Identifying Gaps in Assisted Housing

Michael Rath, Nicholas Maier, and Lara Haase

Final Project Team #13

Michael Rath - mjrath
Nicholas Maier - nmaier
Lara Haase - lhaase

Project Objective

Data set:

Housing Affordability Data System, published by the U.S. Department of Housing and Urban Development

"A set of housing-unit level datasets that measures the affordability of housing units and the housing cost burdens of households, relative to area median incomes, poverty level incomes, and Fair Market Rents." - from the data documentation (available at https://www.huduser.gov/portal/datasets/hads/HADS_doc.pdf)

Objective:

We used the variable "FMTASSISTED", which indicated whether a household receives housing assistance or not, as our dependent or predicted variable. The intention of our analysis is to use the data available to create predictive models able to determine whether a household is likely to receive housing assistance. The value proposition we intend to develop, either for public or private organizations, is the ability to detect households that may qualify for housing assistance, but may not be accessing the available services.

In the end we hope to identify two groups of people. The first and most important is those who are currently not recieving housing assistance that could use it. This could allow HUD or other organizations to proactively target these individuals to give them the help that they need. The second group would be individuals who are receiving assistance that potentially may not need it. If these individuals are taking advantage of the system, that could be a non-effective use of resources. Identifying these individuals could be useful for places like HUD to examine their cases more closely to be proper stewards of housing assistance resources.

Exploratory Data Analysis

Loading and Cleaning Data

The data exploration, understanding, and processing phase of the project was significant, as the dataset consists of 64535 observations of 99 predictors. The first part of our process was to establish which observations and vaariables would be kept for the next phase of data analysis.

The first stage resulted in the removal of 41075 observations of data that did not have our target variable of interest, FMTASSISTED. Because these observations had no value for FMTASSISTED, they would be unusable for our analysis.

The next part was a lengthy search through the 98 variables that were not FMTASSISTED to determine which should be included. Ultimately we ended up with 34 variables left. The 60 variables in the inital screen were removed for falling into one of three groups.

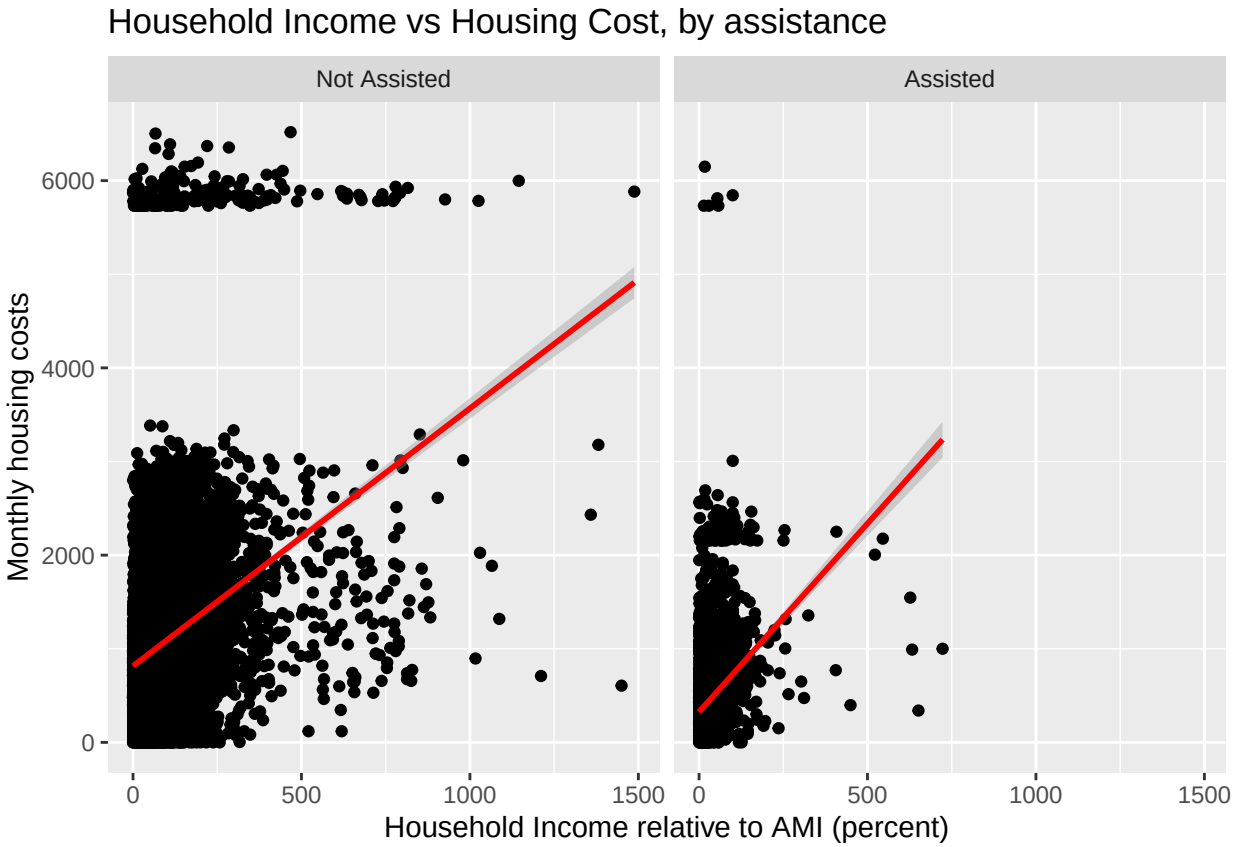
- Simply Transforming Other Data: This category includes over 20 predictors such as COST06, COST 08, COST12, COST06RELAMIPCT, COST08RELAMIPCT and more. These variables are only transformations of existing data using different rates. For example, COSTO6RELAMIPCT is just the COSMEDRELAMIPCT at a 6% interest rate, while COST08AMIPCT is that same variable but at an 8% interest rate. While these may have been valuable to some studies, they had no bearing here (they would have had perfect collinearity in any case) and if necessary, could have been recreated by our team.
- Noninterpretable Predictors: This category includes variables such as WEIGHT, TENURE, TYPE, and others. These variables have values whose meanings cannot be understood. Although there is a data dictionary, it is not detailed enough for every variable. For example, TYPE has numeric values 1-9, but nowhere is it described what each number represents. This is unfortunate, but without an alternative, we made the decision to remove them as we couldn't be sure what they meant or how the data should be treated. As an example, should TYPE be a categorical discreete variable or a continuous one? Without knowing that, we are unable to evaluate it properly.
- Redundancy of Information: 25 variables fall into this category. 25 variables in the dataset were FMT versions of another predictor, which means they had been formatted and treated in some way. As an example, BUILT and FMTBUILT shared information but displayed it in similar ways. BUILT gives the year a housing unit is built (eg. 1983), whereas FMTBUILT groups things into decade in which it was produced and the value would be recored as "1980-1989." For something like BUILT, being able to treat it as a continous variable was more valuable, so we elected to discard FMTBUILT. There were other scenarios where the FMT version was the only one with an interpretable form of the infromation. For example, FMTSTRUCTRETYPE labeled the kind of structure (eg. "Single Family"), whereas STRUCTRETYPE did not.
- Same Values: This category includes variables such as VALUE, OWNRENT, VACANCY, STATUS, and more, which had only one value for the entirity of our data set. For example, VALUE is set to -6 when there is a non-null value for if the observation is specifically classified as assisted or not assisted. In our selection, all data is one of those two groups, so all have a -6 for VALUE.

Exploratory Data Analysis

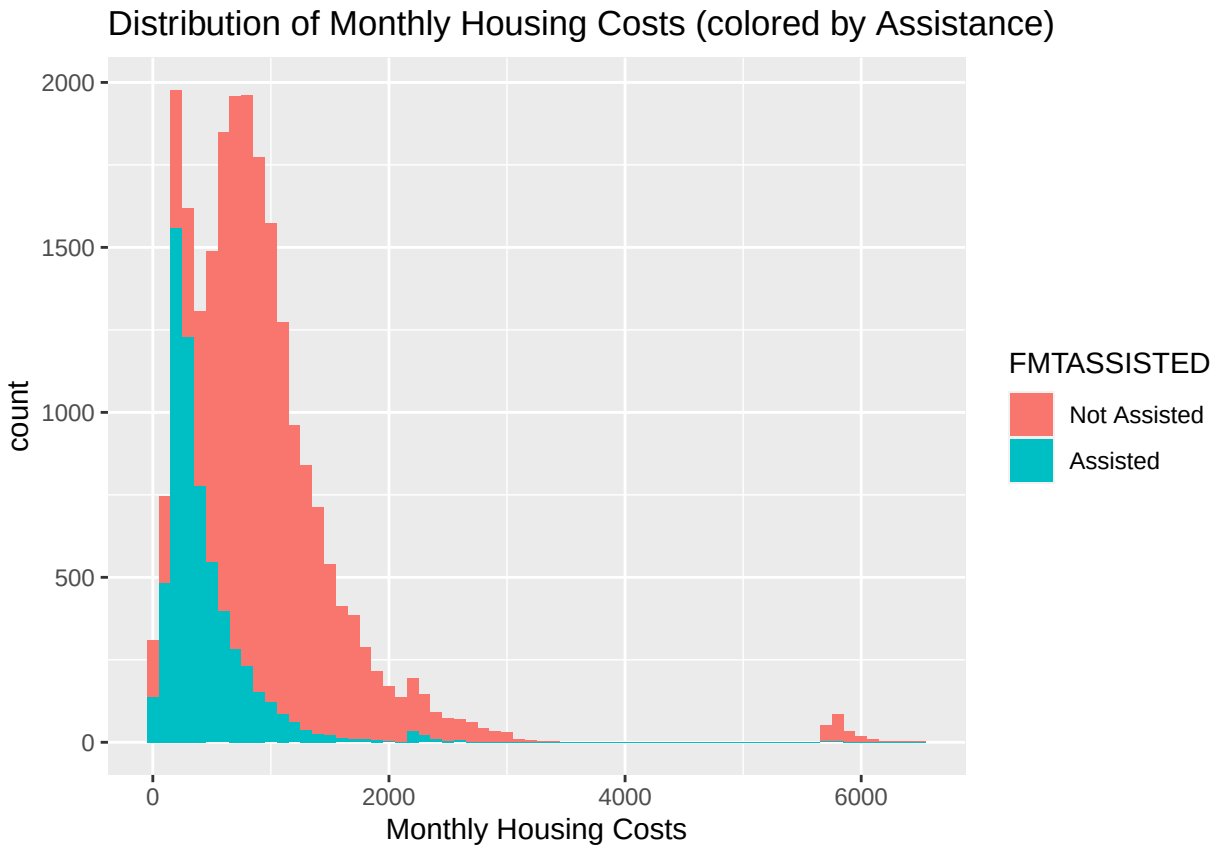
Exploration with Plots

This first plot exemplifies our problem definition, to attempt to identify characteristics of households that are likely in need of assistance, but are not recieving it. The plot compares income (relative to Area Median Income) to housing costs (at median interest relative to area median income). The plots are broken into households that receive assistance vs those that do not. But unfortunately the plots are not particularly different and have quite a bit of overlap.

```
## `geom_smooth()` using formula 'y ~ x'
```



