.564 7256.182 4531.255 5213.033 6541.154 4771.674 7580.016   
## 2025 2026 2027 2028 2029 2030 2031 2032   
## 4417.789 5375.708 5949.422 6298.557 6279.326 4379.110 4519.058 5573.711   
## 2033 2034 2035 2036 2037 2038 2039 2040   
## 5073.792 5386.336 4835.104 4934.806 5439.735 5611.013 5897.389 4366.466   
## 2041 2042 2043 2044 2045 2046 2047 2048   
## 5071.464 8059.798 6173.946 7653.606 5212.932 5632.220 5005.262 5991.432   
## 2049 2050 2051 2052 2053 2054 2055 2056   
## 7565.132 7655.885 5552.198 6422.412 8611.080 6482.490 5591.930 7473.533   
## 2057 2058 2059 2060 2061 2062 2063 2064   
## 7582.594 6827.387 4433.671 4525.953 4240.485 4399.152 4006.405 5575.542   
## 2065 2066 2067 2068 2069 2070 2071 2072   
## 4828.727 5577.960 7825.897 4274.245 6784.367 7792.543 6196.267 5446.618   
## 2073 2074 2075 2076 2077 2078 2079 2080   
## 7880.359 6238.631 6581.392 6976.336 5407.290 6559.780 5684.263 5054.558   
## 2081 2082 2083 2084 2085 2086 2087 2088   
## 5755.924 6076.208 6834.769 6204.974 4700.572 4864.805 5473.645 5259.497   
## 2089 2090 2091 2092 2093 2094 2095 2096   
## 6619.819 5021.053 5095.303 5395.953 4745.458 7450.795 5613.492 4440.464   
## 2097 2098 2099 2100 2101 2102 2103 2104   
## 7744.865 5175.173 7965.593 4523.665 5638.405 4479.387 4110.204 6659.643   
## 2105 2106 2107 2108 2109 2110 2111 2112   
## 8817.992 5594.715 4669.080 6893.735 4298.944 7390.512 4496.586 5407.846   
## 2113 2114 2115 2116 2117 2118 2119 2120   
## 6006.360 5062.707 3862.438 6169.089 5326.965 6026.608 4518.965 5101.074   
## 2121 2122 2123 2124 2125 2126 2127 2128   
## 6274.214 6832.894 5514.390 6079.449 5350.754 7175.515 4628.840 8637.097   
## 2129 2130 2131 2132 2133 2134 2135 2136   
## 8588.304 4560.512 5869.754 7755.140 5983.181 4274.155 5755.316 6045.538   
## 2137 2138 2139 2140 2141   
## 3375.081 6716.480 5754.962 7998.443 5833.362