CHICAGO CRIME ANALYSIS

Jeffrey Akiki (ja3207)

Yerin Jung (yj2477)

Aditi Wadhawan (aw3099)

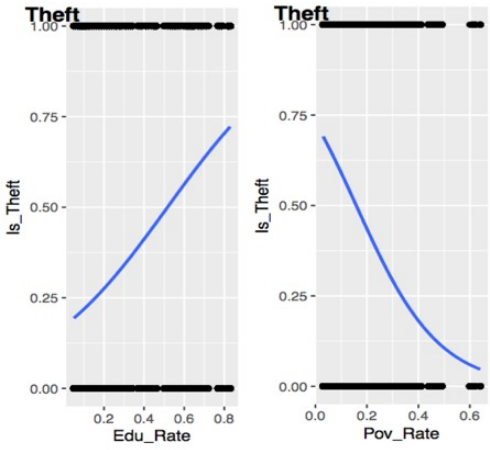
Approach to Problem

In Deliverable 1, we analyzed the historical crime data for Chicago city, using descriptive analysis to identify correlations and past trends. However, we were not able to identify significant correlations among the dependent and independent variables within our original dataset. To overcome this issue, we used socio-economic information from *American Community Survey* dataset and merged it with our dataset from the years 2009 to 2013. We used this merged dataset to predict which type of crime is more likely to occur and are sharing our recommendations based on this.

Recommendation and Conclusion

THEFT

According to our multinomial logistic regression model, in Chicago, the relative risk ratio for a one-unit increase in the variable `Edu_Rate` (education rate) is 2.2278575 for `THEFT` VS. `BATTERY`. In addition, the relative risk ratio for a one-unit increase in the variable `Pov_Rate` (poverty rate) is 0.2977866 for `THEFT` VS. `BATTERY`. Thus, we recommend focusing on areas with high education rates to reduce theft incidents. Similarly, we should focus on areas with low poverty rates too, to decrease theft incidents. We can also see these correlations in <Table1>.



<Table 1>

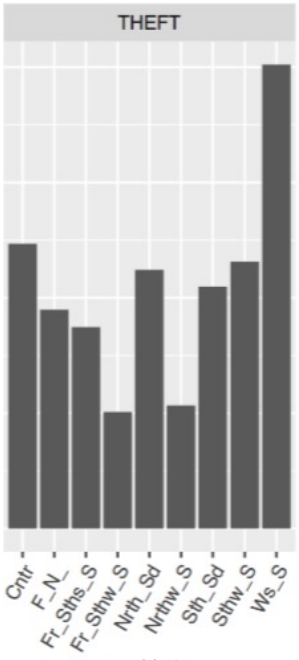
For detailed recommendations, we decided to examine five areas with the lowest poverty rate and five areas with highest education rate as well, by organizing the original data set.

The five areas with lowest poverty rate were 9(Edison Park), 74 (Mount Greenwood), 72(Beverly), 12(Forest Glen), and 5 (North Center). At the same time, the five areas with highest education rate were 7 (Lincoln Park), 32 (Loop), 6 (Lake View), 8 (Near North Side), and 41 (Hyde Park). <Table 2>

Rank	Community Area	Poverty Rate	Rank	Community Area	Education Rate
77	9	0.030	1	7	0.882
76	74	0.036	2	32	0.790
75	72	0.040	3	6	0.782
74	12	0.054	4	8	0.776
73	5	0.060	5	41	0.710

<Table 2: THEFT>

As we all know, each Chicago side includes several community areas. Amongst the 10 community areas that we mentioned above, 5, 6, 7, 8, 32, and 41 were included in one of the following sides, West, Centre, North, Southwest, or South, which were the top five sides where theft occurred the most <Table 3>.



<Table 3>

Apart from those areas that already appeared in the top five theft areas, some areas such as 9, 12, 72, and 74 were not included in the TOP 5 sides with high theft incidents. However, they still have either high education rates or low poverty rates. As such, we consider these areas are prone to a high risk of theft.

According to a local newspaper sources (Cherone, 2017) the number of police officers patrolling the high risk theft areas was not as many as other areas <Table 4>.

District	Total Officers	Officers/Sq.	Population/Officer
1st (Central)	313	66.8	182
2nd (Wentworth)	323	43	343
3rd (Grand Crossing)	321	52.8	234
4th (South Chicago)	327	12	378
5th (Calumet)	332	26	224

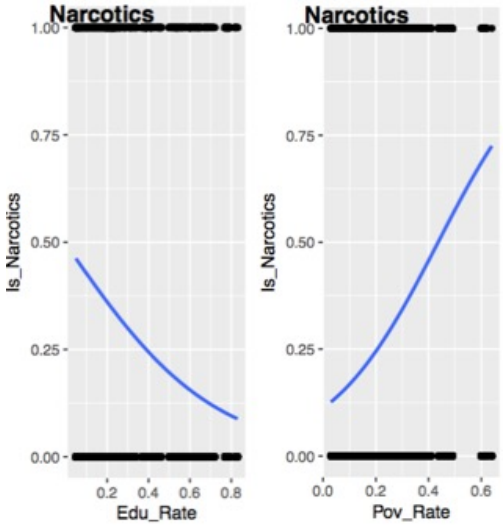
<Table 4: Top 10 Community Areas for the Number of Officers>

This is supported by our findings from the analysis and we would recommend increasing the patrols in these areas.

We expect that the authority will be able to efficiently manage budgets and reduce the amount of social damage through an efficient deployment of patrols across the different areas. Specifically, it is expected that the proportion of theft will be reduced by increasing the deployment of patrols in areas that are potentially and consistently considered to belong to the high-risk group of theft, such as 9, 12, 72, and 74.