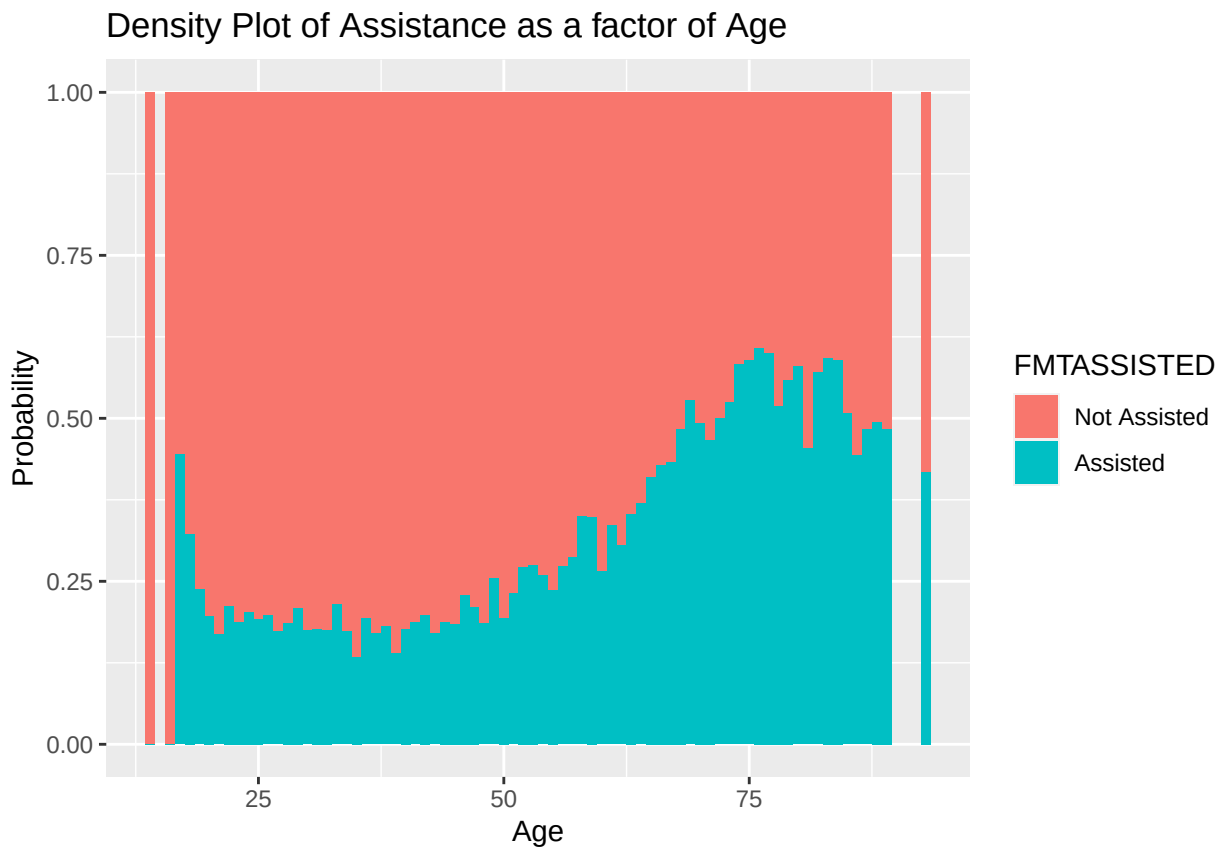
In that plot we can see that there are a few households that are behaving as outliers. To highlight this, below we have plotted a histogram of housing costs (stacked with the assisted variable for the color).



We decided to remove these outliers for the rest of our analyses.

How `AGE` affects whether a household is assisted:

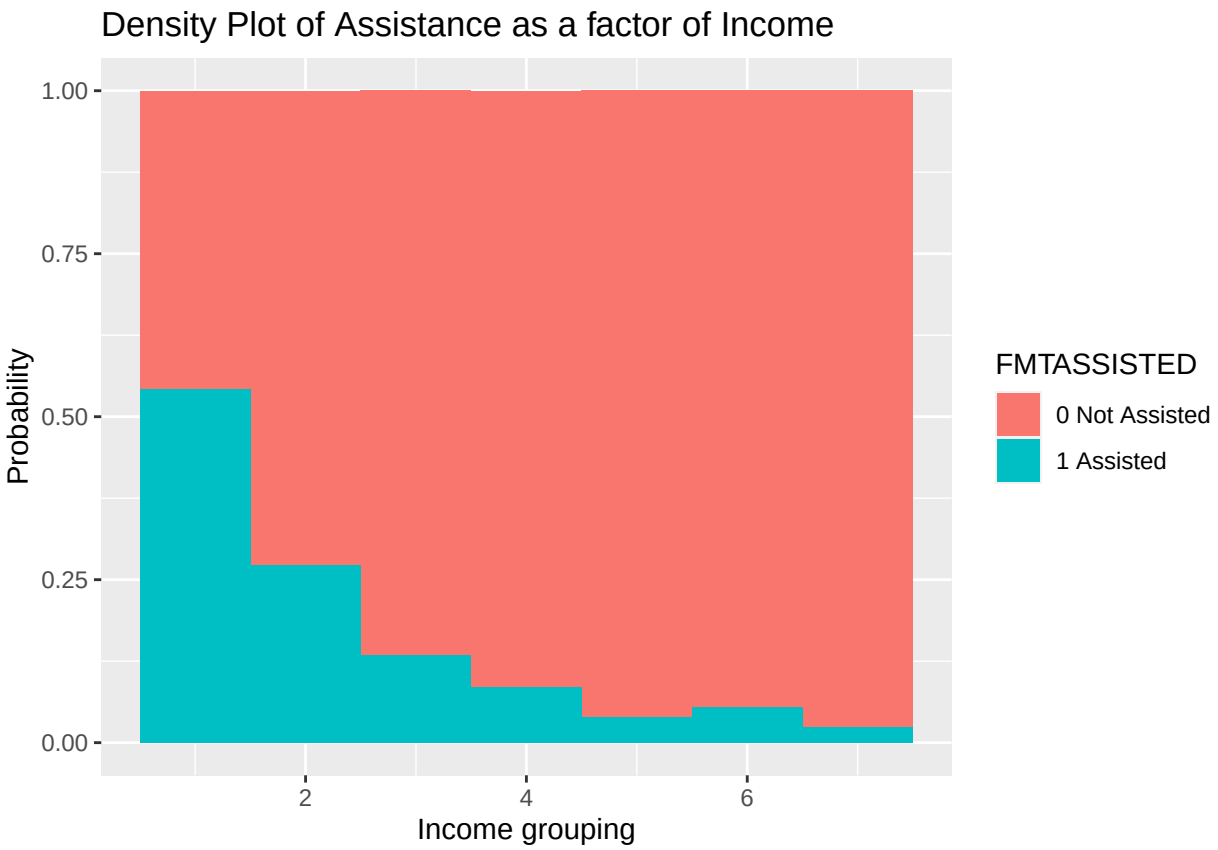
```
## Warning: Removed 8 rows containing missing values (geom_bar).
```



This plot shows that people are more likely to receive housing assistance when there are very young adults (about 16-17 years old) and after the age of about 65 year old. ### Size of Household

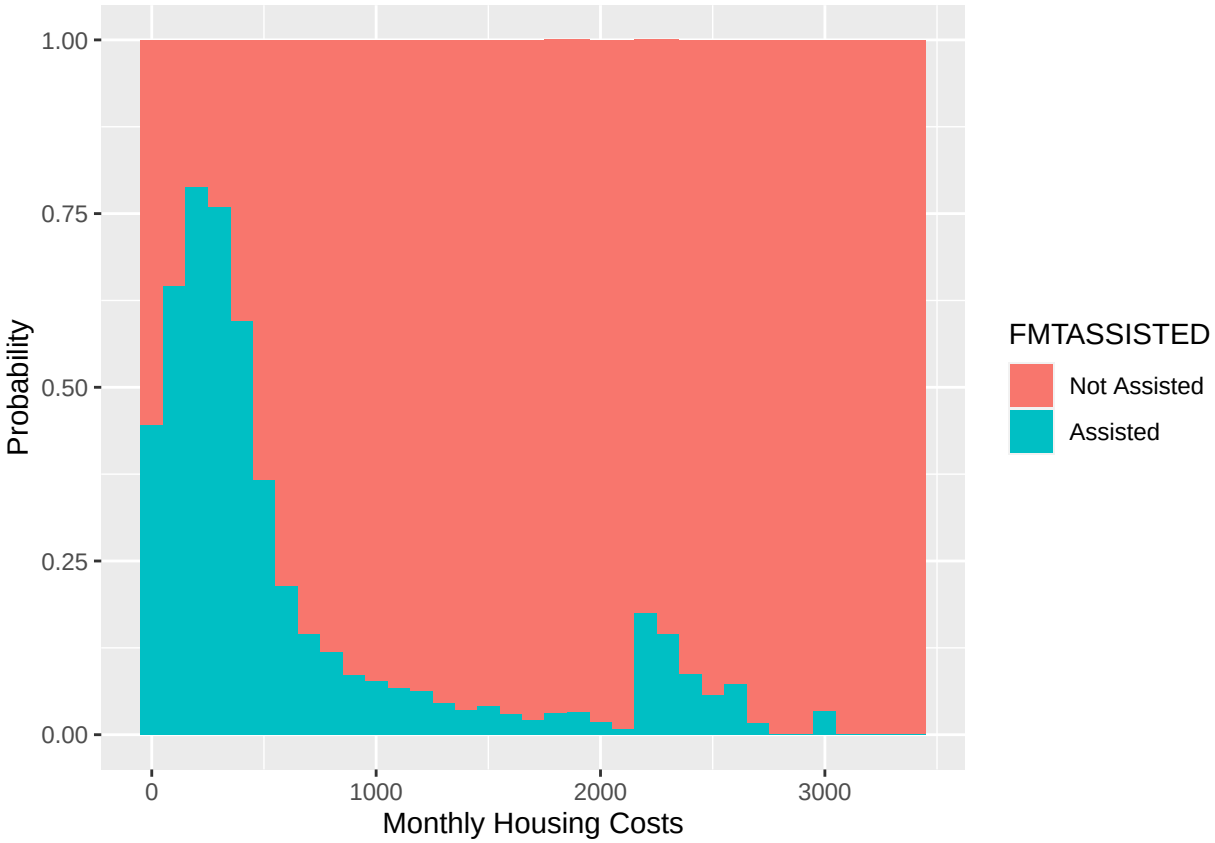
This plot shows the distribution of the number of people in the households, broken up by assistance status. There are interesting insights that contradict our assumptions about assisted households, in that they tend to be smaller (just 1 person) than unassisted households (2 people). This may be a result of two or more people living together having an easier time pooling resources to share housing costs. Other insights are unclear without deeper analysis.

