IS621_hw5

Charley Ferrari

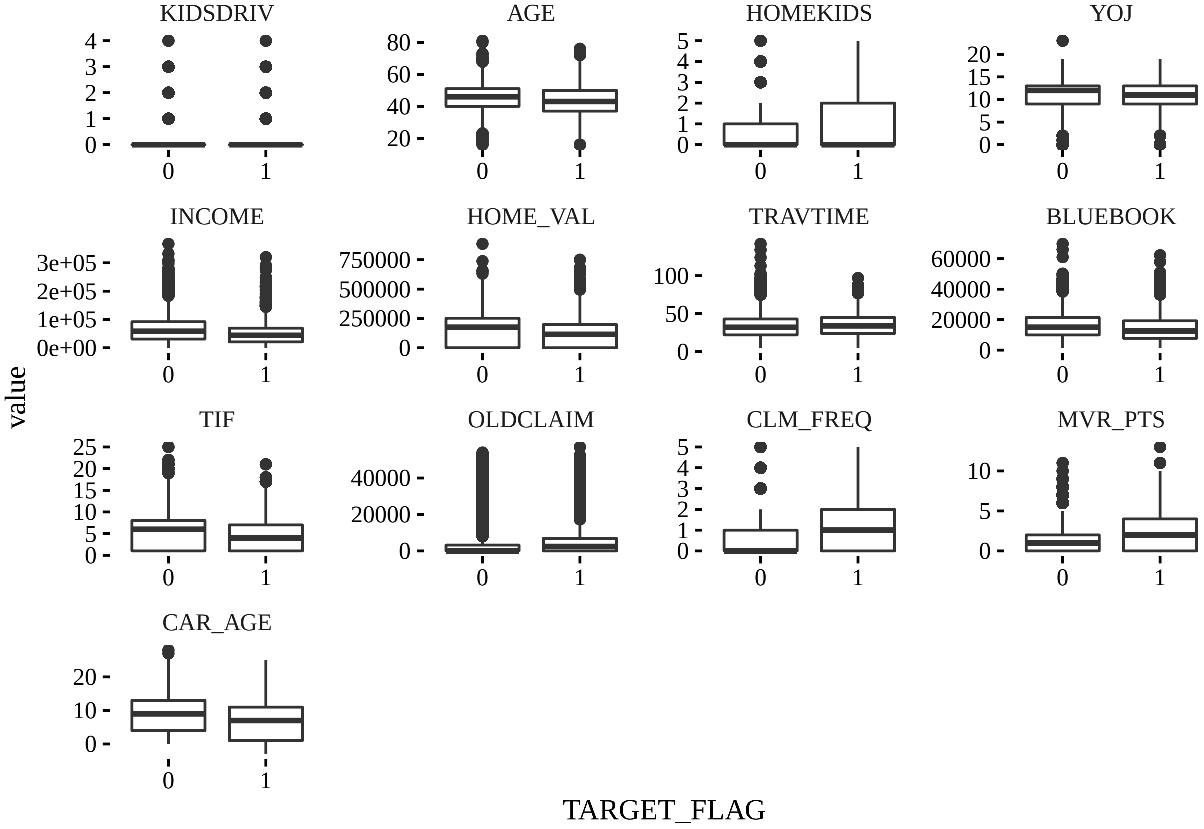
April 3, 2016

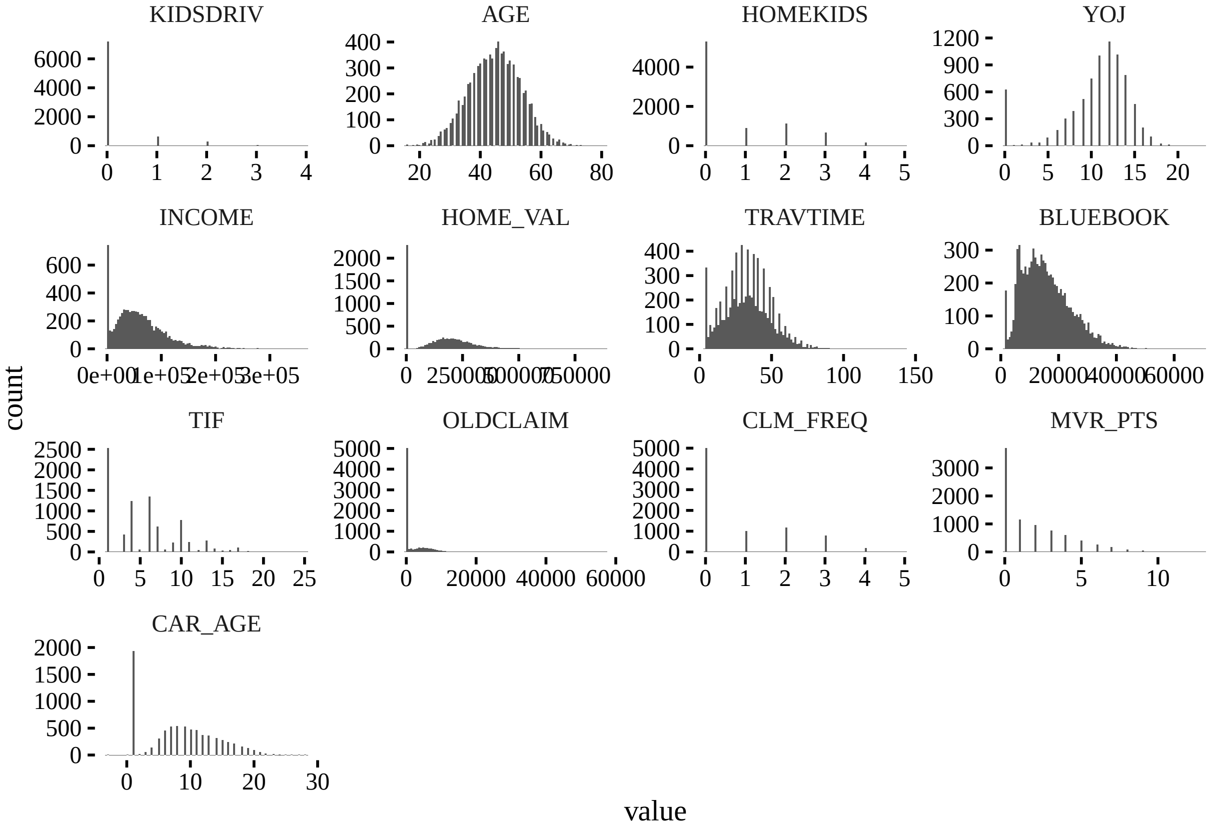
Data Exploration - Logistic Focus

This insurance dataset includes 8161 observations of 24 variables.

There are missing values, but these are localized in the YOJ (Years on Job) and CAR.AGE variables. I'll start my analysis by omitting NAs. There are about 1000 missing values, so I'll still have more than enough data to create a model. If variables with NAs are eliminated, I'll reintroduce the observations if there are no other NAs. So, if I end up removing YOJ or CAR.AGE, I'll add the observations they eliminated back to my model.

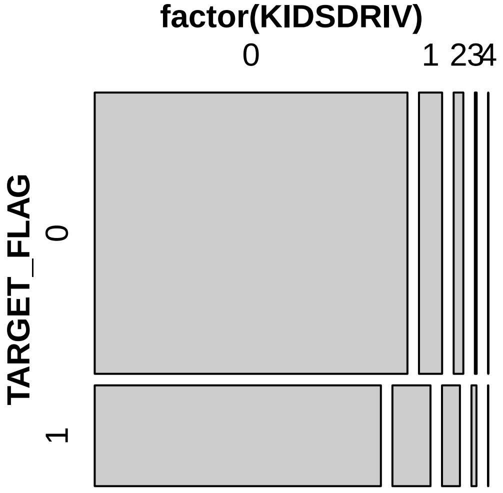
First lets examine our numeric data. To check their distributions along with how they affect the dependent binary target of whether or not a person has been in a crash, I'll plot duel boxplots for each variable, as well as histograms.

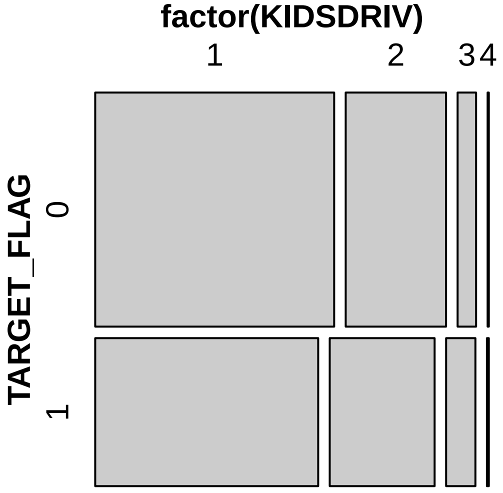


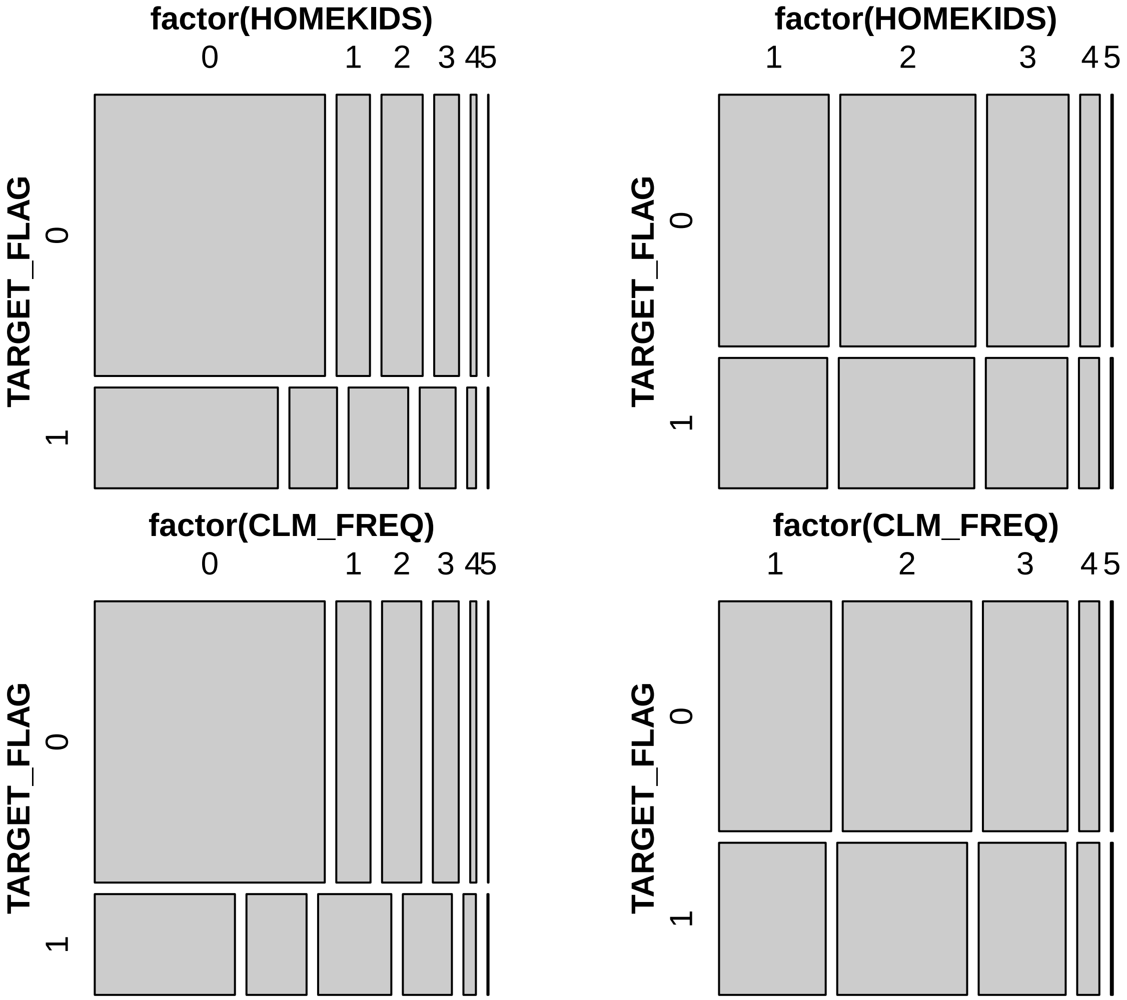


There are a few variables that need some treatment. First, it appears we have count data with low means: KIDSDRIV, HOMEKIDS, and CLM_FREQ. In each of these cases, it seems like there are large numbers of 0 values.

This may imply that it makes more sense to look at these variables as binary. When considering the number of kids a person has, for example, whether or not they have kids might be more important. To analyze this, I'm going to compare Mosaic plots of each of these variables compared to TARGET_FLAG. I'm then going to plot a second Mosaic plot with the '0' variables excluded. This should tell me if there's a numeric relationship, or a categorical one.