Income Effects of Assistance status



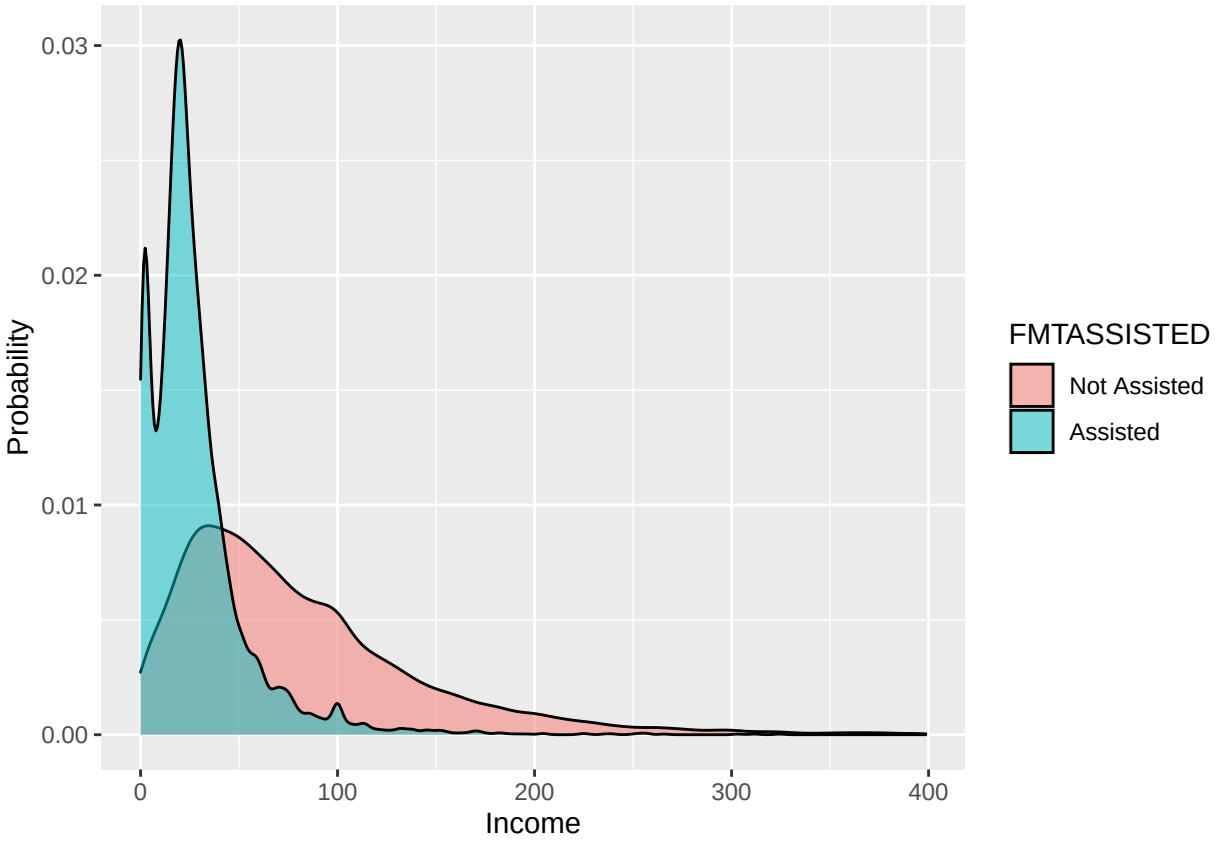
This plot shows that as households incomes increase, the likelihood that they receive assistance is lower, which is quite intuitive.

Housing Cost Effects on Assistance status



This plot shows the trend that households with lower housing costs are more likely to receive assistance.

Income distribution broken up by Assistance Status



This plot shows that households with lower income are more likely to be assisted.

Besides the plots shown, other exploratory analyses included the following, but did not show any particular insights or useful interactions:
Age vs Monthly Housing Cost
MOBILEHOME status vs Assistance
Housing Adequacy vs Assistance
Housing Adequacy vs Income
Urban status vs Assistance
Urban status vs Income and Housing Costs

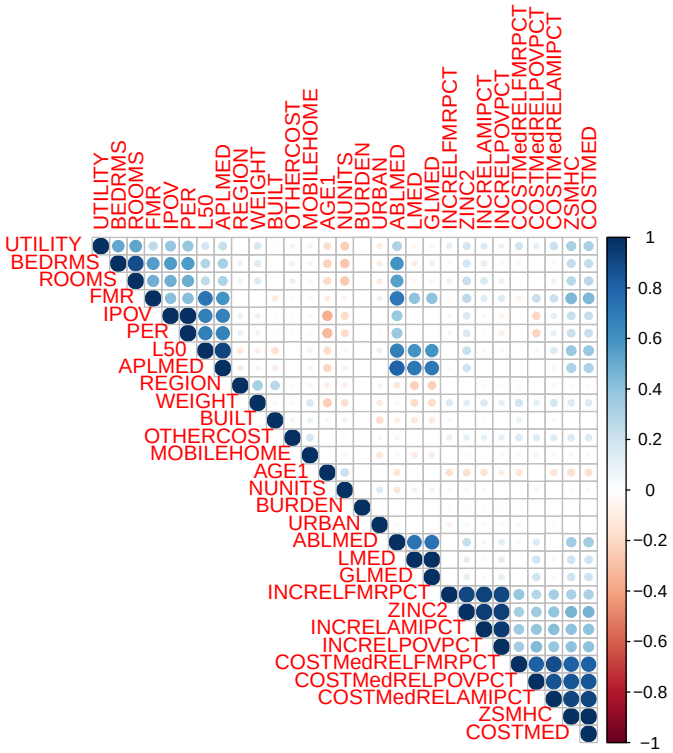
Assessment of Mulit-collinearity

To eliminate redundant variables, and prepare the data for logistic regression, we checked all continuous variables in our "cleaned" data set for collinearity.

```
#restrict data to only continuous variables
cont.data = housingClean[,c(1:27,34,35)]

#create correlation matrix (with p-value matrix)
corrmat <- rcorr(as.matrix(cont.data))

# Create corrlation plot, insignificant correlations are left blank
corrplot(corrmat$r, type="upper", order="hclust",
         p.mat = corrmat$P, sig.level = 0.01, insig = "blank")
```



After computing correlation coefficients of all variables, we discovered two pairs of collinear variables:
IPOV + PER = 1.00
BEDRMS + ROOMS = 0.99

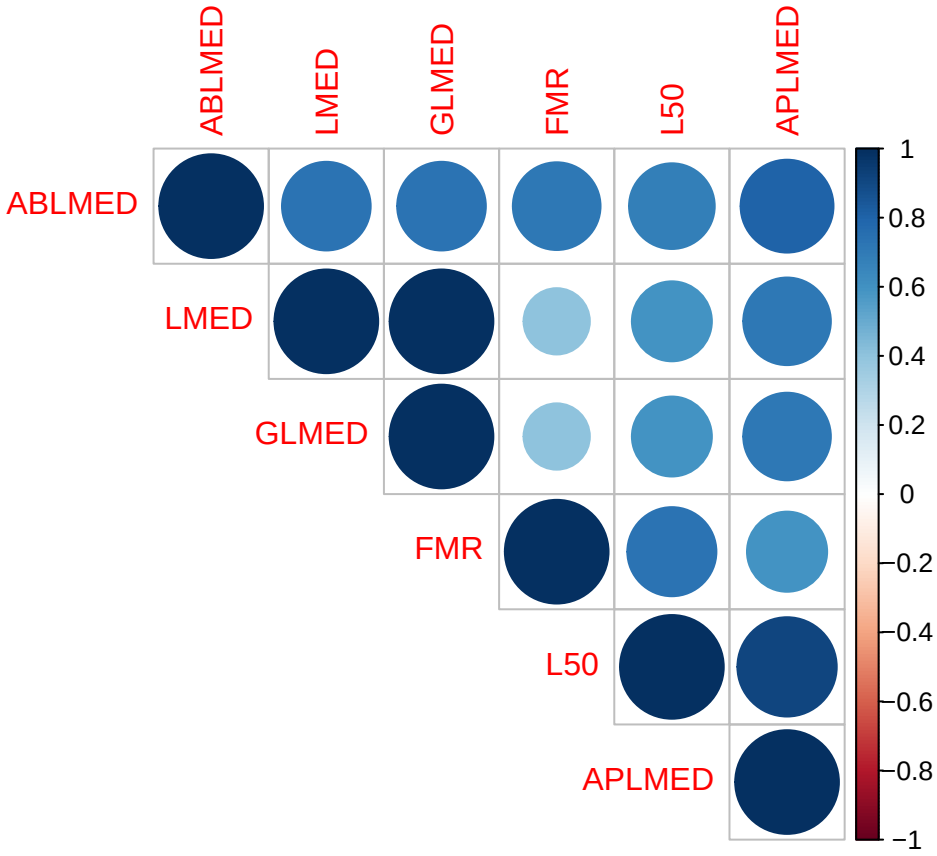
as well as three groupings of collinear variables, which are each visualized, with specific coefficients listed.

Variables related to Area median income and Fair Market Rate

FMR, L50, LMED, GLMED, APLMED, ABLMED

```
#restrict data to "Median Income" related variables
lmed.data = housingClean[,c('FMR', 'L50', 'LMED', 'GLMED','APLMED','ABLMED')]
lmed.corr <- rcorr(as.matrix(lmed.data))

# Plot correlation viz where insignificant correlations are left blank
corrplot(lmed.corr$r, type="upper", order="hclust", p.mat = lmed.corr$P, sig.level = 0.01, insig = "bla
```



LMED + GLMED = 1.00
FMR + L50 = 0.89
FMR + APLMED = 0.86
FMR + ABLMED = 0.91
L50 + APLMED = 0.98
L50 + ABLMED = 0.89
APLMED + ABLMED = 0.93
APLMED + GLMED = 0.71
APLMED + LMED = 0.71
LMED + ABLMED = 0.73
LMED + FMR = 0.41
LMED + L50 = 0.59

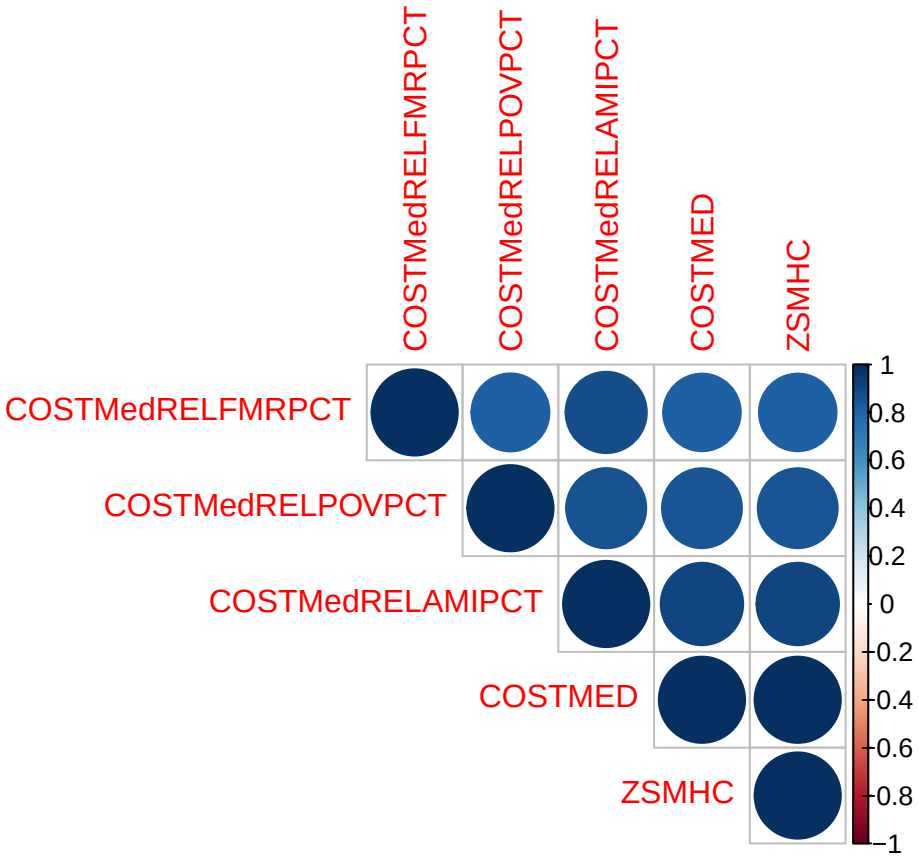
From this grouping we chose to keep `APLMED` (Median Income Adjusted for # of Persons) because it is representative of this whole grouping of collinear variables, since it is the most correlated with other variables in the group. We also decided to keep `FMR` because it is reasonably different in its definition, and its coefficient with `APLMED` of 0.86 is not quite over the threshold of 0.9. We also chose to keep `L50` for similar reasons, as its definition is also different.

Housing Cost Variables

`COSTMED`, `COSTMedRELAMIPCT`, `COSTMedRELPOVPCT`, `COSTMedRELFMRPCT`, `ZSMHC`

```
#restrict data to "Cost" related variables
costmed.data = housingClean[,c('COSTMED', 'COSTMedRELAMIPCT', 'COSTMedRELPOVPCT', 'COSTMedRELFMRPCT', '
costmed.corr <- rcorr(as.matrix(costmed.data))
```

```
# Plot correlation viz where insignificant correlations are left blank
corrplot(costmed.corr$r, type="upper", order="hclust", p.mat = costmed.corr$P, sig.level = 0.01, insig =
```



ZSMHC + COSTMED = 1.00
ZSMHC + COSTMedRELAMIPCT = 0.96
ZSMHC + COSTMedRELPOVPCT = 0.93
ZSMHC + COSTMedRELFMRPCT = 0.92
COSTMED + COSTMedRELAMIPCT = 0.96
COSTMED + COSTMedRELPOVPCT = 0.93
COSTMED + COSTMedRELFMRPCT = 0.92
COSTMedRELAMIPCT + COSTMedRELPOVPCT = 0.96
COSTMedRELAMIPCT + COSTMedRELFMRPCT = 0.98
COSTMedRELPOVPCT + COSTMedRELFMRPCT = 0.96

From this grouping, we chose to keep `ZSMHC` because, as seen in the first exploratory plots, this variable shows a stark gap between most of the households, and a few outliers over $4000. Because of this we also chose to subset our data without these outlier points.

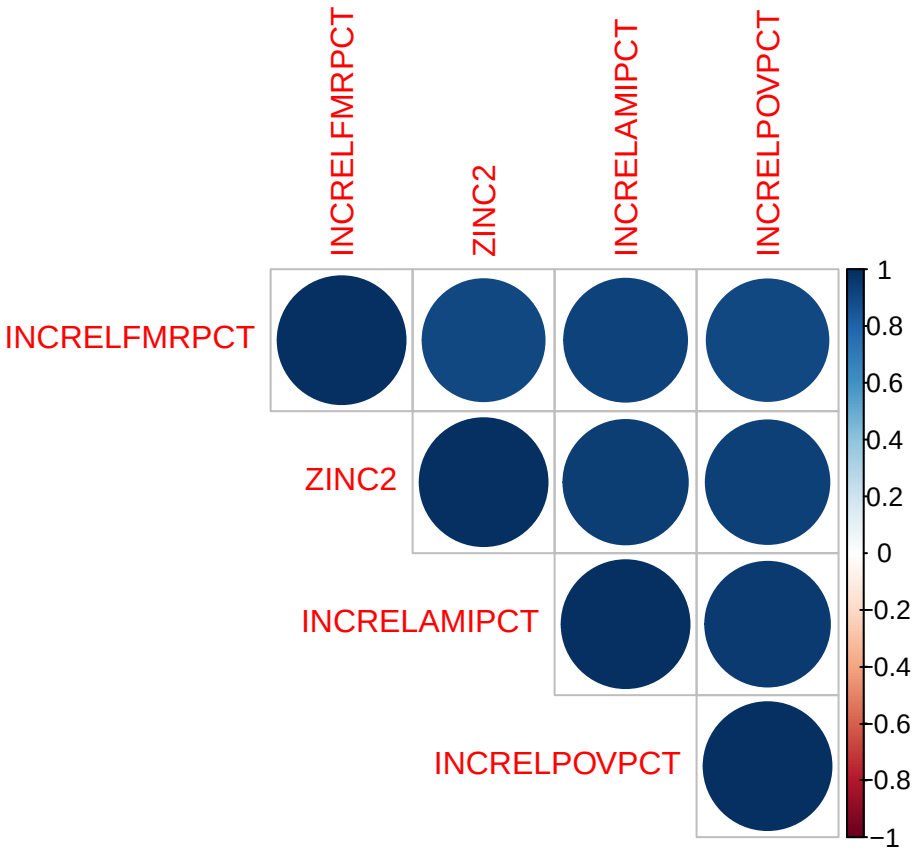
Income Variables

`ZINC2`, `INCRELAMIPCT`, `INCRELPOVPCT`, `INCRELFMRPCT`

```
#restrict data to "Income" related variables
income.data = housingClean[,c('ZINC2', 'INCRELAMIPCT', 'INCRELPOVPCT', 'INCRELFMRPCT')]
income.corr <- rcorr(as.matrix(income.data))

# Plot correlation viz where insignificant correlations are left blank
```

```
corrplot(income.corr$r, type="upper", order="hclust", p.mat = income.corr$P, sig.level = 0.01, insig =
```



ZINC2 + INCRELAMIPCT = 0.97
ZINC2 + INCRELPOVPCT = 0.97
ZINC2 + INCRELFMRPCT = 0.96
INCRELAMIPCT + INCRELPOVPCT = 0.98
INCRELAMIPCT + INCRELFMRPCT = 0.99
INCRELPOVPCT + INCRELFMRPCT = 0.98

From this grouping, we chose to keep `INCRELAMIPCT` because according to the data set documentation, housing cost relative to AMI is the most common standard used in affordability discussions of the three standards provided (fair market rent - FMR, area median income - AMI, and poverty-level income - POV).

Our final data subset

Based on our understanding of the data gained from exploratory analysis and collinearity checked, we have limited our data to 18 variables that we will use to fit predictive models.

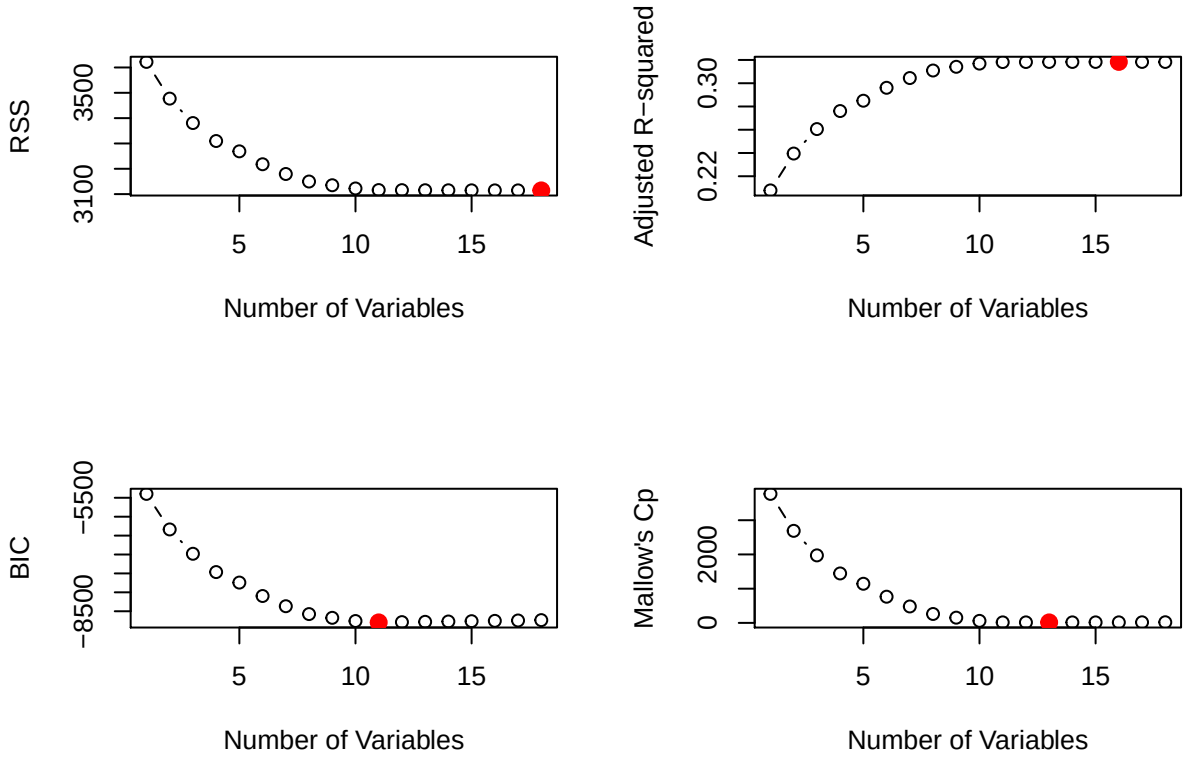
```
housingClean = housingClean[,c("AGE1", "REGION", "FMR", "L50", "BEDRMS", "BUILT", "NUNITS", "PER", "ZSM
```

Model Development

Linear Regression

As a first step in analyzing the data, a set of simple linear regression classifiers was generated. Each model was calculated using a subset of the predictors, ranging from a single predictor to all available features in the dataset. For each model, the best set of predictors was selected to minimize the residual sum of squares using

the "exhaustive" method, comparing all possible combinations. The plots below show four measures of model error plotted against the number of predictors used in the model. The best model identified by each error measure is highlighted in red.



As shown above, RSS decreased with the inclusion of each additional predictor, as expected: the subset of predictors included in each model was selected to minimize the RSS. In contrast, the highest value of adjusted R-squared and the lowest values of Mallow's Cp and BIC were obtained for models containing 16, 13, and 11 out of the total 18 predictors, respectively, as these measures include a penalty for added complexity.

The greatest adjusted R-squared value achieved by any of the models was only 0.318, confirming that a linear model cannot effectively classify households as assisted or unassisted. This was expected due to the significant nonlinearities in the data which were discovered during exploratory data analysis. Of course, an adjusted R-squared value does not actually capture the misclassification rate - an ROC curve was constructed showing the model's performance given various cutoff points, for a more direct measure of the linear model's potential performance as a classifier. The ROC curve is included in the model comparison section.

Due to the low accuracy of the linear models, additional steps to improve them, such as cross-validation, were not completed. However, they are discussed briefly here as a baseline comparison for the more appropriate models which were subsequently developed. Additionally, the five variables which produced the best linear regression model were identified for comparison to the variables found to be most significant in the subsequent models. Their names and coefficients are displayed below. The best subsets regression process was repeated using the "forward" and "backward" methods, and the same five variables were identified as the most important each time.

```
#### Compare exhaustive best subset regressions with forward and backward best subset methods


# Calculate the best subset regressions using the forward and backward methods, for comparison
```

```
# to the best subset regressions calculated using the exhaustive method.
regfit.fwd <- regsubsets(FMTASSISTED ~ ., data = housingClean, nvmax = 17, method = "forward")
regfit.bkwd <- regsubsets(FMTASSISTED ~ ., data = housingClean, nvmax = 17, method = "backward")


# Print the names of the top 10 variables selected using each method
coef(regfit.full, 5)[-1]
```

```
##          AGE1           FMR        NUNITS         ZSMHC  INCRELAMIPCT
##  0.0031590114  0.0001303167  0.0009448162 -0.0003395712 -0.0008131579
```

```
#names(coef(regfit.fwd, 5))
#names(coef(regfit.bkwd, 5))
```

Logistic Regression

Next, a logistic regression model was developed to improve classification of the households. Logistic regression is a more appropriate classification method for the data for a few reasons: First, it allows us to directly estimate the probability of a given household being included in one of the assistance programs captured in the dataset, rather than simply classifying each based on a score as was done with the linear model. Second, and even more importantly, logistic regression does not rely on linear relationships between the predictors and the outcome, and it can easily handle the several categorical and binary predictors in the dataset [James, et al., Introduction to Statistical Learning, 7th ed.).