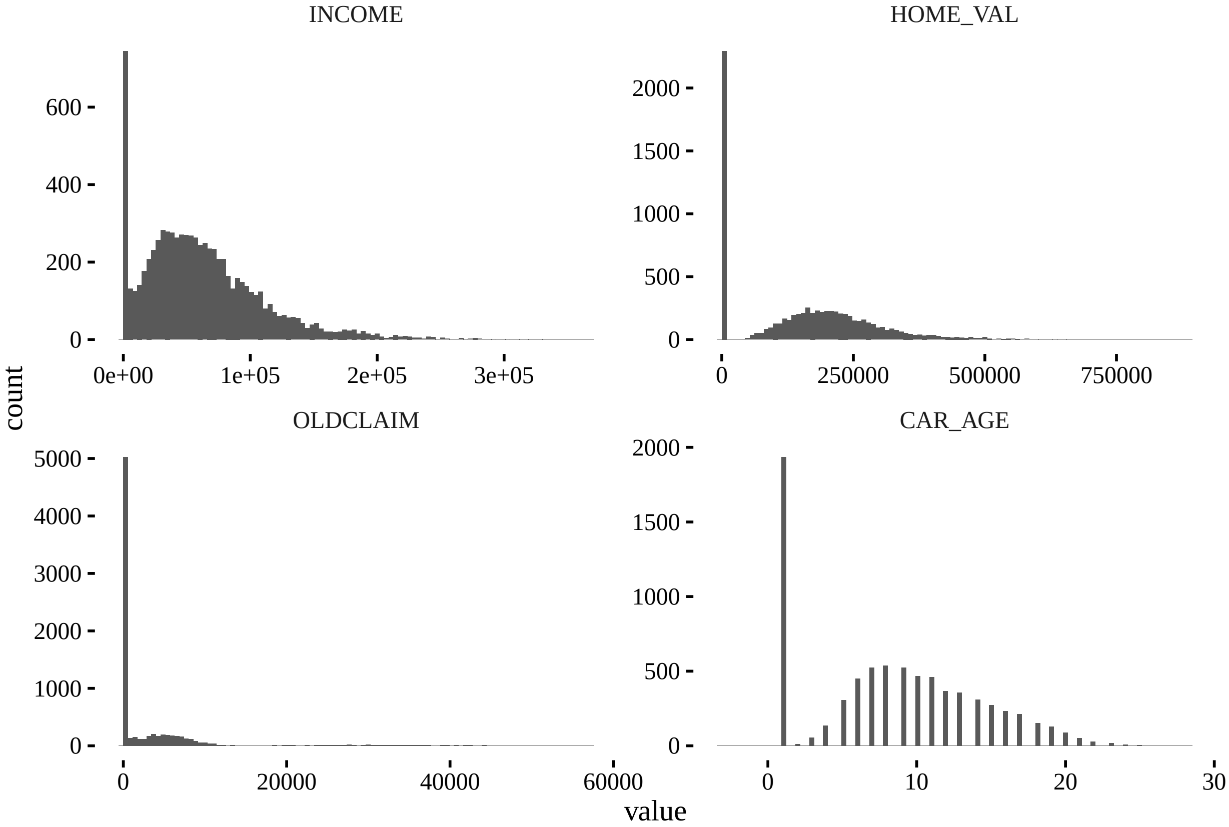






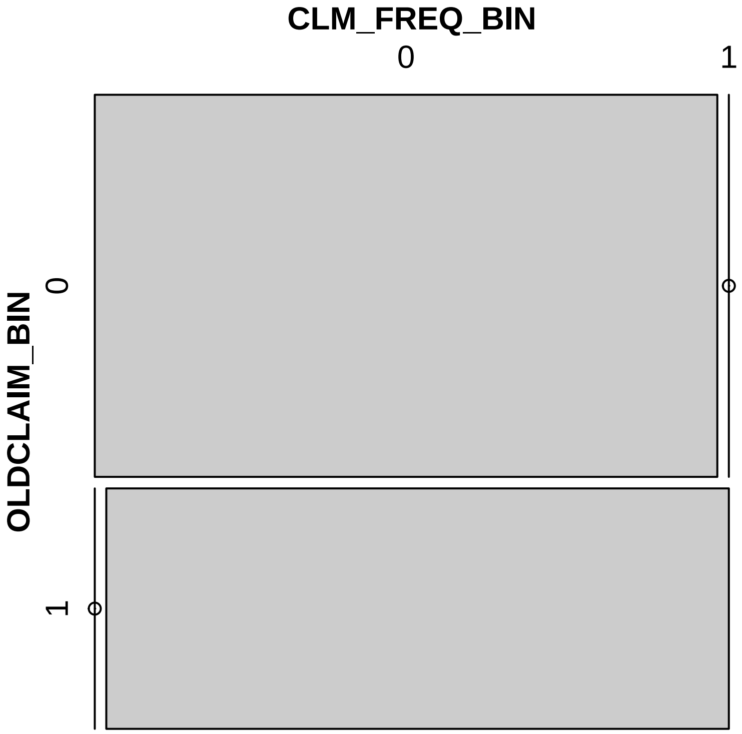
It would appear that number of kids at home and claim frequency have negligible effects once you filter out 0 values. What matters is whether a claim has been filed in the past, or whether there are any kids at home. For kids who drive, there is a bit more of a relationship remaining after filtering for 0 kids, implying that the number of driving kids matters. This intuitively makes sense, if kids are more likely to get into crashes, then more driving children would lead to more crashes.

Next, there are a few suspcious spikes in the data. In particular INCOME, HOME.VAL, OLDCLAIM, and CAR.AGE:

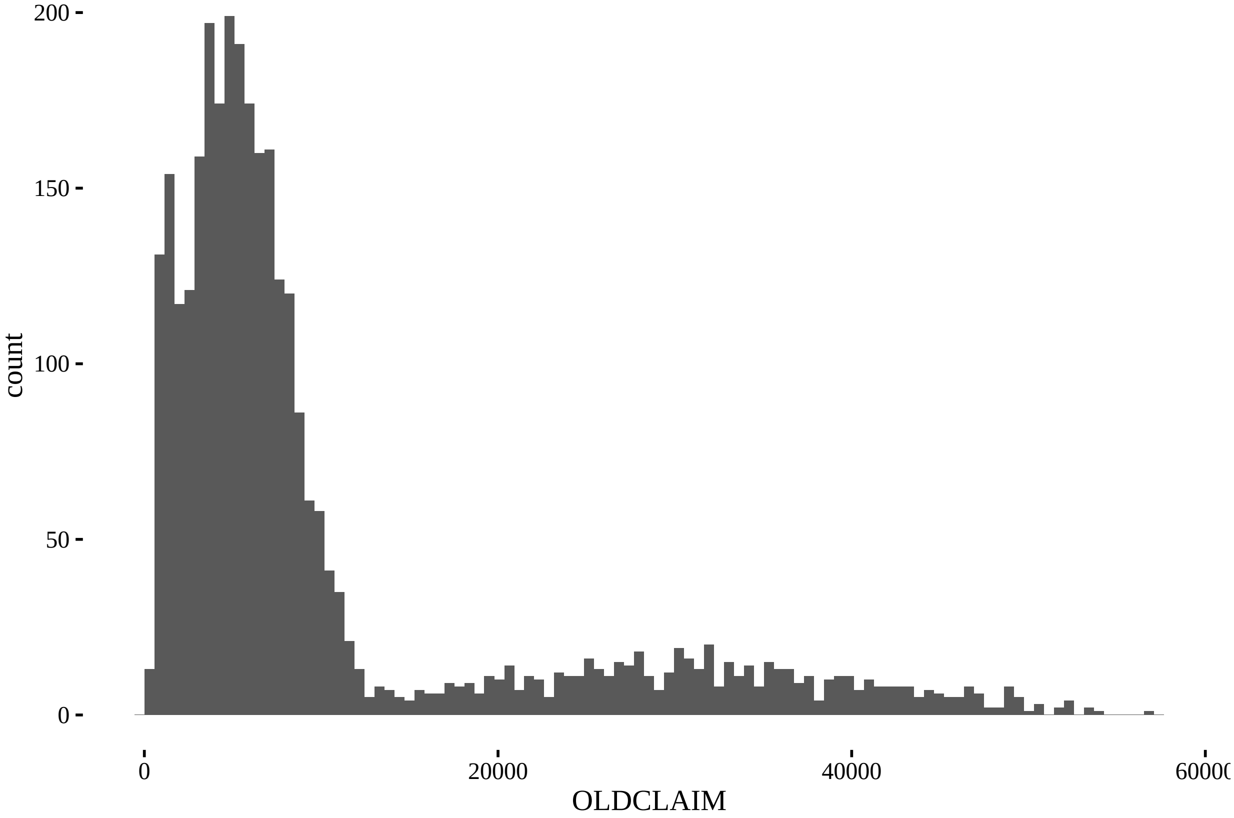


It is possible to deduce what could be going on with a few variables. HOME.VAL, for example, is probably showing 0 values due to renters being included in the sample. Similarly, OLDCLAIM is probably correctly distributed, and reflecting the fact that a number of people aren't filing claims. For INCOME and CAR.AGE, it's more likely that these low values represent missing values (Income could perhaps refer to people who are unemployed, but the number of missing values is too high.)

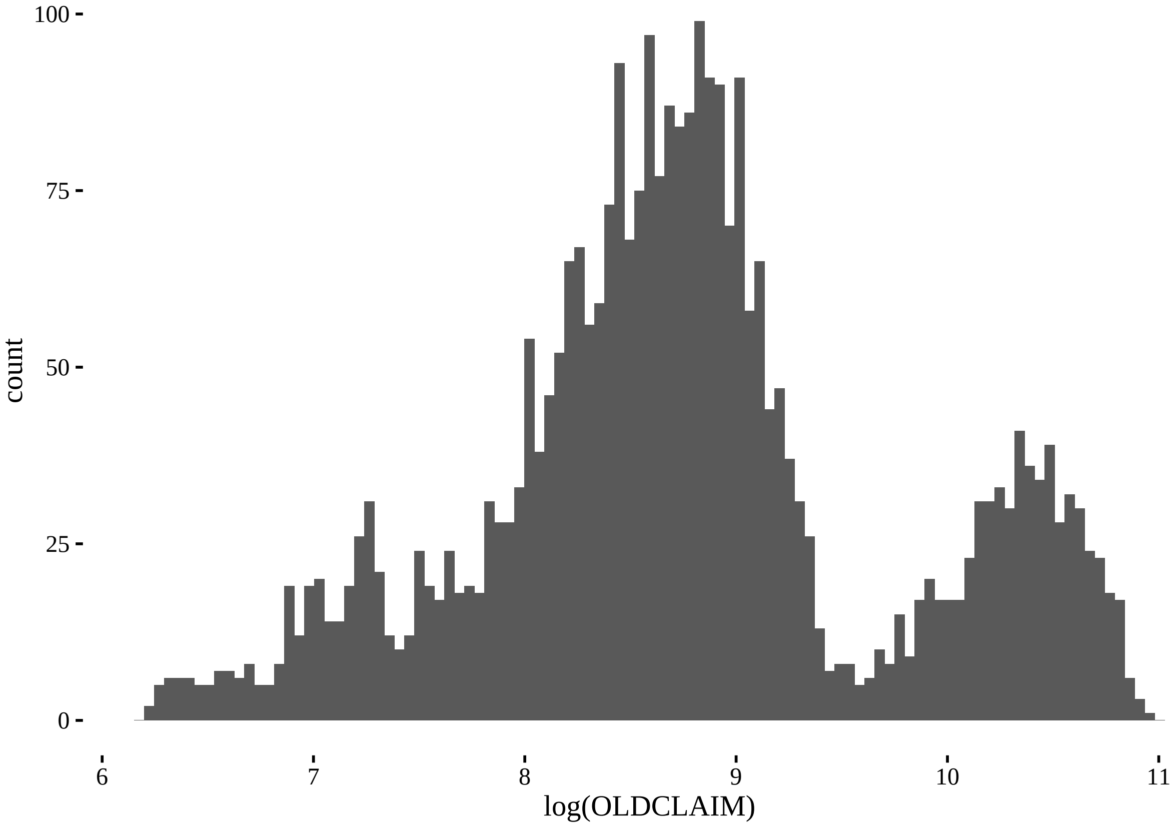
Performing a sanity check comparison between a constructed binary variable from CLM.FREQ (0 if there are 0 claims, 1 if there's more than 0 claims) and a constructed binary variable from OLDCLAIM (0 if the OLDCLAIM amount is 0, 1 if it's greater than 0) shows us these two groups overlap, which is expected.



I'd like to look at the distribution of OLDCLAIM more closely. I'll replot the histogram filtering out the zero values:

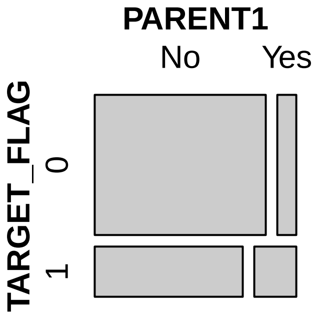


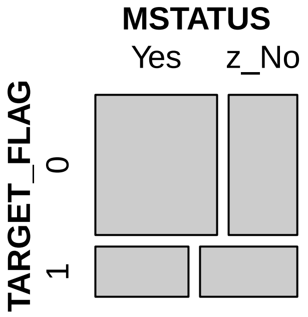
There seems to be a skewed distribution, I'll take the log of this variable to try to make the distribution normal.

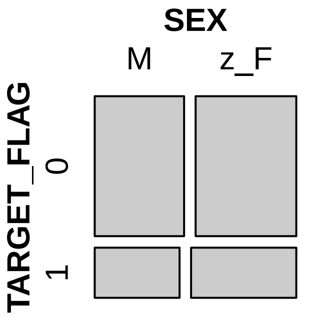


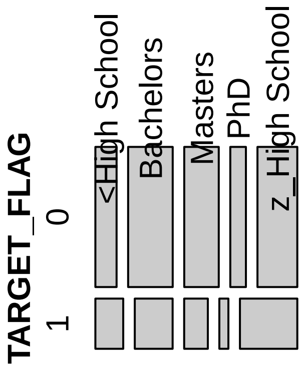
With the log, it makes it clear that there is bimodality in the data. I will have to deal with this in the data transformation section.