First, a logistic regression was calculated using all of the available predictors, fitted to the entire data set. This complex and likely overfitted model serves as a reference of the most accurate possible logistic regression model which can be obtained from the data. A confusion matrix of the model's classifications is shown below, along with a table of the model's coefficients. The ROC curve for the model is also shown later in the report, for comparison with the other models.

\begin{table}

\caption{Full Logistic Model Confusion Matrix (misclassification rate = 13.04%)}

	Classification		
True class	Not assisted	Assisted	Sum
Not assisted	15818 (68%)	1174 (5%)	16992 (73%)
Assisted	1857 (8%)	4399 (19%)	6256 (27%)
Sum	17675 (76%)	5573 (24%)	23248 (100%)

\end{table}

```
##                      Estimate   Std. Error     z value
## (Intercept)     -3.669509e+01 1.723289e+00 -21.2936321
## AGE1             1.424466e-02 1.134522e-03  12.5556527
## REGION          -1.604672e-01 2.080726e-02  -7.7120772
## FMR              1.627773e-04 1.629379e-04   0.9990146
## L50              4.685751e-05 1.412170e-05   3.3181219
## BEDRMS           1.191616e-01 4.521769e-02   2.6352871
## BUILT            1.900736e-02 8.737616e-04  21.7534841
## NUNITS           4.141629e-03 3.163893e-04  13.0902919
## PER              8.690433e-02 3.239895e-02   2.6823197
## ZSMHC           -2.925621e-03 6.693377e-05 -43.7091838
## UTILITY         -2.113266e-03 2.803469e-04  -7.5380402
## OTHERCOST       -1.797424e-02 2.341011e-03  -7.6779801
## BURDEN          -6.605932e-05 1.126555e-04  -0.5863837
```

```
## APLMED                         -1.663884e-05 4.644368e-06  -3.5825832
## INCRELAMIPCT                   -2.208578e-02 7.006714e-04 -31.5208735
## FMTZADEQ2 Moderately Inadequ -1.449838e-01 8.057487e-02  -1.7993670
## FMTZADEQ3 Severely Indadequa -2.496086e-01 1.122728e-01  -2.2232335
## MOBILEHOME                     -2.296539e+00 2.229617e-01 -10.3001496
## URBAN                           3.728275e-01 4.128536e-02   9.0305025
##                                     Pr(>|z|)
## (Intercept)                    1.300479e-100
## AGE1                            3.701165e-36
## REGION                          1.237862e-14
## FMR                             3.177876e-01
## L50                             9.062494e-04
## BEDRMS                          8.406615e-03
## BUILT                          6.402777e-105
## NUNITS                          3.741673e-39
## PER                             7.311355e-03
## ZSMHC                           0.000000e+00
## UTILITY                         4.770869e-14
## OTHERCOST                       1.616165e-14
## BURDEN                          5.576177e-01
## APLMED                          3.402132e-04
## INCRELAMIPCT                   4.496822e-218
## FMTZADEQ2 Moderately Inadequ  7.196064e-02
## FMTZADEQ3 Severely Indadequa  2.620006e-02
## MOBILEHOME                      7.035177e-25
## URBAN                           1.708853e-19
```

Next, a few steps were taken to increase the robustness of the model. A lasso was applied to reduce the complexity of the model by limiting the number of terms, and the model was trained using only half of the dataset, which was randomly partitioned. The plot below shows the relationship between the cross-validation mean squared error and the level of complexity of the model, lambda.

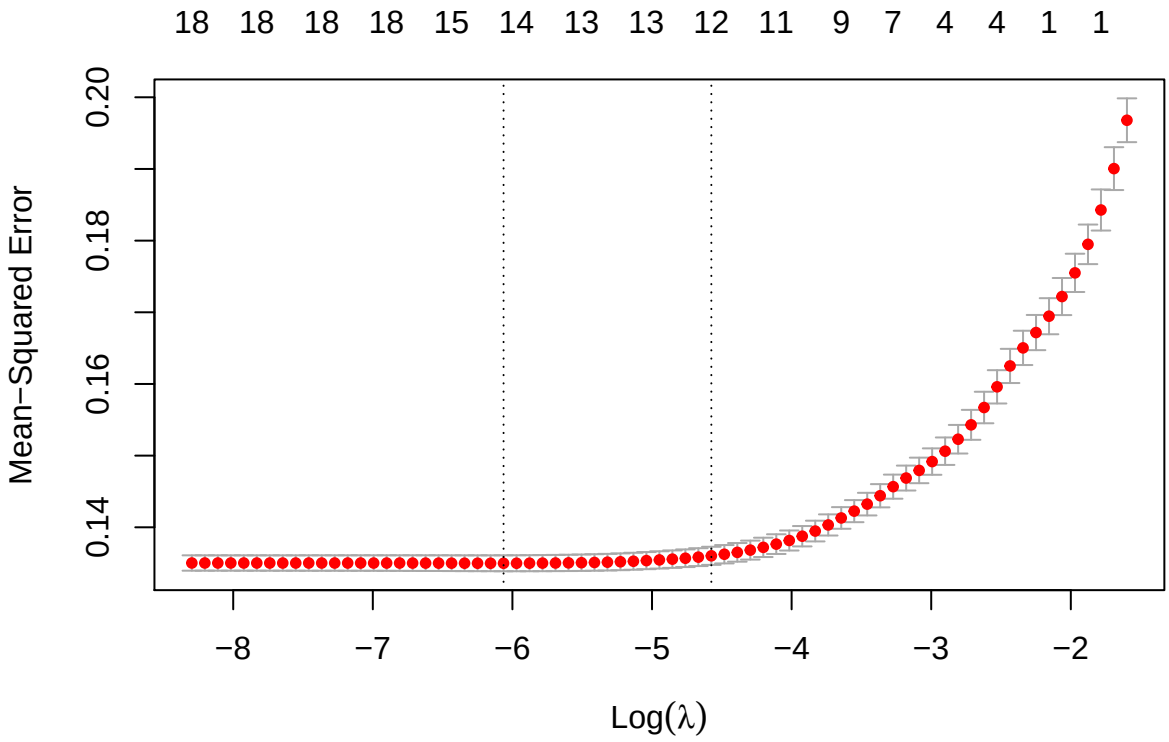
```
#### Use a lasso to limit the number of terms in the logistic regression model

# !NOTE! This method converts all qualitative variables into dummy variables
# Prepare a predictors matrix and an outcomes vector
x <- model.matrix(FMTASSISTED ~ ., housingClean)[, -1]
y <- as.numeric(housingClean$FMTASSISTED) - 1 # Convert assisted/not assisted to binary dummy

# Split the data into a training set and a test set
set.seed(1)
train <- sample(1:nrow(housingClean), nrow(housingClean)/2)
test <- (-train)
y.test <- y[test]

# Apply a lasso
lambda.grid <- 10^seq(10, -2, length = 100)
lasso.mod <- glmnet(x[train,], y[train], alpha = 1, lambda = lambda.grid, family = "binomial")
# plot(lasso.mod)

# Determine best lambda value to minimize misclassification rate (and hence, determine the best number
cv.out <- cv.glmnet(x[train,], y[train], alpha = 1)
#bestlam <- cv.out$lambda.min
```



A lambda value of 0.018 was selected, in order to reduce the number of terms to 10. The use of the lasso allows us to select the 10 variables which result in the lowest error rate. The confusion matrix is shown below, along with a table showing the coefficients of the 10 variables which were selected, with the variables which were removed from the model indicated by a dot.

```
# !NOTE! This lambda value was selected manually to obtain a model containing only the 10 most importan
# The minimum cv error rate is obtained with all of the predictors included.
lam <- 0.018
lasso.mod <- glmnet(x, y, alpha = 1, lambda = lam, family = "binomial")
```

```
#### Extract probability predictions from the size-limited logistic regression model, to plot an ROC cu
lasso.probs <- predict(lasso.mod, newx = x, s = lam, type = "response") # pull "response" values from m
min.p <- min(lasso.probs)
max.p <- max(lasso.probs)
lasso.probs <- (lasso.probs - min.p)/(max.p - min.p) # Rescale response values to 0-1 scale for input i
```

\begin{table}

\caption{Logistic (10 predictors) Model Confusion Matrix (misclassification rate = 13.45%)}

	Classification		
True class	Not assisted	Assisted	Sum
Not assisted	15969 (69%)	1023 (4%)	16992 (73%)
Assisted	2104 (9%)	4152 (18%)	6256 (27%)
Sum	18073 (78%)	5175 (22%)	23248 (100%)

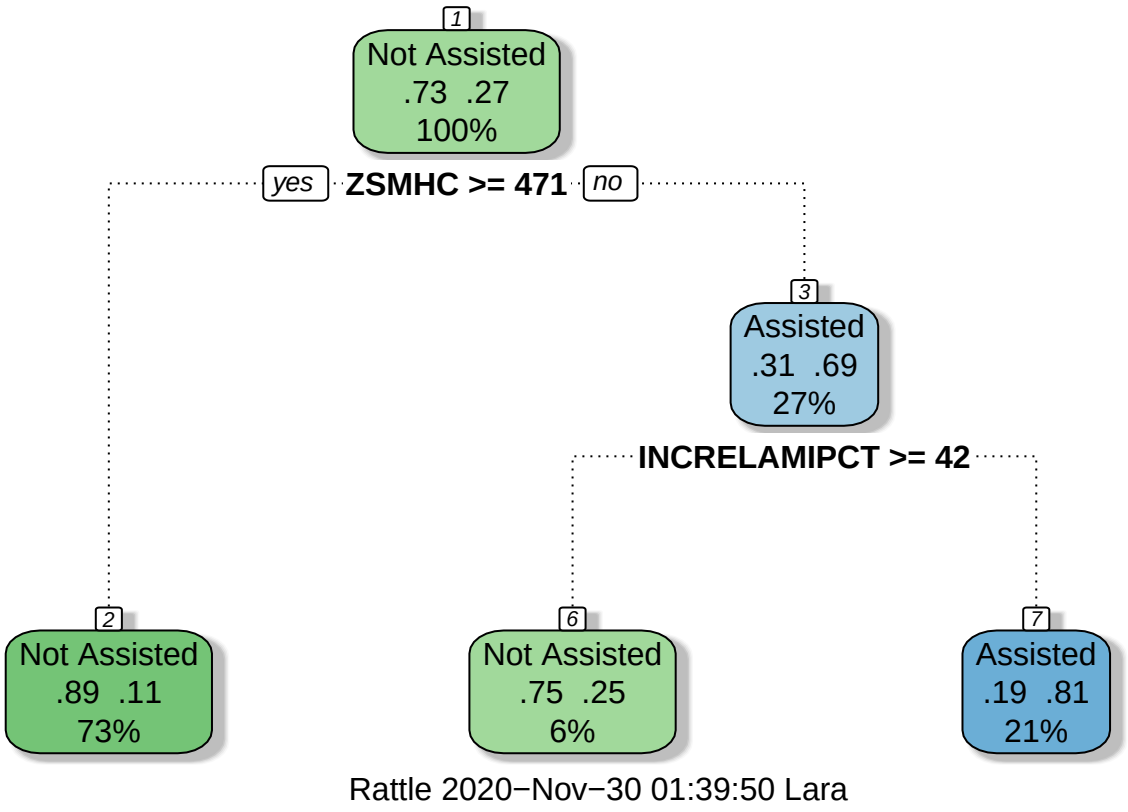
\end{table}

```r
# Determine the number of terms in the size-limited logistic regression model
lasso.coef <- predict(lasso.mod, type = "coefficients")
#length(lasso.coef[lasso.coef != 0]) # 10 terms included in the model
lasso.coef
```

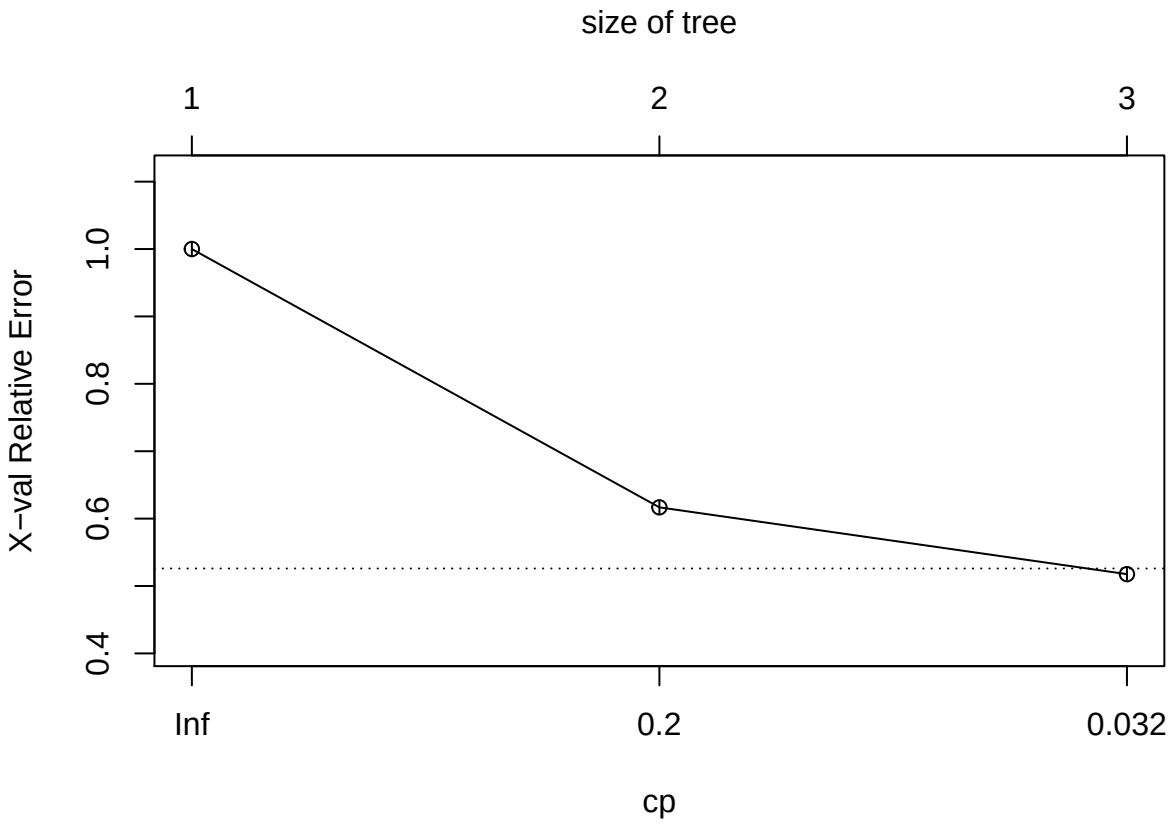
```
## 19 x 1 sparse Matrix of class "dgCMatrix"
##                                          s0
## (Intercept)                  -1.464768e+01
## AGE1                          8.561073e-03
## REGION                        .
## FMR                           .
## L50                           7.736204e-06
## BEDRMS                        .
## BUILT                         7.729996e-03
## NUNITS                        2.845732e-03
## PER                           .
## ZSMHC                        -2.321571e-03
## UTILITY                      -6.875235e-04
## OTHERCOST                     .
## BURDEN                        .
## APLMED                        .
## INCRELAMIPCT                 -1.323108e-02
## FMTZADEQ2 Moderately Inadequ  .
## FMTZADEQ3 Severely Indadequa  .
## MOBILEHOME                   -7.615301e-01
## URBAN                         1.793330e-01
```

Classification Tree Approach

Single Tree

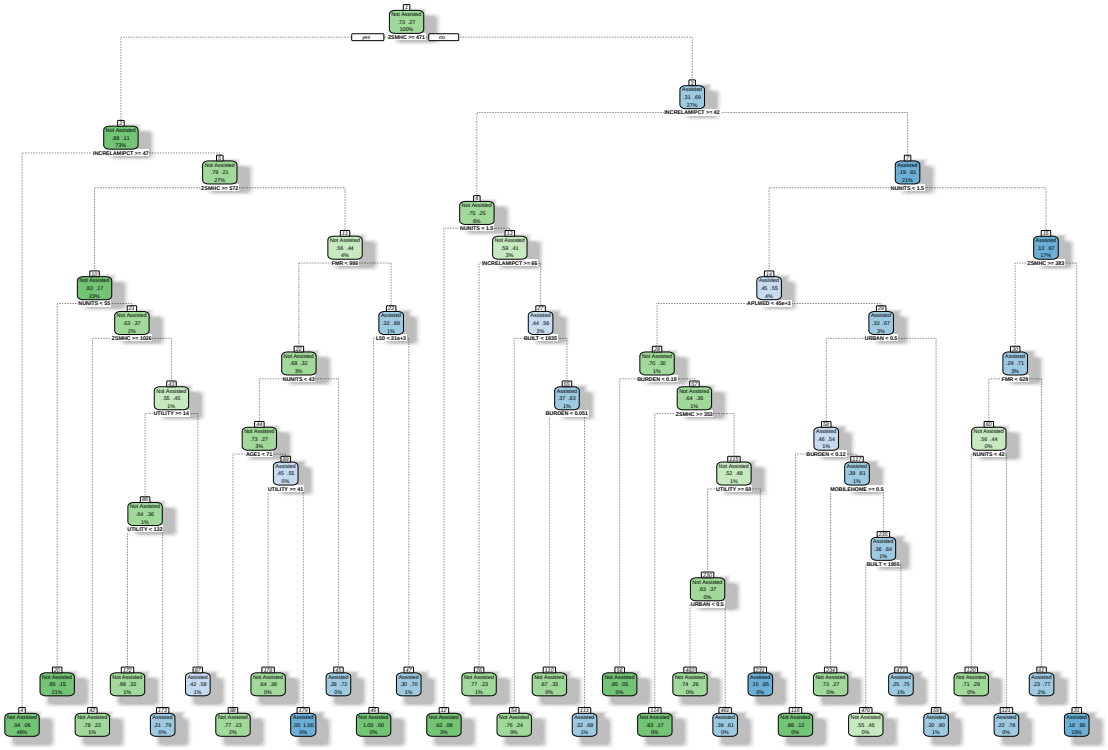


To begin with we ran a simple classification tree. This produced a fairly strong result, with a misclassification rate of only 13.78%. An initial concern was that the tree used so few predictors out of the overall set of 17, we may not be maximizing the predictive power we could be getting out fo the data. To assess that we created a complexity parameter plot to assess potential performance.



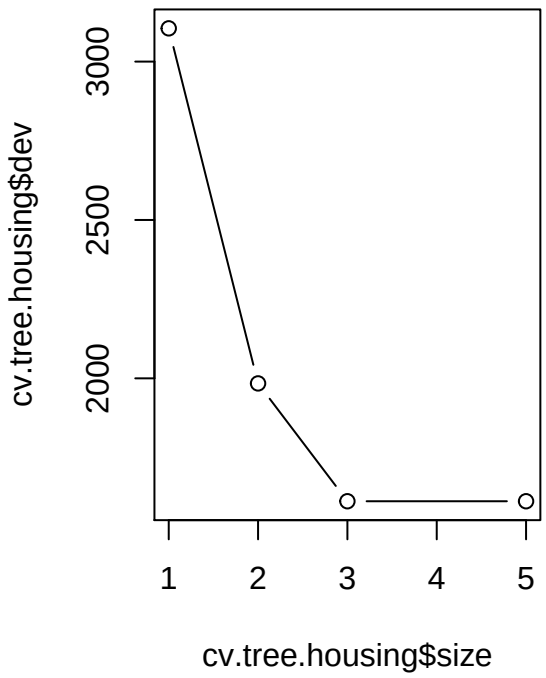
This plot shows us that the relative error rate fails to decrease substantially as we increase the number of predictors, which is why the tree automatically pruned itself to the level.

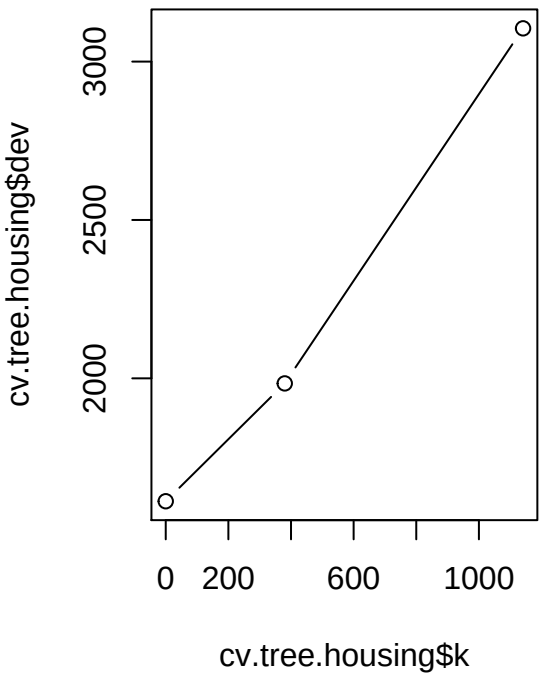
As an additional check, we induced a lower complexity parameter to see what the tree would look like incorporating more factors and what the misclassifcation rate would be.



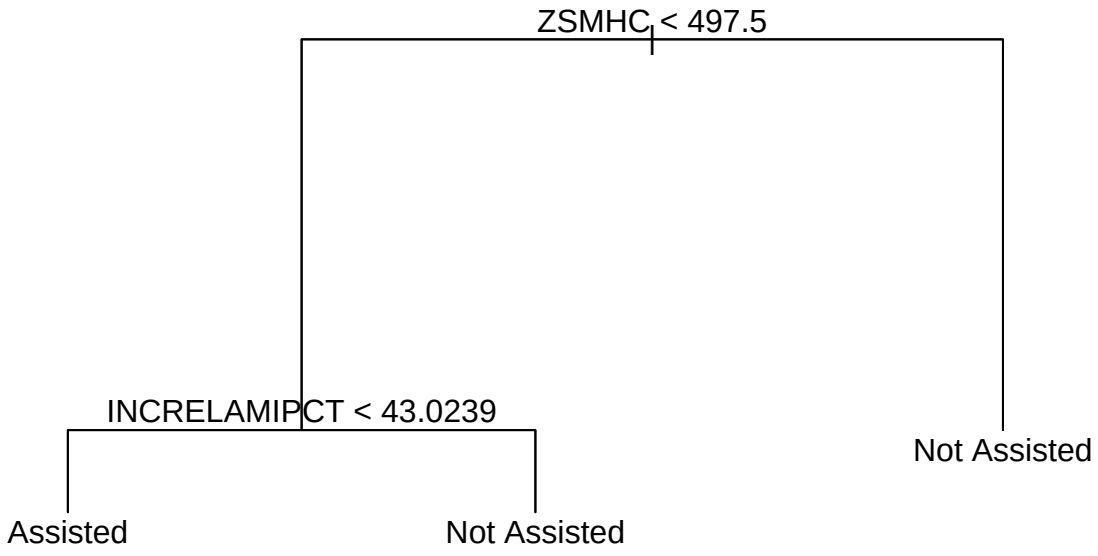
We see here a significantly more complex classification tree, but a misclassification rate that is 11.4%, a rather minor improvement. This shows us why the initial classification tree automatically restricted itself to the size that it did. This helps to prevent overfitting from occuring and is a reasonable trade-off for a minor decrease in accuracy.