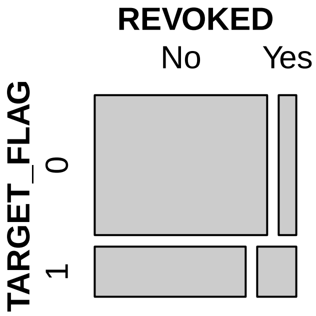
Now, to check our categorical variables. I'll look at these with a series of mosaic plots, comparing them to the TARGET.FLAG one by one:

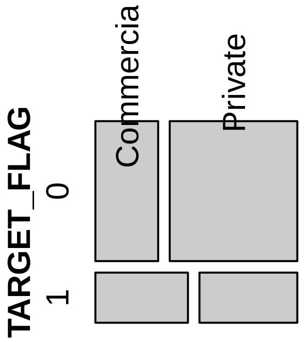

PARENT1
No
Yes
TARGET_FLAG
0
1

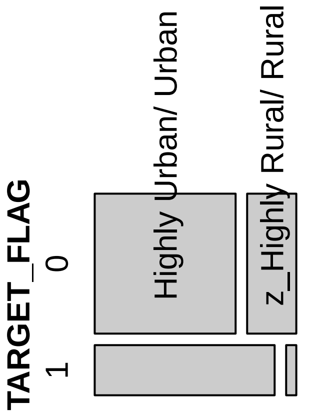

MSTATUS
Yes
z_No
TARGET_FLAG
0
1

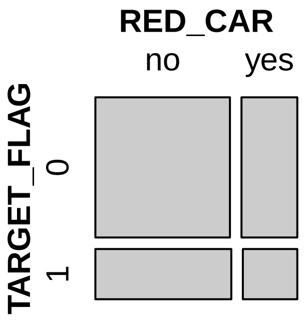

SEX
M
z_F
TARGET_FLAG
0
1

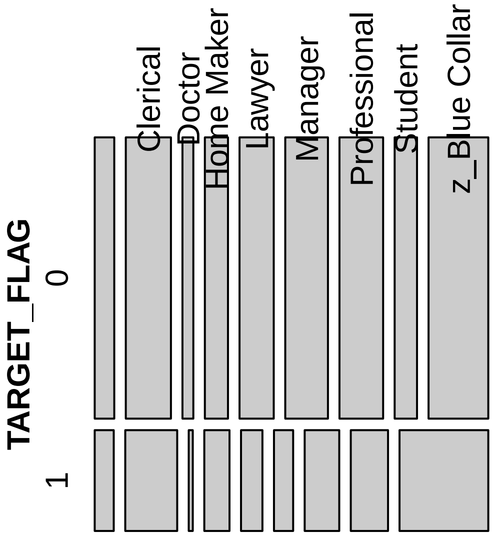

<High School
Bachelors
Masters
PhD
z_High School
TARGET_FLAG
0
1

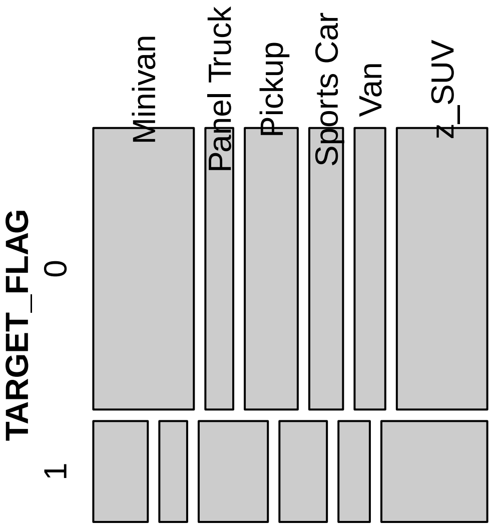

REVOKED
No
Yes
TARGET_FLAG
0
1


Commercia
Private
TARGET_FLAG
0
1


Highly Urban/ Urban
z_Highly Rural/ Rural
TARGET_FLAG
0
1


RED_CAR
no
yes
TARGET_FLAG
0
1





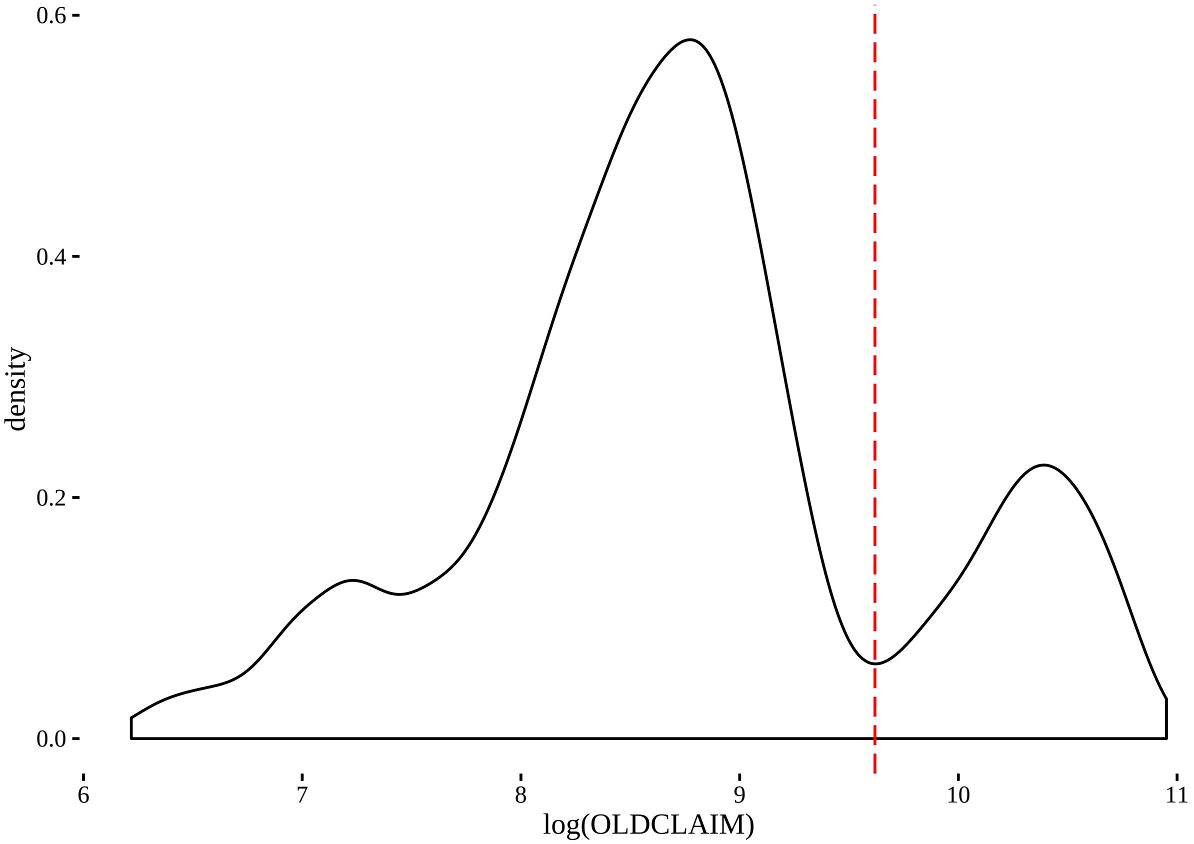
I'm not seeing any problems with these categorical variables. It would appear we have a variety of types of relationships between these variables.

Data Transformation - Logistic Focus

Most of my data transformations will involve converting complicated numeric variables into categorical ones. As explained above, several numeric variables represent both a numeric measure and a category. Home value represents the value of someone's home if they own a home, and hidden in both the OLDCLAIM and CLM_FREQ variables is a binary variable of whether or not someone submitted a claim in the past.

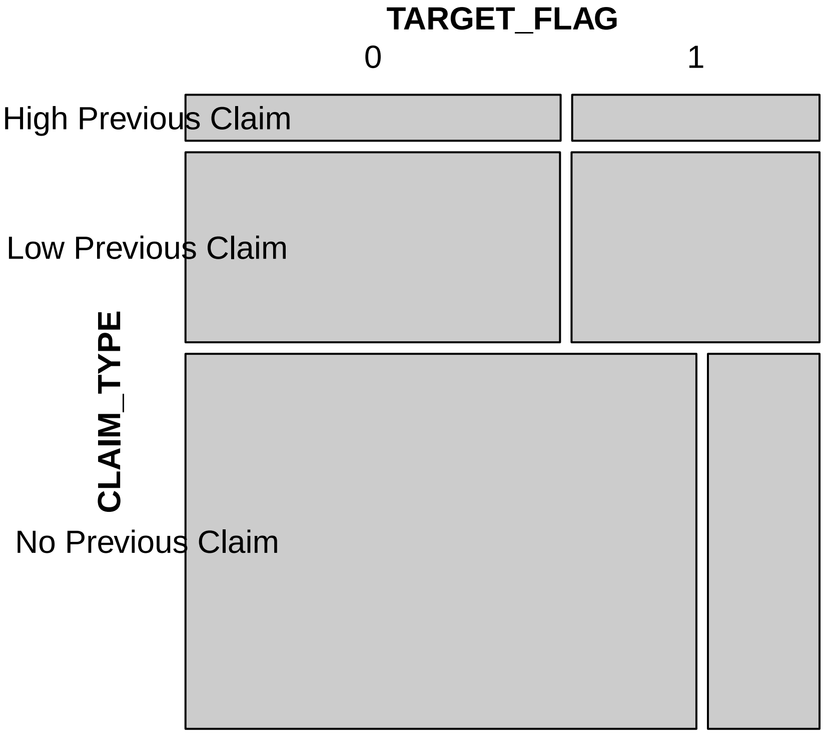
As shown in the mosaic plots above, once 0 values are filtered out, the number of claims doesn't seem to correlate with the target flag. For this reason, I feel comfortable reducing this variable to a binary categorical one: 0 if no claim has been filed, and 1 if at least 1 claim has been filed. We also made sure that this binary variable maps to OLDCLAIM, so going forward we can calculate this binary variable from OLDCLAIM (0 if OLDCLAIM is 0, and 1 if there is a non-zero previous claim amount) and remove the variable CLM_FREQ.

We also previously discovered a bimodal distribution with the log of OLDCLAIM. To analyze this further, I will plot the kernel density of the log of OLDCLAIM:



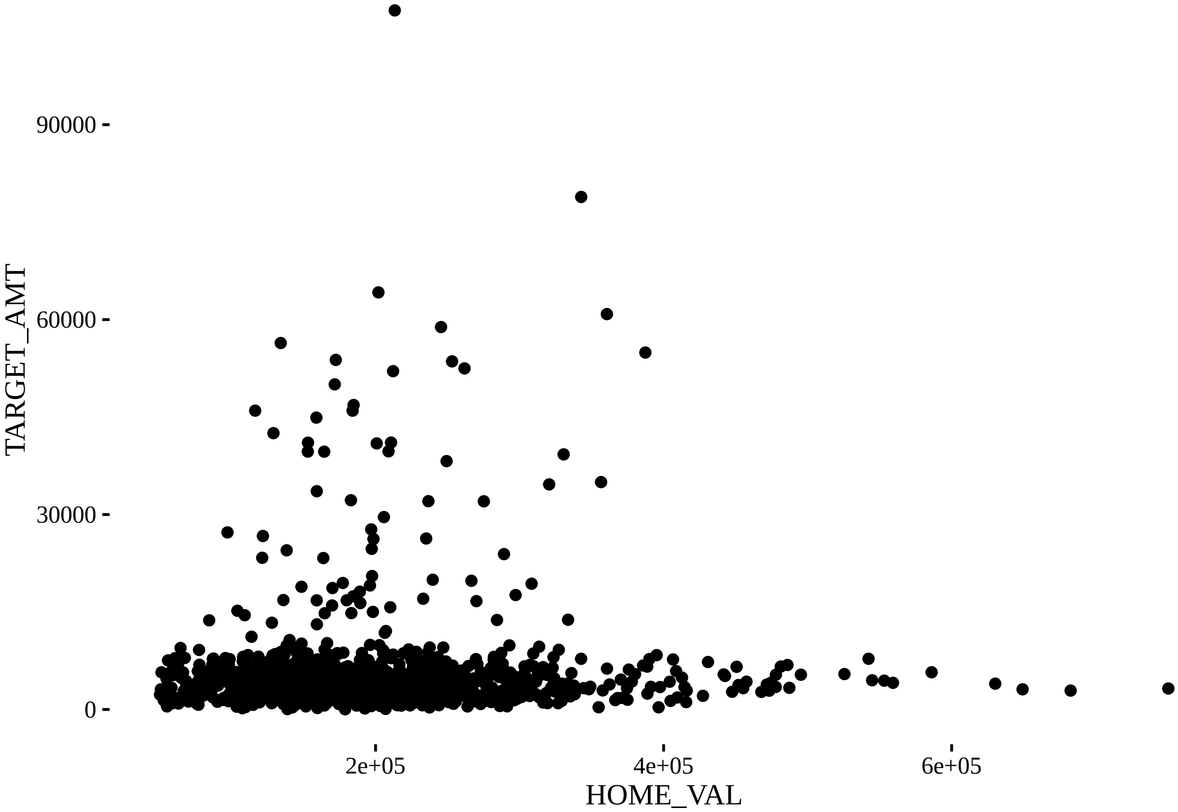
Using the local minimum (of log(x) = 9.62) I can divide the data between its two modes. Since I'm already forced to divide the data categorically to take out 0 values, further dividing the data into low and high claim amounts will allow me to keep some information about the claim amount without having to split the data between modes.