Cross Validation & Pruning The next stage in addressing the issue of potential overfitting is to perform some cross validation.





After perfoming this analysis, we can see that the deviation in our sample's deviation stops decreasing once the tree reaches an approximate size of 3. Our inital classificatoin tree used a size of 5, and so we see here an opportunity to prune the tree and reach similar results and still decrease the risks of overfitting.



We see now that by pruning the tree to a size of 3, the tree is extremely easy to interpret. The expectation based on the plot of deviation is that it should have a nearly identical misclassificaiton rate as our base tree.

```
##                assist.test
## tree.pred       Not Assisted Assisted
##   Not Assisted         7952     1098
##   Assisted              521     2053
```

We validated that assumption by creating the confusion matrix above, which resulted in the expected outcome, which is one that has nearly identical performance to the initial tree. The misclassication rate for the pruned tree was 13.93%, compared to the unpruned rate of 13.78%, showcasing the essentially identical performance.

Bagging and Boosting While the performance of a single pruned tree was good, we wanted to see if we could improve upon it in a way that does not result in overfitting. To do this, we want to complete the bagging and boosting process.

To begin with, we will create a random forest with 5000 trees and see how it performs.

```
set.seed(1)

boost.housing <- gbm(FMTASSISTED ~ ., housingClean[train,], distribution = "gaussian", n.trees = 5000, i

rf.model2 <- boost.housing
rf.probs2 <- predict(rf.model2, housing.test, type = "response") # Extract "response" values from rando
```

```
## Using 541 trees...
```

```
min.p <- min(rf.probs2)
max.p <- max(rf.probs2)
rf.probs2 <- (rf.probs2 - min.p)/(max.p - min.p) # Rescale response values to 0-1 scale for input into
rf.matrix2 <- conf.matrix(rf.probs2, housing.test$FMTASSISTED, "Initial GBM")
rf.matrix2
```

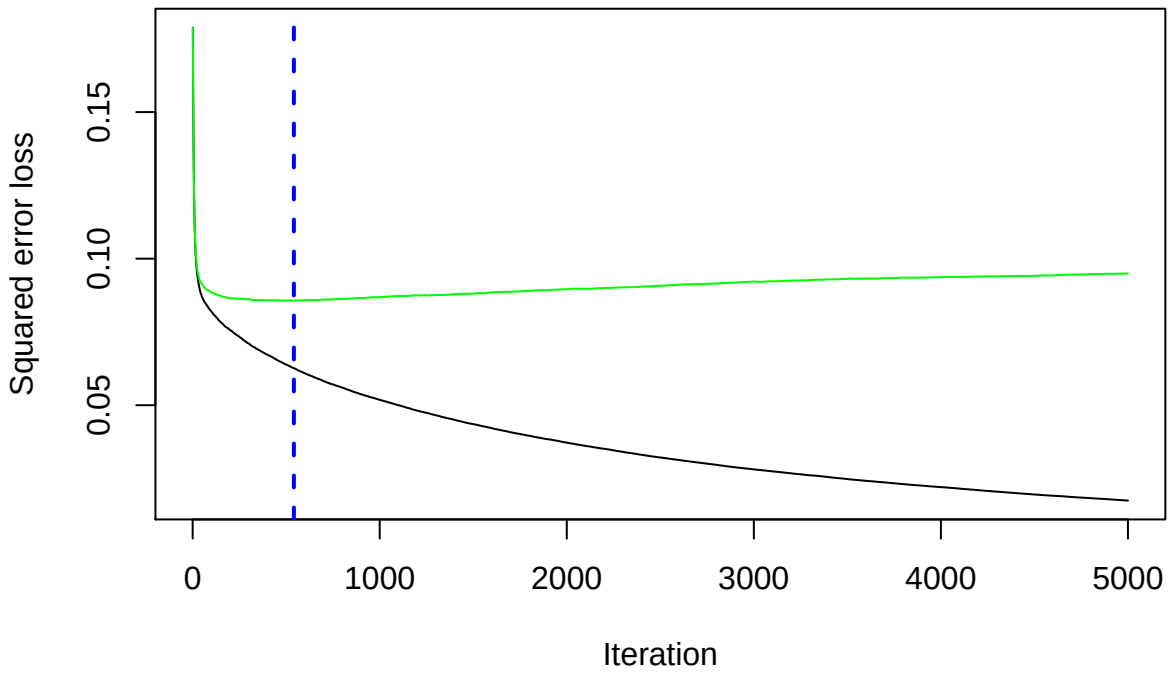
\begin{table}

\caption{Initial GBM Model Confusion Matrix (misclassification rate = 10.8%)}

	Classification		
True class	Not assisted	Assisted	Sum
Not assisted	8101 (70%)	372 (3%)	8473 (73%)
Assisted	883 (8%)	2268 (20%)	3151 (27%)
Sum	8984 (77%)	2640 (23%)	11624 (100%)

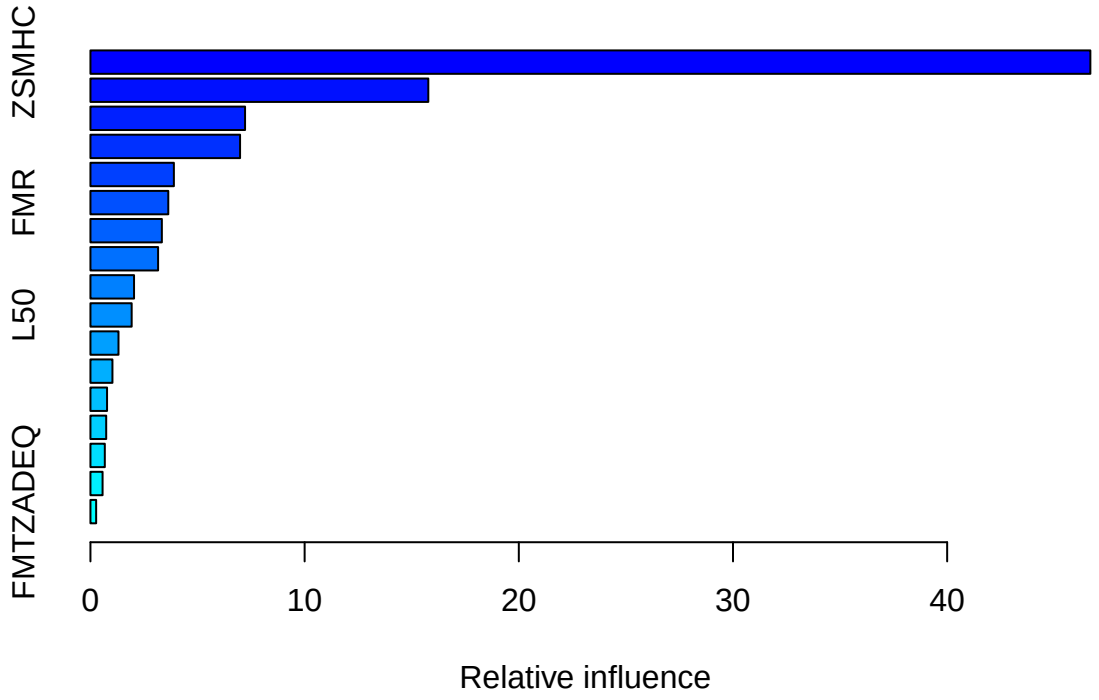
\end{table}

It performs well and shows an improvement in misclassification rate, as you can see in the confusion matrix. This is because this model is undergoing the boosting process, allowing for 5,000 trees to be grown and tested. This results in an improved classification rate over the previous models because we're allowing for multiple different kinds of trees where we can sample multiple different predictors, even ones that may not initially seem to have much predictive power.



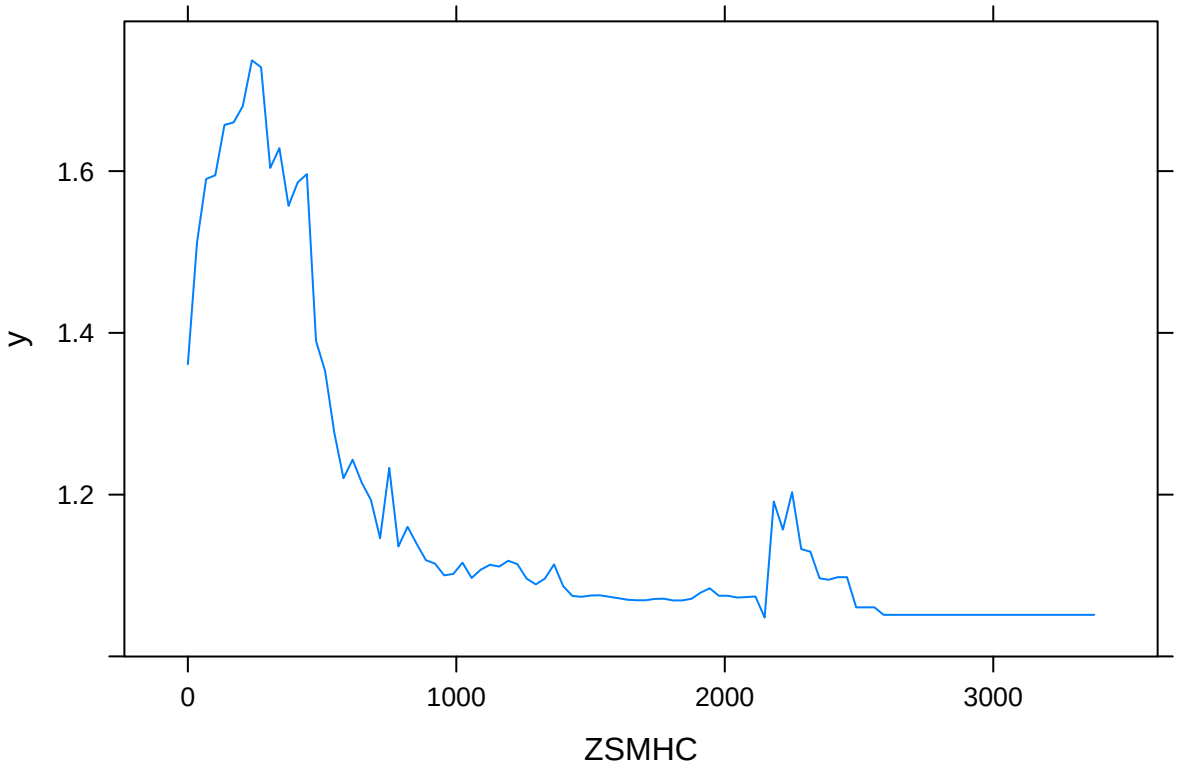
Even though we grew 5,000 trees, that does not mean that each subsequent tree improves accuracy. There is likely to be some number of trees between 1 and 5,000 that yields the best results. In this case, we can store the optimal number for minimizing errors and we can construct our final model using that number of trees.