So, the two variables: CLM.FREQ and OLDCLAIM, will be divided into a 3 level categorical variable: no previous claim, low previous claim, and high previous claim. No previous claim will include all OLDCLAIMS equal to 0, low previous claims will include non-zero claim amounts less than or equal to 9.62, and high previous claims will include claim amounts greater than 9.62.

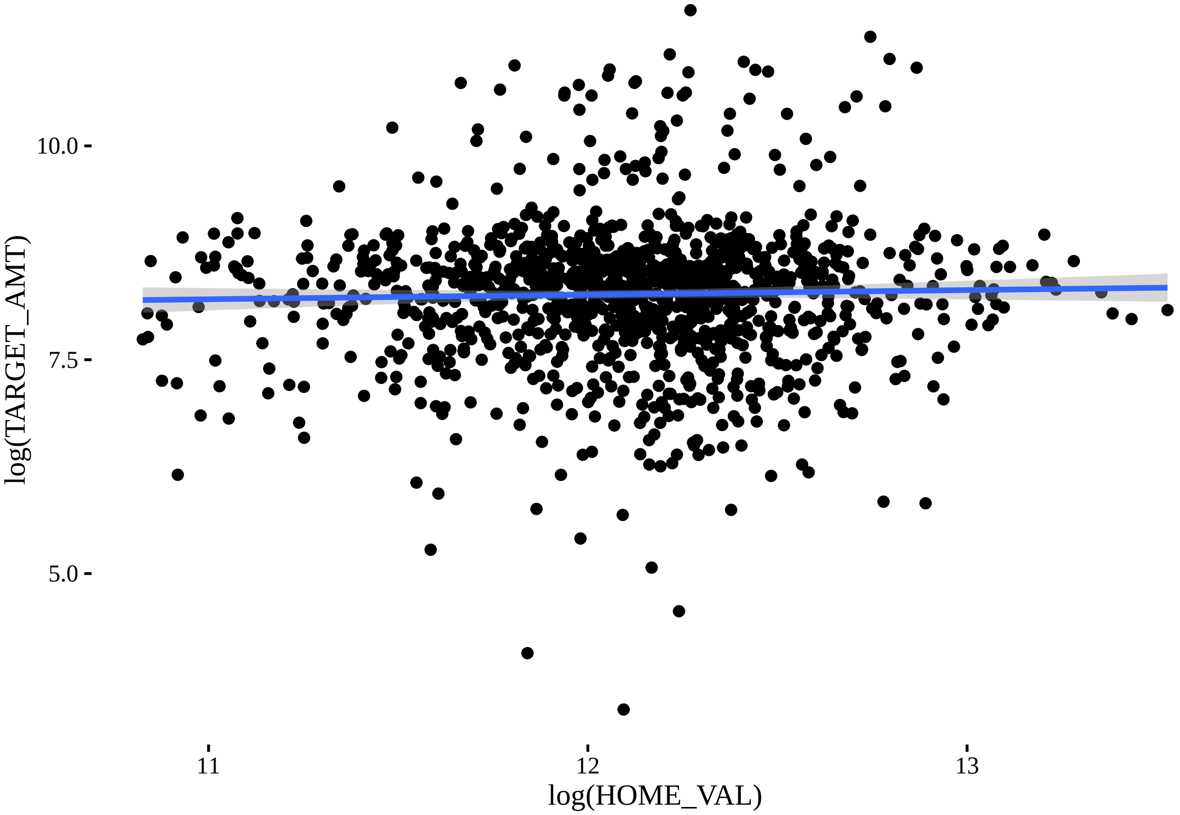


Interestingly, it would appear that whether or not there is a high previous claim or low previous claim doesn't really affect the TARGET.FLAG.

The HOME.VAL variable potentially contains too much information to reduce to a categorical variable without further investigation. To try to get an idea of how the home value can affect my outcome, I'll first look at a scatter plot of HOME.VAL and TARGET.AMT, filtering for observations where there has been a crash.



It would appear both variables have skewed distributions. Taking the log of both and replotting will give me a clearer idea of their relationship:



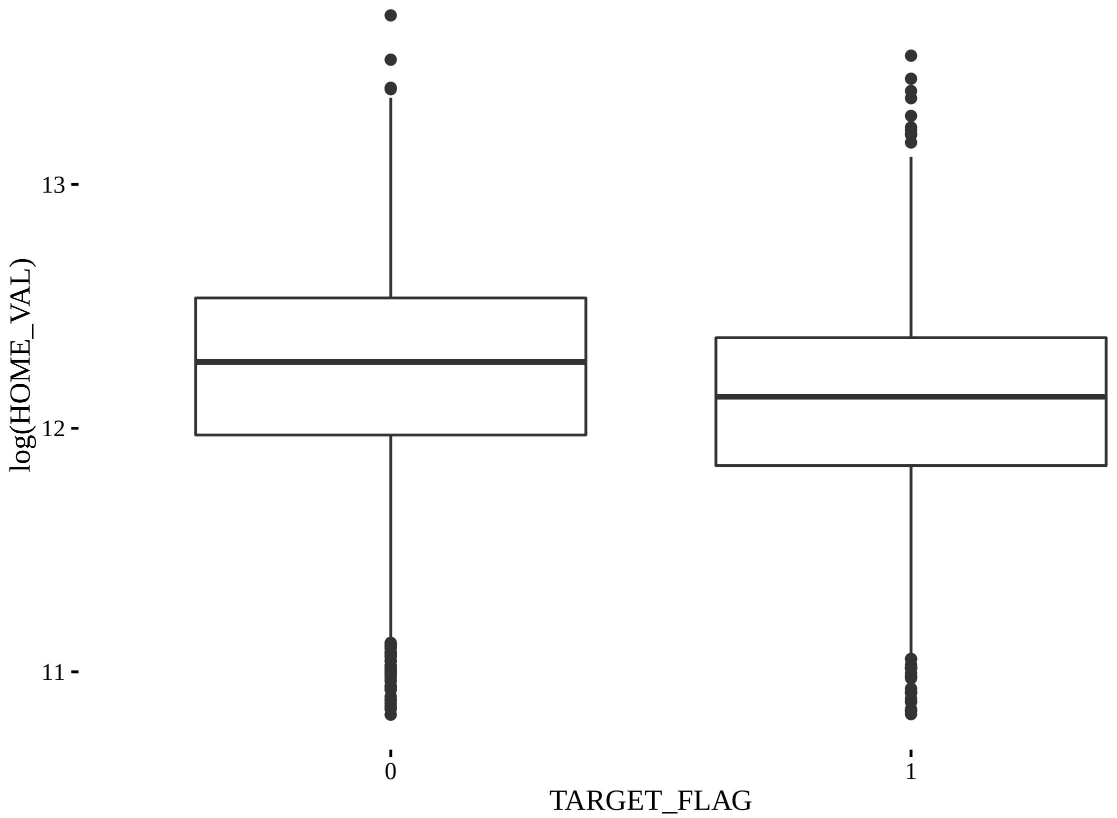
It would appear that there is not a significant relationship between these two variables. I'll further test this by analyzing an OLS model between these two variables:

```
##
## Call:
## lm(formula = log(TARGET_AMT) ~ log(HOME_VAL), data = filter(insurance,
##     HOME_VAL != 0 & TARGET_AMT != 0))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.8552 -0.4143  0.0657  0.4015  3.3110
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    7.61825    0.67879  11.223   <2e-16 ***
## log(HOME_VAL)  0.05352    0.05605   0.955     0.34
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8369 on 1184 degrees of freedom
## Multiple R-squared:  0.0007694,  Adjusted R-squared:  -7.45e-05
## F-statistic: 0.9117 on 1 and 1184 DF,  p-value: 0.3399
```

Home Values does indeed have a high p value, and my adjusted R-squared is extremely small.

Since I would have to subset my data to take home value into account, I think it would be better to just deal with home value as a categorical when looking at the TARGET.AMT.

There is, however, a relationship between HOME.VALUE and TARGET.FLAG. The boxplot below shows this effect visually:



And the Analysis of Variance test shows differing means with a very low p-value (along with a two sample p-test in the R code appendix.)

```
##               Df Sum Sq Mean Sq F value Pr(>F)
## TARGET_FLAG    1   18.2   18.21   95.94 <2e-16 ***
## Residuals   5401 1024.9    0.19
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The other problematic variables will be dealt with in fairly standard ways. KIDSDRIV and HOMEKIDS will be changed into binary variables, with 0 representing no kids and 1 representing kids. For INCOME and CAR_AGE I would need more information to deal with 0 values, so I will assume for now that these 0 values are mistakes, and treat them as NAs.

These findings will affect how I build my models.

Build Models - Logistic Focus

Because of the factors listed above, I will begin building my models in stages.

I have established that the HOME.VAL variable affects the TARGET.FLAG, but doesn't affect the TARGET.AMT. I will split the data set into two pieces depending on whether or not the HOME.VAL is 0. If the HOME.VAL is 0, I will omit the HOME.VAL variable and create a model, and if the HOME.VAL is greater than 0 I will have it in my model. By splitting the model in this way, I hope to capture the effect of renters vs homeowners, while also capturing the effect of the value of a homeowner's home.

There are 464 values of HOME.VAL that are NA, and I will omit these observations from my model.

We have too many variables to perform a best subset selection process, so I will perform a stepwise selection method:

```
##
## Call:
## glm(formula = TARGET_FLAG ~ URBANICITY + JOB + PARENT1 + CAR_TYPE +
##     MVR_PTS + TRAVTIME + CAR_USE + REVOKED + KIDSDRIV_BIN + BLUEBOOK +
##     INCOME + TIF + MSTATUS + RED_CAR, family = "binomial", data = insurancehv0)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4235  -0.7645  -0.4269   0.8220   2.6755
##
## Coefficients:
##                                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)                     -1.669e+00  6.472e-01  -2.579 0.009898 **
## URBANICITYz_Highly Rural/ Rural -2.654e+00  2.870e-01  -9.247  < 2e-16 ***
## JOBClerical                      9.241e-01  3.806e-01   2.428 0.015176 *
## JOBDoctor                       -3.209e-01  4.625e-01  -0.694 0.487870
## JOBHome Maker                   -3.625e-01  5.902e-01  -0.614 0.539098
## JOBLawyer                        2.806e-01  3.496e-01   0.803 0.422208
## JOBManager                      -7.977e-01  3.355e-01  -2.378 0.017430 *
## JOBProfessional                  1.923e-01  3.226e-01   0.596 0.551235
## JOBStudent                       4.345e-01  4.544e-01   0.956 0.338989
## JOBz_Blue Collar                 5.463e-01  3.172e-01   1.722 0.085044 .
## PARENT1Yes                       5.933e-01  1.817e-01   3.266 0.001091 **
## CAR_TYPEPanel Truck              1.027e+00  3.716e-01   2.763 0.005725 **
## CAR_TYPEPickup                   6.398e-01  2.499e-01   2.560 0.010473 *
## CAR_TYPESports Car               1.112e+00  2.807e-01   3.963 7.41e-05 ***
## CAR_TYPEVan                      7.505e-01  2.888e-01   2.599 0.009352 **
## CAR_TYPEz_SUV                    5.304e-01  2.268e-01   2.338 0.019378 *
## MVR_PTS                          1.222e-01  3.017e-02   4.050 5.13e-05 ***
## TRAVTIME                         2.080e-02  4.893e-03   4.250 2.14e-05 ***
## CAR_USEPrivate                  -7.896e-01  2.295e-01  -3.440 0.000582 ***
## REVOKEDYes                       6.331e-01  1.939e-01   3.264 0.001097 **
## KIDSDRIV_BIN                     8.377e-01  2.610e-01   3.210 0.001329 **
## BLUEBOOK                        -2.812e-05  1.187e-05  -2.368 0.017892 *
## INCOME                          -4.404e-06  2.134e-06  -2.064 0.039042 *
## TIF                             -3.268e-02  1.773e-02  -1.843 0.065349 .
## MSTATUSz_No                      7.187e-01  4.267e-01   1.684 0.092148 .
## RED_CARyes                      -2.798e-01  1.916e-01  -1.461 0.144143
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1538.3  on 1214  degrees of freedom
## Residual deviance: 1187.6  on 1189  degrees of freedom
## AIC: 1239.6
##
## Number of Fisher Scoring iterations: 5
```

I came to the same model using both forward and backward stepping. So, when HOME.VAL is 0, I will use model0.

Lets do the same thing when there is a positive value for HOME.VAL:

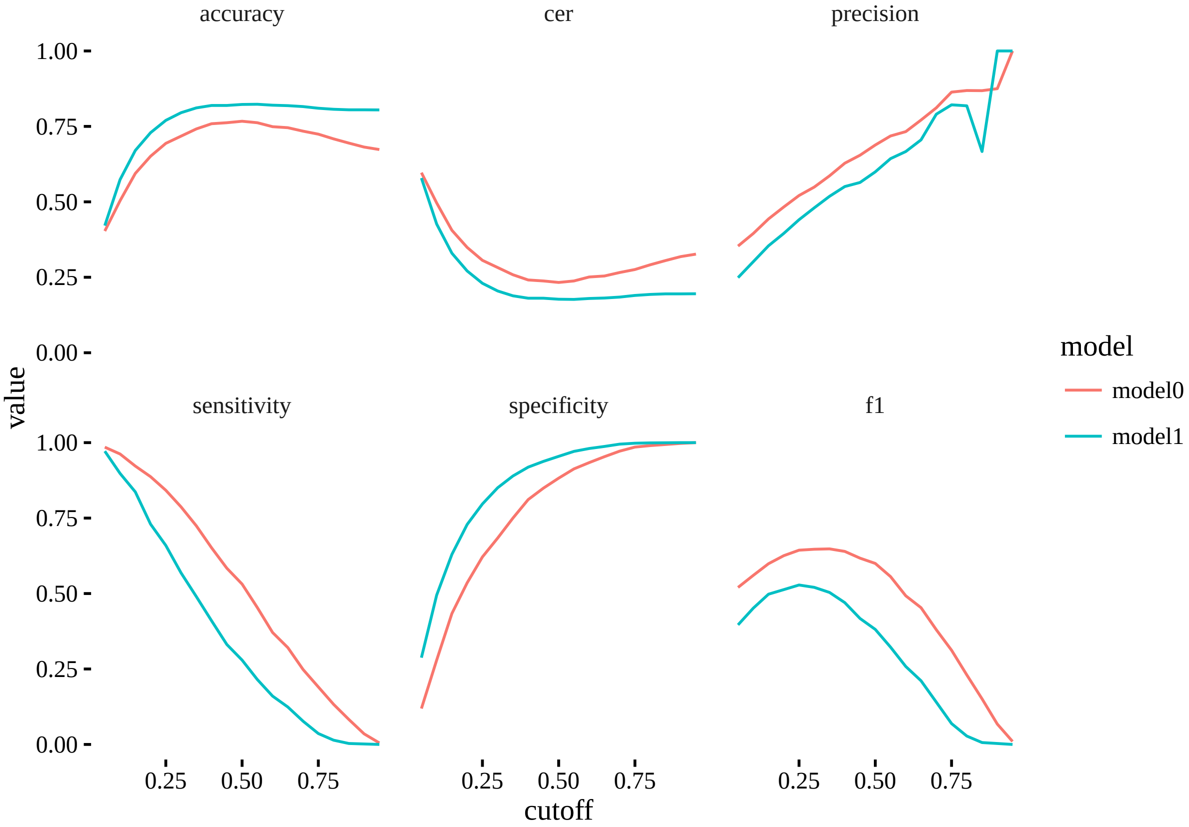
```
##
```

```
## Call:
## glm(formula = TARGET_FLAG ~ CLAIM_TYPE + REVOKED + JOB + URBANICITY +
##     CAR_TYPE + TRAVTIME + TIF + KIDSDRIV_BIN + MSTATUS + EDUCATION +
##     CAR_USE + INCOME + MVR_PTS + HOMEKIDS_BIN + SEX + YOJ + BLUEBOOK,
##     family = "binomial", data = insurancehv1)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0954  -0.6294  -0.3787  -0.1850   3.1237
##
## Coefficients:
##                                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)                     -2.772e+00  5.178e-01  -5.354 8.58e-08 ***
## CLAIM_TYPELow Previous Claim     1.077e+00  2.158e-01   4.992 5.99e-07 ***
## CLAIM_TYPENo Previous Claim      1.937e-01  2.092e-01   0.926 0.354394
## REVOKEDYes                       1.127e+00  1.596e-01   7.062 1.64e-12 ***
## JOBClerical                      4.340e-01  3.122e-01   1.390 0.164466
## JOBDoctor                       -3.362e-01  4.006e-01  -0.839 0.401310
## JOBHome Maker                    1.597e-01  3.597e-01   0.444 0.657013
## JOBLawyer                        2.144e-01  2.537e-01   0.845 0.398084
## JOBManager                      -2.640e-01  2.572e-01  -1.027 0.304522
## JOBProfessional                  3.944e-01  2.681e-01   1.471 0.141187
## JOBStudent                      -7.293e-01  7.228e-01  -1.009 0.312970
## JOBz_Blue Collar                 6.445e-01  2.883e-01   2.236 0.025374 *
## URBANICITYz_Highly Rural/ Rural -2.039e+00  2.215e-01  -9.203  < 2e-16 ***
## CAR_TYPEPanel Truck              5.697e-01  2.694e-01   2.114 0.034477 *
## CAR_TYPEPickup                   8.151e-01  1.797e-01   4.536 5.73e-06 ***
## CAR_TYPESports Car               1.341e+00  2.197e-01   6.105 1.03e-09 ***
## CAR_TYPEVan                      8.003e-01  2.133e-01   3.752 0.000175 ***
## CAR_TYPEz_SUV                    1.109e+00  1.866e-01   5.946 2.74e-09 ***
## TRAVTIME                         1.824e-02  3.205e-03   5.691 1.26e-08 ***
## TIF                             -6.542e-02  1.276e-02  -5.128 2.93e-07 ***
## KIDSDRIV_BIN                     5.065e-01  1.637e-01   3.093 0.001980 **
## MSTATUSz_No                      5.947e-01  1.191e-01   4.994 5.92e-07 ***
## EDUCATIONBachelors              -6.092e-01  2.032e-01  -2.998 0.002720 **
## EDUCATIONMasters                -3.477e-01  2.800e-01  -1.242 0.214338
## EDUCATIONPhD                     6.808e-02  3.387e-01   0.201 0.840693
## EDUCATIONz_High School          -3.307e-02  1.921e-01  -0.172 0.863328
## CAR_USEPrivate                  -5.984e-01  1.655e-01  -3.616 0.000299 ***
## INCOME                          -4.400e-06  1.610e-06  -2.733 0.006272 **
## MVR_PTS                          7.065e-02  2.536e-02   2.786 0.005340 **
## HOMEKIDS_BIN                     2.822e-01  1.244e-01   2.269 0.023295 *
## SEXz_F                          -2.066e-01  1.601e-01  -1.291 0.196858
## YOJ                              3.765e-02  1.885e-02   1.997 0.045846 *
## BLUEBOOK                        -1.276e-05  8.521e-06  -1.498 0.134203
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3220.6  on 3259  degrees of freedom
## Residual deviance: 2553.3  on 3227  degrees of freedom
## AIC: 2619.3
##
```