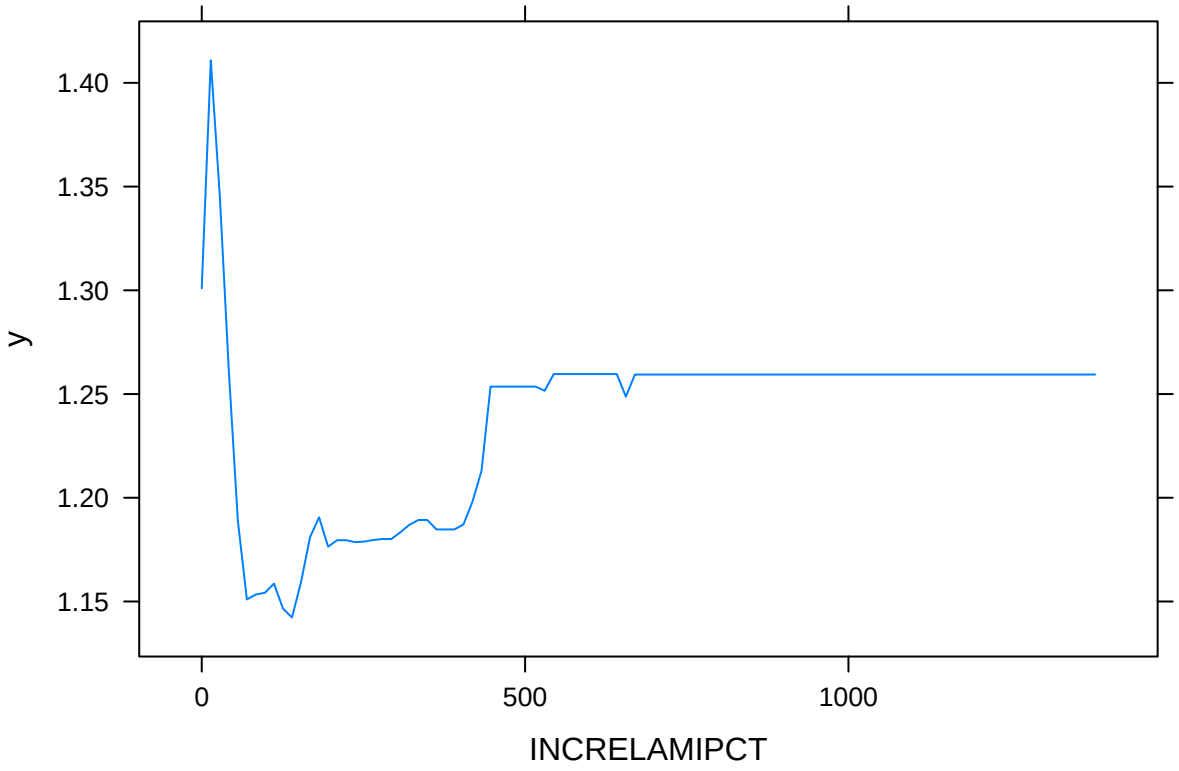
```
## Using 541 trees...
```




```
##                        var    rel.inf
## ZSMHC                ZSMHC 46.6853815
## INCRELAMIPCT INCRELAMIPCT 15.7750267
## BURDEN             BURDEN  7.2182994
## NUNITS             NUNITS  6.9855607
## UTILITY           UTILITY  3.8960592
## FMR                   FMR  3.6341967
## AGE1                 AGE1  3.3340581
## BUILT               BUILT  3.1581200
## APLMED             APLMED  2.0360866
## L50                   L50  1.9249866
## BEDRMS             BEDRMS  1.3099323
## PER                   PER  1.0274385
## URBAN               URBAN  0.7758080
## OTHERCOST       OTHERCOST  0.7353898
## REGION             REGION  0.6709422
## MOBILEHOME     MOBILEHOME  0.5667411
## FMTZADEQ         FMTZADEQ  0.2659727
```

The next part in evaluating the model is to see which factors are contributing most significantly to its success. We can see here that the monthly housing units (ZSMHC) and the income relative to area median income (INCRELAMIPCT) contribute most to the accuracy of the model. We can inspect their impact even closer here.


y
1.6
1.4
1.2
0
1000
2000
3000
ZSMHC



Compare and Contrast

The final step in our statistical analysis was to compare the generated models and select the best classifier. This not only means selecting the most accurate model, but verifying that it is making sound judgments based on logical variables which should be relevant to whether an individual receives housing assistance. This holistic comparison confirmed that the optimized random forest model was the best suited for the task.

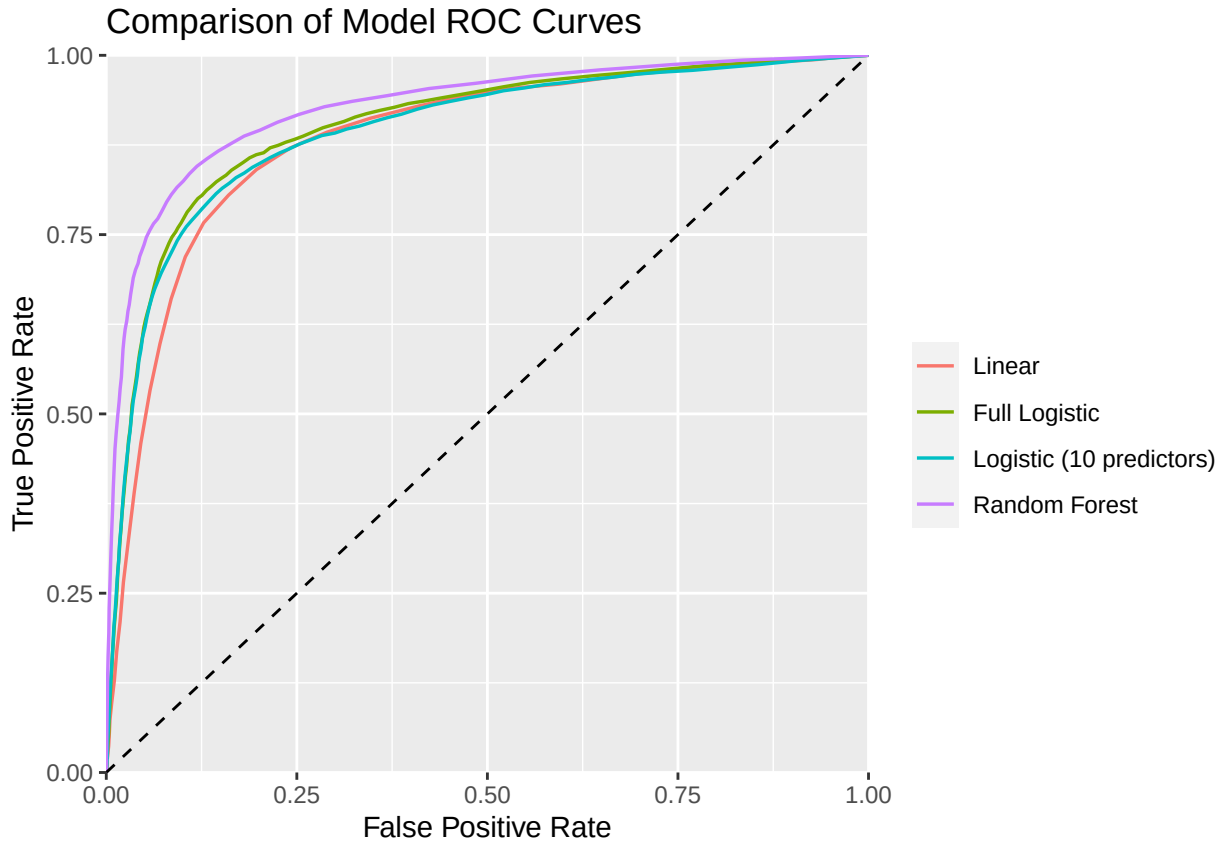
As discussed previously, each successive model demonstrated improvement in reducing the misclassification rate. Classification by linear regression (using an optimally selected "cutoff score") proved to be extremely ineffective at this particular classification problem, due to the complexity of the relationships between the variables recorded for the dataset and the receipt of housing assistance. A logistic model was a good start, providing a reasonably accurate classifier. The logistic model suffered only a minimal loss of accuracy after greatly reducing the complexity of the model, providing a fairly accurate and robust model which could be applied quite effectively to this problem. The accuracy of the logistic model was roughly matched by a simple classification tree, which also proved quite resilient to forced reduction in complexity, so a robust random forest model was developed using bagging and boosting, resulting in an extremely effective classifier. This is the model which was ultimately selected. The ROC curve below was generated for a more thorough comparison of the accuracy of each model.

```
#### Generate ROC curves using the previously-defined function

# Create a data frame for each model's ROC curve
lin.roc.df <- roc.curve(regsubsets.predictions, housingClean$FMTASSISTED, "Linear")
log.roc.df <- roc.curve(glm.probs, housingClean$FMTASSISTED, "Full Logistic")
lasso.roc.df <- roc.curve(lasso.probs, housingClean$FMTASSISTED, "Logistic (10 predictors)")
rf.roc.df <- roc.curve(rf.probs, housing.test$FMTASSISTED, "Random Forest")
```

```r
# All ROC curve dataframes should be binded together, in order to be displayed
# on a single graph with a legend.
all.roc.curves <- rbind(lin.roc.df, log.roc.df, lasso.roc.df, rf.roc.df)

# Plot all of the ROC curves using the all.roc.curves data frame
ggplot(data = all.roc.curves, mapping = aes(x = FPR, y = TPR, color = model)) +
  geom_line(size = 0.6) +
  geom_abline(slope=1, intercept=0, linetype = "dashed") +
  theme_gray() +
  scale_x_continuous(expand = c(0,0)) +
  scale_y_continuous(expand = c(0,0)) +
  labs(title = "Comparison of Model ROC Curves",
       x = "False Positive Rate",
       y = "True Positive Rate",
       col = "")
```



Finally, the importance of each predictor in the final model was evaluated to ensure that the variables being used were logical and appropriate. Each important variable was found to have a clear and direct connection to the awarding of housing assistance, with the most important variables being the monthly housing costs, income (relative to the area's median income), and the number of units in the building – all having reasonable, direct relationships with assisted housing.

Discussion

Ultimately, we found that this problem was not as complex to classify as we may have initally believed. Early theories may have been that things like regionality, the quality of the housing, and other characteristics may have played a large role in the classification process turned out not to be true. Ultimately, the two biggest factors ended up being strictly monetary concerns with monthly housing costs and how an individual's income relates to the median for their area.

There are a number of reasons why this could be, but the most likely is that the government's definition for whether someone requires assistance or not primiarly depends on a person's income and their expenses. While this makes sense as they are objective measures that are easy to track and plug into formulas to determine assistance, it does indicate an opportunity for further study. What beyond simplistic measures of income and expenses can we look at to identify others who need assistance?

Ultimately, regarding our initial question of trying to identify households that may require assitance that are not receiving it, we have developed a list of the indexes for these titled `needHelp`, which is comprised of 606. Looking further at these households may be warranted to see if they have needs that could be met by current services. We have also created the list `examineCloser` with 1596 for individuals that may currently be using the system in a way that it was not intended. Looking at these cases in more detail may help improve the program and ensure that aid is reaching those who need it most.