```
## Number of Fisher Scoring iterations: 5
```

Once again, my forward and backwards stepwise models are the same. Out of four potential logistic models for the first step in our modeling process, we ended up matching the forward and backwards step variable choices twice.

Lets examine the performance of these two models:



As a side note, it's interesting looking at these measures as the cutoff changes. This model shows some weird jumps in the precision measure. In previous tests of this model, I've seen dropoffs at the extreme cutoffs that can correspond to false positives with high predicted probabilities (or the converse with false negatives, although I haven't run into any of those.)

Below is a table of all of my measures at the 0.5 threshold:

measure	model0	model1
accuracy	0.7670782	0.8226994
cer	0.2329218	0.1773006
precision	0.6883117	0.5993266
sensitivity	0.5313283	0.2794349
specificity	0.8823529	0.9546321
f1	0.5997171	0.3811563

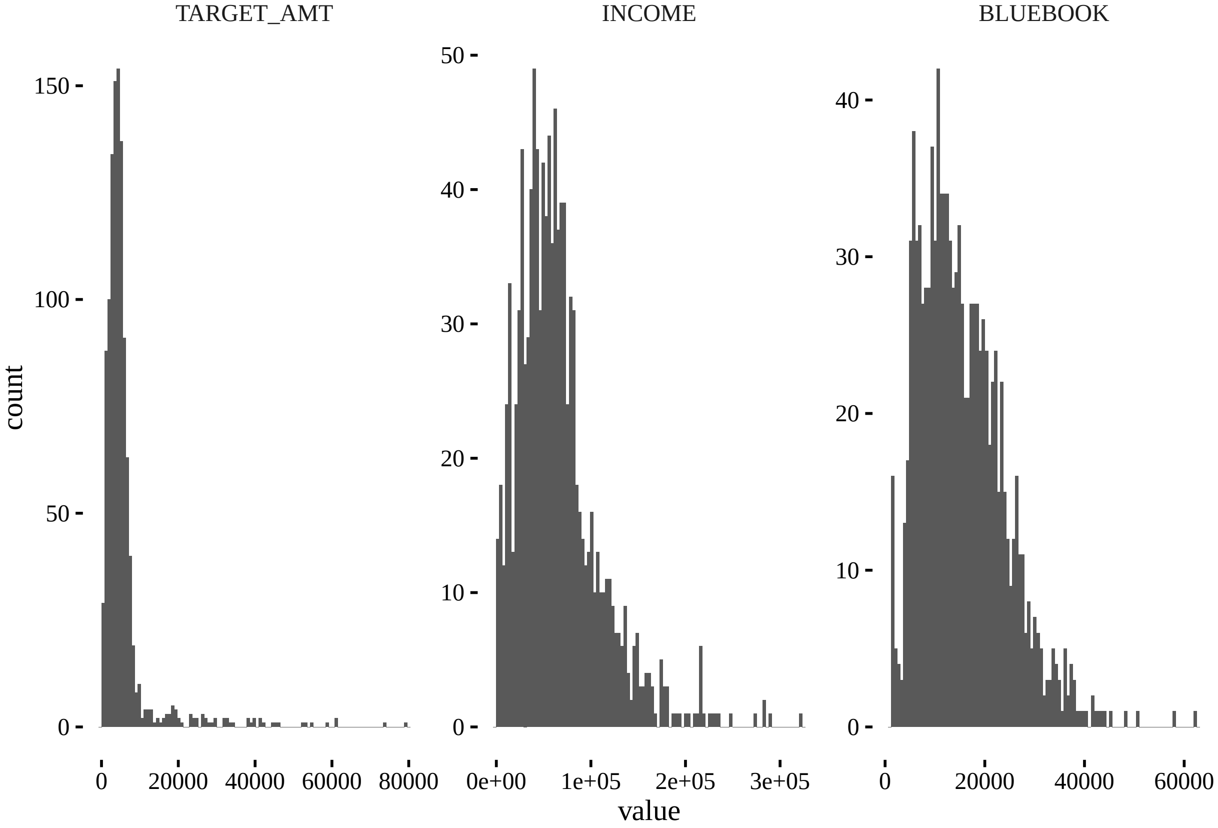
The results of our measures seeem mixed. Model1 has a higher accuracy and specificity, but lower precision and sensitivity. This seems to imply that Model1 performs really well in terms of false negatives, and that's driving the specificity and overall accuracy. Our forward and backward model selection processes both gave me the same model, and I need two models to deal with my split data. I'll talk more about what this could mean in terms of model selection in the next section.

Next, we'll build a few linear regression models to try to predict TARGET.AMT. I've established above that the HOME.VAL numeric variable doesn't have an effect on the TARGET.AMT, so I am able to consider only the binary variable derived from HOME.VAL indicating whether the observation belongs to a renter (if HOME.VAL is 0) or a homeowner (if HOME.VAL is greater than 0.) Because of this, I don't have to worry about splitting my data as I have above.

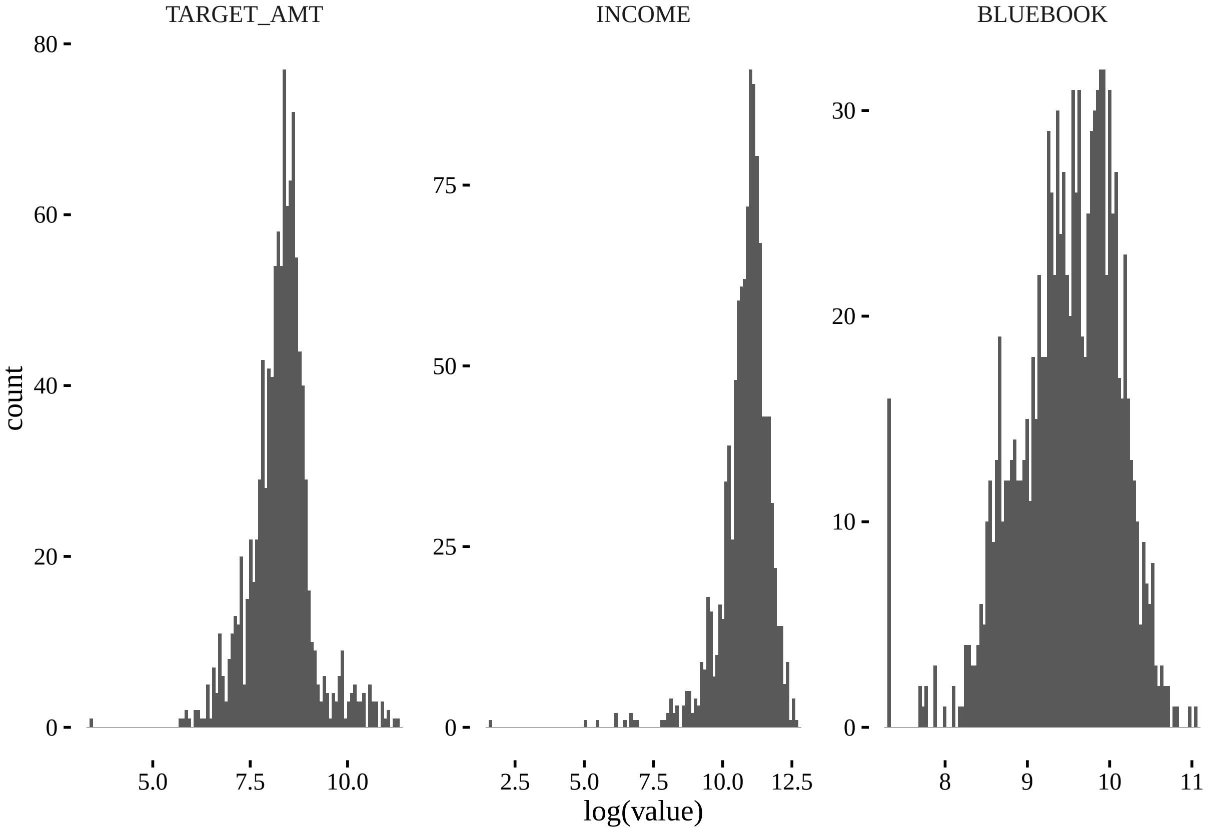
Data Exploration - Linear Model Focus

Because we are looking at TARGET.AMT, I do need to filter my evaluation data based on TARGET.FLAG.

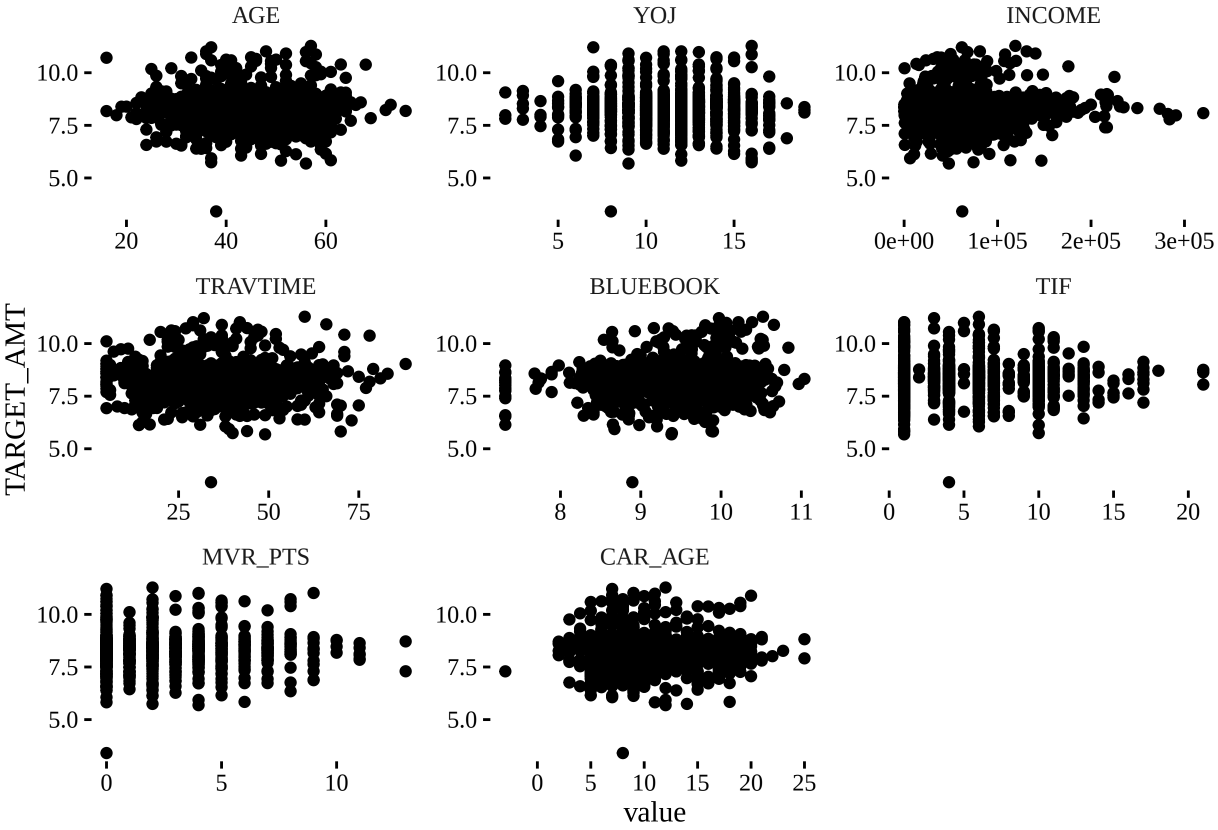
Revisiting my numeric analysis above, it would appear a few variables could use some transformations. Lets look at a few distributions that haven't been handled yet:



These three variables seem to be more normal when we apply a log transform to them (except for INCOME, which seems to get more skewed to the opposite direction):



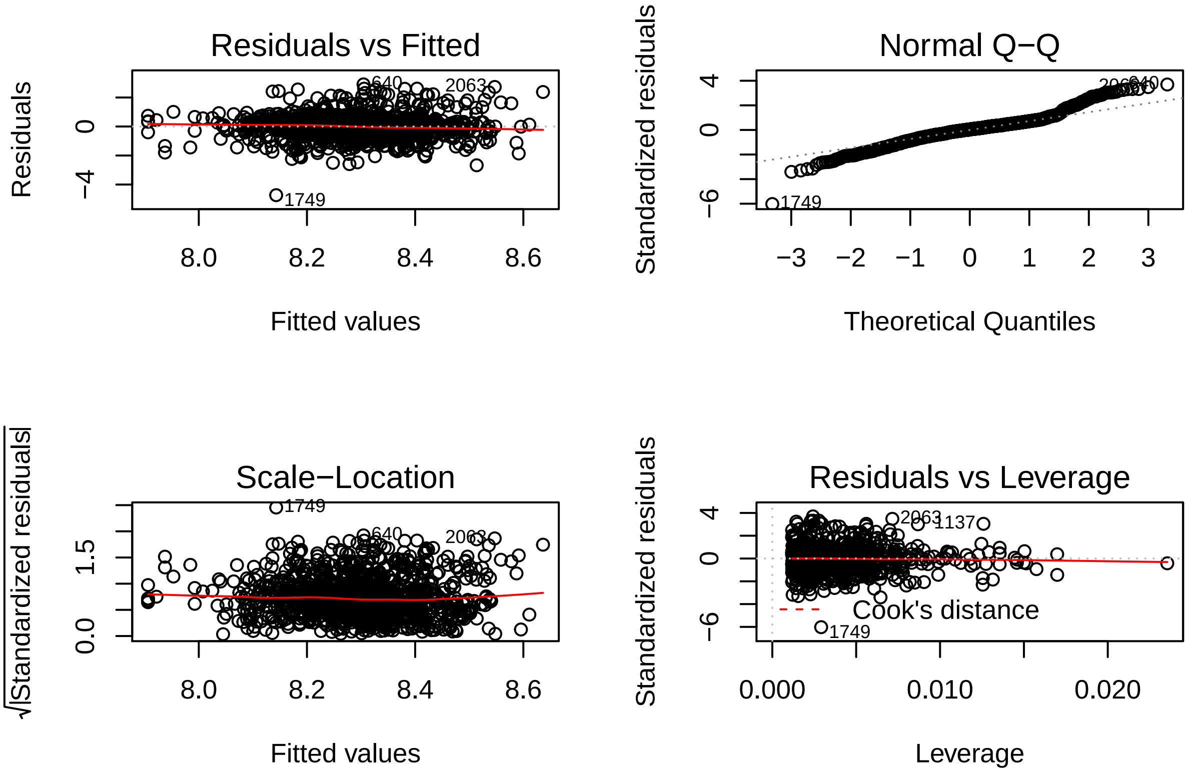
Lets look at a few scatter plots of our numeric data now, plotted versus TARGET.AMT:



With this transformation complete, I can move back to my build models section. But, I'm a bit concerned about the lack of relationships I'm seeing in these scatter plots.

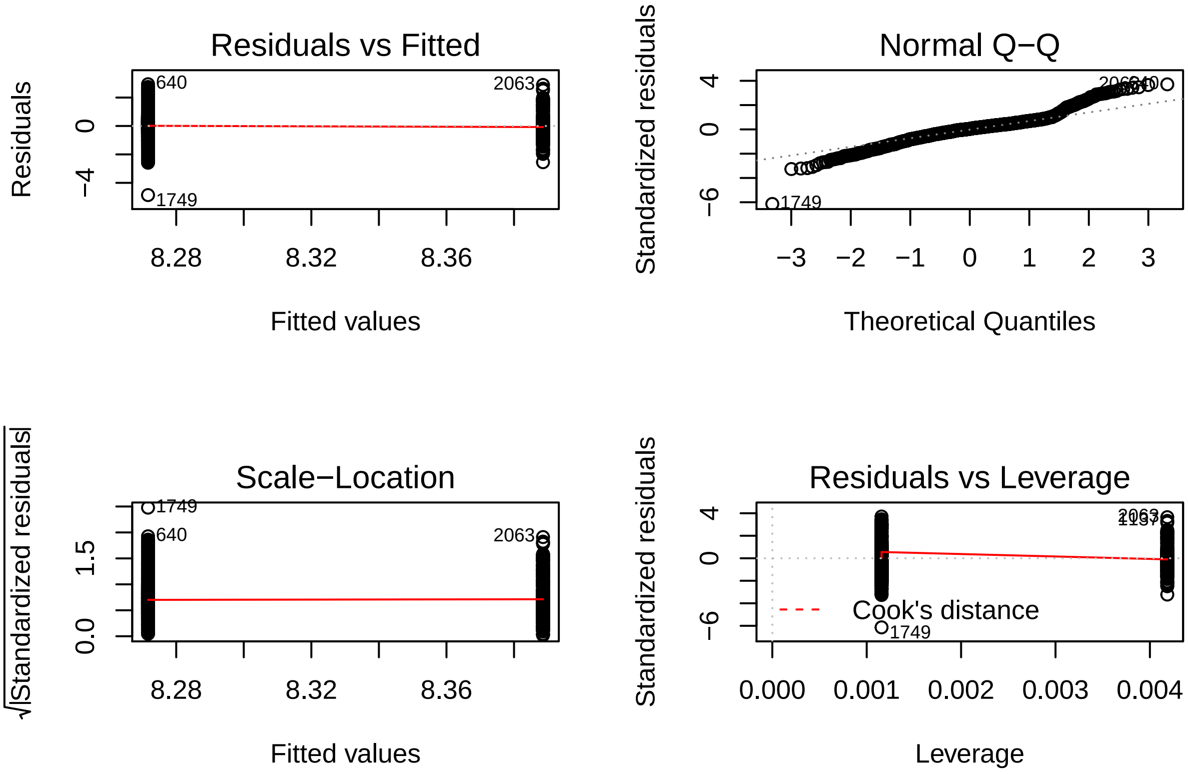
Build Models

For my first two models, I'll repeat the method used to build my logistic models, and step through my variables.



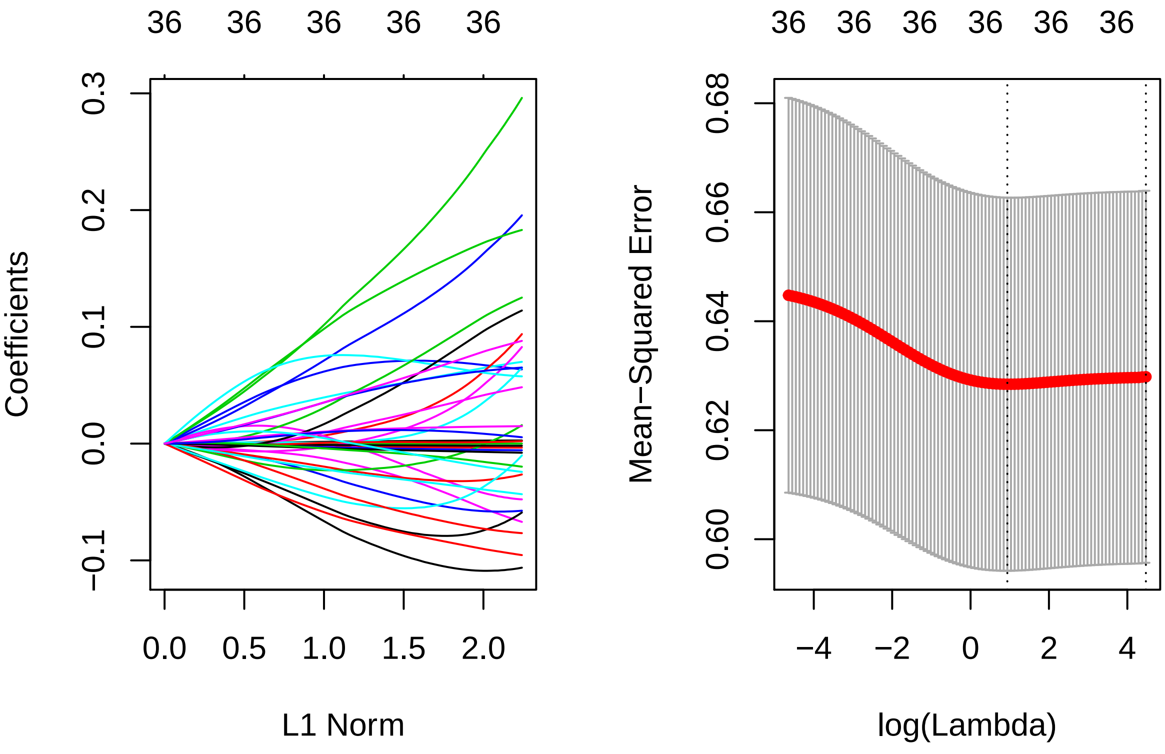
Once again, I arrived at the same model doing both forward and backward steps. My adjusted R-squared is very low (at 0.01673), and my plots show some problems in the Normal Q-Q plot. As I mentioned above, I questioned the predictive power of my numeric variables, so this poor model performance isn't too surprising.

For my second model selection process, I'll once again apply forward and backward steps, but this time I'll apply it to only my categorical variables. I'm hoping this might allow me to arrive at some models not reached by my forward and backward selection above.



It seems we weren't mistaken in our subset selection for Model 1. When manually removing numeric variables, my best stepwise model (once again the same for both forward and backward selection) has an even lower adjusted R-Squared, at 0.002785. There is only one variable, and it has a p-value of just below 0.05

Since my models have been poor performing so far, I'll try a ridge regression for my third linear model to see if there's any improvement.



We end up witha best lambda of 4.904021. The mean prediction error is 0.6229257 and the standard error is 0.1306891, with our coefficients below:

```
## 38 x 1 sparse Matrix of class "dgCMatrix"
##                                             1
## (Intercept)                      7.997383e+00
## (Intercept)                      .           
## AGE                              3.931880e-04
## YOJ                             -8.158494e-04
## INCOME                          -7.839744e-08
## PARENT1Yes                       2.472288e-02
## MSTATUSz_No                      1.591893e-02
## SEXz_F                          -4.157421e-03
## EDUCATIONBachelors              -1.739691e-02
## EDUCATIONMasters                 5.285068e-04
## EDUCATIONPhD                     2.966395e-02
## EDUCATIONz_High School          -6.292197e-03
## JOBClerical                      6.116083e-04
## JOBDoctor                        1.249057e-02
## JOBHome Maker                   -1.721684e-02
## JOBLawyer                       -6.225738e-03
## JOBManager                      -9.639573e-03
## JOBProfessional                  2.079656e-02
## JOBStudent                      -1.626101e-02
## JOBz_Blue Collar                -5.149871e-03
## TRAVTIME                        -3.481993e-04
## CAR_USEPrivate                  -2.051552e-03
## BLUEBOOK                         3.153094e-02
## TIF                             -9.258176e-04
## CAR_TYPEPanel Truck              3.898110e-02
## CAR_TYPEPickup                  -2.006338e-03
## CAR_TYPESports Car              -3.200308e-03
## CAR_TYPEVan                     -8.034754e-03
## CAR_TYPEz_SUV                    2.177342e-03
## RED_CARyes                       1.054004e-02
## REVOKEDYes                      -7.327211e-03
## MVR_PTS                          3.461998e-03
## CAR_AGE                         -9.175047e-04
## URBANICITYz_Highly Rural/ Rural -2.146699e-02
## CLAIM_TYPELow Previous Claim     4.505257e-04
## CLAIM_TYPENo Previous Claim      1.888931e-03
## KIDSDRIV_BIN                     8.861669e-03
## HOMEKIDS_BIN                     1.123960e-02
```

Model Selection

The method I've chosen, and the results of the models I've found, makes my model selection method less than straightforward.

My logistic models were done to subsets of the original data due to the nature of the HOME.VAL variable. Forward and backward stepwise variable selection was done, and in both cases we arrived at the same model. The models for predicting the TARGET.FLAG were decent. I would like to perform more research into why the measures seemed to differ (one preferring false positives and one preferring false negatives).

I wouldn't recommend using any models for predicting TARGET.AMT. From the beginning, it seemed like the numeric variables weren't providing any predictive value based on the scatter plots. The first model provided the best adjusted R squared, but it was extremely low. Performing the Ridge Regression allowed us to get some shrinkage, but didn't really improve the validity of our model. The data provided just does not predict the claim amount well.

Intuitively, one can see this as making sense. It's easier to predict whether or not someone gets into a crash than how much a claim will be filed for. The damage caused by a crash can be seen as more random.

R Code Appendix

Data Exploration - Logistic Focus

```r
cleanup <- function(x){
  return(as.numeric(gsub("[,$]", "", x)))
}

insurance$INCOME <- sapply(insurance$INCOME, cleanup)
insurance$HOME_VAL <- sapply(insurance$HOME_VAL, cleanup)
insurance$BLUEBOOK <- sapply(insurance$BLUEBOOK, cleanup)
insurance$OLDCLAIM <- sapply(insurance$OLDCLAIM, cleanup)

insurance$TARGET_FLAG <- factor(insurance$TARGET_FLAG)

insurancenum <- insurance %>%
  select(-c(PARENT1, MSTATUS, SEX, EDUCATION, JOB, CAR_USE,
            CAR_TYPE, RED_CAR, REVOKED, URBANICITY))

insurancebin <- insurance %>%
  select(INDEX, TARGET_FLAG, TARGET_AMT, PARENT1, MSTATUS, SEX, EDUCATION, JOB,
         CAR_USE, CAR_TYPE, RED_CAR, REVOKED, URBANICITY)

insurancemelt <- melt(insurancenum, id.vars=c('INDEX', 'TARGET_FLAG', 'TARGET_AMT'))

ggplot(insurancemelt, aes(x=TARGET_FLAG, y=value)) +
  geom_boxplot() + facet_wrap( ~ variable, scales = 'free') + theme_tufte()

ggplot(insurancemelt, aes(x=value)) + geom_histogram(bins = 100) +
  facet_wrap( ~ variable, scales = 'free') + theme_tufte()
```

```r
p1 <- grid.grabExpr(mosaic(~ TARGET_FLAG + factor(KIDSDRIV), data=insurance %>%
                              filter(KIDSDRIV != 0)))
p2 <- grid.grabExpr(mosaic(~ TARGET_FLAG + factor(KIDSDRIV), data=insurance))
grid.arrange(p2, p1, ncol=2)
```

```r
p1 <- grid.grabExpr(mosaic(~ TARGET_FLAG + factor(HOMEKIDS), data=insurance %>%
                              filter(HOMEKIDS != 0)))
p2 <- grid.grabExpr(mosaic(~ TARGET_FLAG + factor(HOMEKIDS), data=insurance))
grid.arrange(p2, p1, ncol=2)
```

```
p1 <- grid.grabExpr(mosaic(~ TARGET_FLAG + factor(CLM_FREQ), data=insurance %>%
                             filter(CLM_FREQ != 0)))
p2 <- grid.grabExpr(mosaic(~ TARGET_FLAG + factor(CLM_FREQ), data=insurance))
grid.arrange(p2, p1, ncol=2)
```

```
insurancenum2 <- select(insurancenum, INDEX, TARGET_FLAG, TARGET_AMT, INCOME,
                       HOME_VAL, OLDCLAIM, CAR_AGE)

insurancemelt2 <- melt(insurancenum2, id.vars=c('INDEX', 'TARGET_FLAG', 'TARGET_AMT'))

ggplot(insurancemelt2, aes(x=value)) + geom_histogram(bins = 100) +
  facet_wrap( ~ variable, scales = 'free') + theme_tufte()
```

```
insurance$OLDCLAIM_BIN <- ifelse(insurance$OLDCLAIM == 0, 0, 1)
insurance$CLM_FREQ_BIN <- ifelse(insurance$CLM_FREQ == 0, 0, 1)
mosaic( ~ OLDCLAIM_BIN + CLM_FREQ_BIN, data=insurance)

insurance$CLAIM_BIN <- insurance$CLM_FREQ_BIN

insurance <- select(insurance, -c(CLM_FREQ_BIN, OLDCLAIM_BIN))
```

```
ggplot(filter(insurance, OLDCLAIM != 0), aes(x=OLDCLAIM)) +
  geom_histogram(bins=100) + theme_tufte()
```

```
ggplot(filter(insurance, OLDCLAIM != 0), aes(x=log(OLDCLAIM))) +
  geom_histogram(bins=100) + theme_tufte()
```

```
p1 <- grid.grabExpr(mosaic(~ TARGET_FLAG + PARENT1, data=insurancebin))
p2 <- grid.grabExpr(mosaic(~ TARGET_FLAG + MSTATUS, data=insurancebin))
p3 <- grid.grabExpr(mosaic(~ TARGET_FLAG + SEX, data=insurancebin))
p4 <- grid.grabExpr(mosaic(~ TARGET_FLAG + EDUCATION, data=insurancebin,
                           rot_labels=c(90,90,90,90),
                           varnames = c(TRUE, FALSE)))
p5 <- grid.grabExpr(mosaic(~ TARGET_FLAG + JOB, data=insurancebin,
                           rot_labels=c(90,90,90,90),
                           varnames = c(TRUE, FALSE)))
p6 <- grid.grabExpr(mosaic(~ TARGET_FLAG + CAR_USE, data=insurancebin,
                           rot_labels = c(90,90,90,90),
                           varnames = c(TRUE, FALSE)))
p7 <- grid.grabExpr(mosaic(~ TARGET_FLAG + CAR_TYPE, data=insurancebin,
                           rot_labels=c(90,90,90,90),
                           varnames = c(TRUE, FALSE)))
p8 <- grid.grabExpr(mosaic(~ TARGET_FLAG + RED_CAR, data=insurancebin))
p9 <- grid.grabExpr(mosaic(~ TARGET_FLAG + REVOKED, data=insurancebin))
p10 <- grid.grabExpr(mosaic(~ TARGET_FLAG + URBANICITY, data=insurancebin,
                            rot_labels=c(90,90,90,90),
                            varnames = c(TRUE, FALSE)))

grid.arrange(p1, p2, p3, p4, ncol=2)

grid.arrange(p9, p6, p10, p8, ncol=2)

grid.arrange(p5, p7, ncol=2)
```

Data Transformation - Logistic Focus

```
d <- density(log(insurance$OLDCLAIM))
densitydata <- data.frame(x=d$x, y=d$y) %>%
  filter(x > 9 & x < 10) %>%
  arrange(y)

ggplot(filter(insurance, OLDCLAIM != 0), aes(x=log(OLDCLAIM))) +
  geom_density() + geom_vline(xintercept = densitydata$x[1], color = 'red',
                              linetype = 'longdash') + theme_tufte()
```

```
insurance$CLAIM_TYPE <- factor(ifelse(insurance$OLDCLAIM == 0, 'No Previous Claim',
                               ifelse(log(insurance$OLDCLAIM) <= 9.62,
                                      'Low Previous Claim', 'High Previous Claim')))

insurance <- insurance %>%
  select(-c(OLDCLAIM, CLAIM_BIN, CLM_FREQ))

mosaic(~ CLAIM_TYPE + TARGET_FLAG, data=insurance, rot_labels=c(0,0,0,0))
```

```
ggplot(filter(insurance, HOME_VAL != 0 & TARGET_AMT != 0),
       aes(x=HOME_VAL, y=TARGET_AMT)) + geom_point() + theme_tufte()
```

```
ggplot(filter(insurance, HOME_VAL != 0 & TARGET_AMT != 0),
       aes(x=log(HOME_VAL), y=log(TARGET_AMT))) + geom_point() + theme_tufte() +
  stat_smooth(method = 'lm')
```

```
model <- lm(log(TARGET_AMT) ~ log(HOME_VAL),
            data = filter(insurance, HOME_VAL != 0 & TARGET_AMT != 0))

summary(model)
```

```
ggplot(filter(insurance, HOME_VAL != 0),
       aes(y=log(HOME_VAL), x=TARGET_FLAG)) + geom_boxplot() + theme_tufte()
```

```
summary(aov(log(HOME_VAL) ~ TARGET_FLAG, data = filter(insurance, HOME_VAL != 0)))


############################################################
#
# Two sample p-test sanity check
#
#a <- na.omit(filter(insurance, HOME_VAL !=0 & TARGET_FLAG == 0))$HOME_VAL
#b <- na.omit(filter(insurance, HOME_VAL !=0 & TARGET_FLAG == 1))$HOME_VAL

#SE <- sqrt((sd(a)^2)/length(a) + (sd(b)^2)/length(b))

#(mean(a) - mean(b))-2*SE
############################################################
```

```
insurance$INCOME[insurance$INCOME == 0] <- NA
insurance$CAR_AGE[insurance$CAR_AGE == 1] <- NA

insurance$KIDSDRIV_BIN <- ifelse(insurance$KIDSDRIV == 0, 0, 1)
insurance$HOMEKIDS_BIN <- ifelse(insurance$HOMEKIDS == 0, 0, 1)

insurance <- select(insurance, -c(KIDSDRIV, HOMEKIDS))
```

Build Models - Logistic Focus

```
insurancehv0 <- na.omit(filter(insurance, HOME_VAL == 0 & !is.na(HOME_VAL)) %>%
  select(-c(TARGET_AMT, INDEX, HOME_VAL)))
insurancehv0 <- cbind(select(insurancehv0, -TARGET_FLAG),
                      select(insurancehv0, TARGET_FLAG))

insurancehv1 <- na.omit(filter(insurance, HOME_VAL != 0 & !is.na(HOME_VAL)) %>%
  select(-c(TARGET_AMT, INDEX)))
insurancehv1 <- cbind(select(insurancehv1, -TARGET_FLAG),
                      select(insurancehv1, TARGET_FLAG))

#bestmodel <- bestglm(na.omit(insurancehv0), IC= "BIC", family = binomial)

fullmod0 <- glm(TARGET_FLAG ~ ., family = 'binomial', data=na.omit(insurancehv0))
backmodel0 <- step(fullmod0, trace=0)

nothing0 <- glm(TARGET_FLAG ~ 1, family = 'binomial', data=insurancehv0)
forwardmodel0 <- step(nothing0,
                      scope=list(lower=formula(nothing0),upper=formula(fullmod0)),
                      direction = 'forward', trace=0)

model0 <- forwardmodel0

insurancehv0$model <- predict(model0,insurancehv0,type='response')

summary(model0)
```

```
fullmod1 <- glm(TARGET_FLAG ~ ., family = 'binomial', data=na.omit(insurancehv1),
                trace=0)
backmodel1 <- step(fullmod1, trace=0)

nothing1 <- glm(TARGET_FLAG ~ 1, family = 'binomial', data=insurancehv1)
forwardmodel1 <- step(nothing1,
                      scope=list(lower=formula(nothing1),upper=formula(fullmod1)),
                      direction = 'forward', trace=0)

model1 <- forwardmodel1

insurancehv1$model <- predict(model1,insurancehv1,type='response')

#roc.results1 <- roc(TARGET_FLAG ~ backmodel, data=insurancehv0)
```

```r
#plot(roc.results1)

#auc(roc.results1)

summary(model1)
```

```r
accuracy.calc <- function(data, actual, proportion, cutoff){
  data$predicted <- ifelse(data[,proportion]>cutoff, 1, 0)
  confusion <- table(dplyr::select(data, get(actual), predicted))
  if(dim(confusion)[2] == 1){
    if(colnames(confusion) == 0){
      accuracy <- confusion[1,1]/sum(confusion)
    } else{
      accuracy <- confusion[2,2]/sum(confusion)
    }
  } else{
    accuracy <- (confusion[1,1] + confusion[2,2])/sum(confusion)
  }
  return(accuracy)
}

cer.calc <- function(data, actual, proportion, cutoff){
  data$predicted <- ifelse(data[,proportion]>cutoff, 1, 0)
  confusion <- table(dplyr::select(data, get(actual), predicted))
  if(dim(confusion)[2] == 1){
    if(colnames(confusion) == 0){
      cer <- confusion[2,1]/sum(confusion)
    } else{
      cer <- confusion[1,1]/sum(confusion)
    }
  } else{
    cer <- (confusion[1,2] + confusion[2,1])/sum(confusion)
  }
  return(cer)
}

precision.calc <- function(data, actual, proportion, cutoff){
  data$predicted <- ifelse(data[,proportion]>cutoff, 1, 0)
  confusion <- table(dplyr::select(data, get(actual), predicted))
  if(dim(confusion)[2] == 1){
    if(colnames(confusion) == 0){
      precision <- 1
    } else{
      precision <- confusion[2,1]/(confusion[1,1]+confusion[2,1])
    }
  } else{
    precision <- confusion[2,2]/(confusion[1,2] + confusion[2,2])
  }
  return(precision)
}

sensitivity.calc <- function(data, actual, proportion, cutoff){
  data$predicted <- ifelse(data[,proportion]>cutoff, 1, 0)
  confusion <- table(dplyr::select(data, get(actual), predicted))
  if(dim(confusion)[2] == 1){
    if(colnames(confusion) == 0){
      sensitivity <- 0
    } else{
      sensitivity <- 1
    }
  } else{
      sensitivity <- confusion[2,2]/(confusion[2,1] + confusion[2,2])
  }

  return(sensitivity)
}

specificity.calc <- function(data, actual, proportion, cutoff){
  data$predicted <- ifelse(data[,proportion]>cutoff, 1, 0)
  confusion <- table(dplyr::select(data, get(actual), predicted))
  if(dim(confusion)[2] == 1){
    if(colnames(confusion) == 0){
      specificity <- 1
    } else{
      specificity <- 0
    }
  } else{
    specificity <- confusion[1,1]/(confusion[1,1] + confusion[1,2])
  }
  return(specificity)
}

f1.calc <- function(data, actual, proportion, cutoff){
  precision <- precision.calc(data, actual, proportion, cutoff)
  sensitivity <- sensitivity.calc(data, actual, proportion, cutoff)
  f1 <- (2 * precision * sensitivity)/(precision + sensitivity)
  return(f1)
}
```

```
cutofflist <- seq(0.05,0.95,by=0.05)

measures0 <- data.frame(cutoff = cutofflist, model = 'model0',
                       accuracy = apply(matrix(cutofflist), 1, accuracy.calc,
                                        data = insurancehv0, actual = "TARGET_FLAG",
                                        proportion = "model"),
                       cer = apply(matrix(cutofflist), 1, cer.calc,
                                   data = insurancehv0, actual = "TARGET_FLAG",
                                   proportion = "model"),
                       precision = apply(matrix(cutofflist), 1, precision.calc,
                                         data = insurancehv0, actual = "TARGET_FLAG",
                                         proportion = "model"),
                       sensitivity = apply(matrix(cutofflist), 1, sensitivity.calc,
                                           data = insurancehv0, actual = "TARGET_FLAG",
                                           proportion = "model"),
                       specificity = apply(matrix(cutofflist), 1, specificity.calc,
                                           data = insurancehv0, actual = "TARGET_FLAG",
                                           proportion = "model"),
                       f1 = apply(matrix(cutofflist), 1, f1.calc,
                                  data = insurancehv0, actual = "TARGET_FLAG",
                                  proportion = "model"))

measures1 <- data.frame(cutoff = cutofflist, model = 'model1',
                       accuracy = apply(matrix(cutofflist), 1, accuracy.calc,
                                        data = insurancehv1, actual = "TARGET_FLAG",
                                        proportion = "model"),
                       cer = apply(matrix(cutofflist), 1, cer.calc,
                                   data = insurancehv1, actual = "TARGET_FLAG",
                                   proportion = "model"),
                       precision = apply(matrix(cutofflist), 1, precision.calc,
                                         data = insurancehv1, actual = "TARGET_FLAG",
                                         proportion = "model"),
                       sensitivity = apply(matrix(cutofflist), 1, sensitivity.calc,
                                           data = insurancehv1, actual = "TARGET_FLAG",
                                           proportion = "model"),
                       specificity = apply(matrix(cutofflist), 1, specificity.calc,
                                           data = insurancehv1, actual = "TARGET_FLAG",
                                           proportion = "model"),
                       f1 = apply(matrix(cutofflist), 1, f1.calc,
                                  data = insurancehv1, actual = "TARGET_FLAG",
                                  proportion = "model"))

measures <- rbind(measures0, measures1)

measures <- melt(measures, id.vars = c('cutoff', 'model'), variable.name = 'measure',
                 value.name = 'value')

ggplot(measures, aes(x=cutoff, y=value, color=model)) +
  geom_line() + facet_wrap(~ measure) + theme_tufte()
```

```
kable(dcast(filter(measures, cutoff == 0.5), measure ~ model))
```

Data Exploration - Linear Model Focus

```
#insurancetf1 <- na.omit(filter(insurance, TARGET_FLAG == 1) %>%
#                          select(-c(INDEX, TARGET_FLAG, HOME_VAL)))

insurancetf1 <- na.omit(filter(insurance, TARGET_FLAG == 1) %>%
                          select(-c(TARGET_FLAG, HOME_VAL)))
```

```
insurancenum <- insurancetf1 %>%
  select(INDEX, TARGET_AMT, INCOME, BLUEBOOK)

insurancemelt <- melt(insurancenum, id.vars='INDEX')

ggplot(insurancemelt, aes(x=value)) + geom_histogram(bins = 100) +
  facet_wrap( ~ variable, scales = 'free') + theme_tufte()
```

```r
ggplot(insurancemelt, aes(x=log(value))) + geom_histogram(bins = 100) +
  facet_wrap( ~ variable, scales = 'free') + theme_tufte()

insurancetf1$TARGET_AMT <- log(insurancetf1$TARGET_AMT)
#insurancetf1$INCOME <- log(insurancetf1$INCOME)
insurancetf1$BLUEBOOK <- log(insurancetf1$BLUEBOOK)

#insurancetf1$TARGET_AMT <- exp(insurancetf1$TARGET_AMT)
#insurancetf1$INCOME <- exp(insurancetf1$INCOME)
#insurancetf1$BLUEBOOK <- exp(insurancetf1$BLUEBOOK)
```

```r
insurancenum <- insurancetf1 %>% select(TARGET_AMT, AGE, YOJ, INCOME, TRAVTIME,
                                        BLUEBOOK, TIF, MVR_PTS, CAR_AGE, INDEX)

insurancemelt <- melt(insurancenum, id.vars=c('INDEX', 'TARGET_AMT'))

ggplot(insurancemelt, aes(x=value, y=TARGET_AMT)) + geom_point() +
  facet_wrap( ~ variable, scales = 'free') + theme_tufte()
```

Build Models

```r
insurancetf1model1 <- insurancetf1 %>% select(-INDEX)

fullmodta <- lm(TARGET_AMT ~ ., data=na.omit(insurancetf1model1))
backmodelta <- step(fullmodta, trace=0)

nothingta <- lm(TARGET_AMT ~ 1, data=na.omit(insurancetf1model1))
forwardmodelta <- step(nothingta,
                      scope=list(lower=formula(nothingta),upper=formula(fullmodta)),
                      direction = 'forward', trace=0)

modelta1 <- forwardmodelta

par(mfrow=c(2,2))
plot(modelta1)
par(mfrow=c(1,1))
```

```r
insurancetf1model2 <- select(insurancetf1, TARGET_AMT, PARENT1, MSTATUS, SEX,
                             EDUCATION, JOB, CAR_USE, CAR_TYPE, RED_CAR, REVOKED,
                             URBANICITY, CLAIM_TYPE, KIDSDRIV_BIN, HOMEKIDS_BIN)

fullmodta2 <- lm(TARGET_AMT ~ ., data=na.omit(insurancetf1model2))
backmodelta2 <- step(fullmodta2, trace=0)

nothingta2 <- lm(TARGET_AMT ~ 1, data=na.omit(insurancetf1model2))
forwardmodelta2 <- step(nothingta2,
                      scope=list(lower=formula(nothingta2),upper=formula(fullmodta2)),
                      direction = 'forward', trace=0)

modelta2 <- forwardmodelta2
```

```r
par(mfrow=c(2,2))
plot(modelta2)
par(mfrow=c(1,1))
```

```r
insurancetf1model3 <- select(insurancetf1, -INDEX)

x <- model.matrix(TARGET_AMT ~ ., data=insurancetf1model3)

modelta3 <- glmnet(x, as.matrix(select(insurancetf1model3, TARGET_AMT)), alpha=0)

cv.modelta3 <- cv.glmnet(x,as.matrix(select(insurancetf1model3, TARGET_AMT)), alpha=0)

par(mfrow = c(1,2))
plot(modelta3)
plot(cv.modelta3)
par(mfrow=c(1,1))

bestlam <- cv.modelta3$lambda.min

modelta3.bestlam <- glmnet(x, as.matrix(select(insurancetf1model3, TARGET_AMT)),
                           alpha = 0, lambda = bestlam)

insurancetf1model3$modelta3.pred <- predict(modelta3, s = bestlam, newx = x)

mpe <- mean((insurancetf1model3$modelta3.pred - insurancetf1model3$TARGET_AMT)^2)
se <- sd((insurancetf1model3$modelta3.pred -
            insurancetf1model3$TARGET_AMT)^2)/
  sqrt(length(insurancetf1model3$TARGET_AMT))
```

```
## 38 x 1 sparse Matrix of class "dgCMatrix"
##                                     1
## (Intercept)              7.997383e+00
## (Intercept)              .
## AGE                      3.931880e-04
## YOJ                     -8.158494e-04
## INCOME                  -7.839744e-08
## PARENT1Yes               2.472288e-02
## MSTATUSz_No              1.591893e-02
## SEXz_F                  -4.157421e-03
## EDUCATIONBachelors      -1.739691e-02
## EDUCATIONMasters         5.285068e-04
## EDUCATIONPhD             2.966395e-02
## EDUCATIONz_High School  -6.292197e-03
## JOBClerical              6.116083e-04
## JOBDoctor                1.249057e-02
## JOBHome Maker           -1.721684e-02
## JOBLawyer               -6.225738e-03
## JOBManager              -9.639573e-03
## JOBProfessional          2.079656e-02
## JOBStudent              -1.626101e-02
## JOBz_Blue Collar        -5.149871e-03
## TRAVTIME                -3.481993e-04
## CAR_USEPrivate          -2.051552e-03
## BLUEBOOK                          3.153094e-02
## TIF                               -9.258176e-04
## CAR_TYPEPanel Truck                3.898110e-02
## CAR_TYPEPickup                    -2.006338e-03
## CAR_TYPESports Car                -3.200308e-03
## CAR_TYPEVan                       -8.034754e-03
## CAR_TYPEz_SUV                      2.177342e-03
## RED_CARyes                         1.054004e-02
## REVOKEDYes                        -7.327211e-03
## MVR_PTS                            3.461998e-03
## CAR_AGE                           -9.175047e-04
## URBANICITYz_Highly Rural/ Rural -2.146699e-02
## CLAIM_TYPELow Previous Claim       4.505257e-04
## CLAIM_TYPENo Previous Claim        1.888931e-03
## KIDSDRIV_BIN                       8.861669e-03
## HOMEKIDS_BIN                       1.123960e-02
```