NARCOTICS

According to our multinomial logistic regression model, the relative risk ratio for a one-unit increase in the variable `Edu_Rate` (education rate) is 0.1791452 for `NARCOTICS` VS. `BATTERY`. In addition, the relative risk ratio for a one-unit increase in the variable `Pov_Rate` (poverty rate) is 3.9724731 for `NARCOTICS` VS. `BATTERY`. Thus, in order to

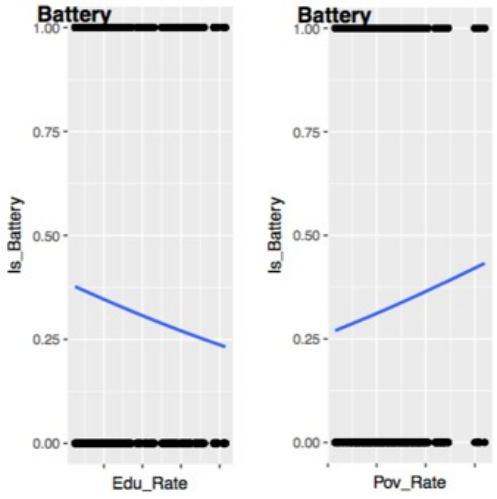


<Table 5>

decrease the occurrence of narcotics incidents, authorities should focus more on the Chicago sides which have high poverty rates. We can also see these correlations in <Table 5>. By this research we were able to find the correlation between high poverty rates and narcotics. However, as it is not an experiment, we cannot impose causality between them. As such, we would like to recommend the city council to conduct an experiment to further understand the relationship between high poverty rates and Narcotics to validate the causality and improve the condition.

BATTERY

As we used Battery as our baseline in the multinomial regression, we could not formulate a proper recommendation. However, we can see from the relative risk ratios that we previously mentioned, Battery is highly correlated with areas that have high poverty rates, and low education rates. We can also see these correlations in <Table 6>. Thus, in order to target battery, authorities should deploy more patrols in these areas.



<Table 6>

Methodology

Concepts of Methodology used

We selected multinomial logistic regression as the methodology for our analysis. We decided to use different variables such as poverty rates and education rates as our independent variables and primary types (Theft, Narcotics and Battery) as our independent variables.

We then used a variable selection method to identify which independent variables to use in our model. We used the AIC values to estimate the better independent variables to be used with our

model, to resolve our previous issue of weak correlation between our dependent and independent variable.

Strength and weakness of methodologies

For our model, there were various merits to use multinomial logistic regression. Since we had polytomous variables it was better to use multinomial logistic regression over logistic regression (Multinomial Logistic Regression, n.d.). It also helped us to predict the nominal dependent variable (crime_type) using independent variables (Poverty, education etc.).

Multinomial logistic regression models are not that straightforward to understand as logistic regression models, we faced challenges in understanding the important data points. We had to interpret the results of the different crime types in reference to another crime type to share our recommendations. It uses maximum likelihood estimation methods and hence required a large sample size (Multinomial Logistic Regression | R, n.d.). We had a large dataset, but faced another challenge that occurrences of a particular crime type over others, irrespective of a strong correlation, can be misinterpreted as significant. We had to use a random and homogeneous sample of all crime types we were analyzing in our model to ensure that relevant crime types are reflected.

We used AIC as it is an unbiased measure of accuracy in comparison to RMSE as RMSE tends to depend on degree of freedom, variance etc. while AIC is does not. AIC has tendency to vary a lot and generally when used with test data the AIC value tends to increase as we also observed in our data. However, we used those variables whose overall AIC value was the least.

References

Cherone, H. (2017, April 17). Here's How Many Officers Are Patrolling Your Neighborhood. Retrieved from https://www.dnainfo.com/chicago/20170417/logan-square/heres-how-many-officers-are-patrolling-your-neighborhood-watchdog

Multinomial Logistic Regression using SPSS Statistics. (n.d.). Retrieved from https://statistics.laerd.com/spss-tutorials/multinomial-logistic-regression-using-spss-statistics.php

Multinomial Logistic Regression | R DATA ANALYSIS EXAMPLES. (n.d.). Retrieved from https://stats.idre.ucla.edu/r/dae/multinomial-logistic-regression/

Data Sources

Chicago Community Area Data. (n.d.). Retrieved from http://www.robparal.com/ChicagoCommunityAreaData.html

Crimes - 2001 to present. (n.d.). Retrieved from https://catalog.data.gov/dataset/crimes-2001-to-present-398a4

Weather History for KORD. (n.d.). Retrieved from https://www.wunderground.com/history/airport/KORD/2009/1/1/DailyHistory.html?req_city=Chicago&req_state=IL&req_statename=&reqdb.zip=60666&reqdb.magic=4&reqdb.wmo=99999

Appendix

After Deliverable one we went ahead with working on decision trees as a model for our dataset. We also tested out other models as well for comparison to identify the best fit. We tried C5.0, rpart, and random forests, but observed that we were getting very high error rates. We discussed with professor Kitty K. Chan and Landon about our observations and they recommended that we try out other independent variables for better results.

But our dataset did not have any other independent variables to be used in our models. We had to search other viable datasets that could be merged with our original dataset to get these independent variables. We searched online and finally found a dataset with demographic information on percentage of education, income per capita, percentage of foreigner, percentage of poverty etc., from a consensus website: *American Community services* (ACS).

In our initial analysis with this dataset we did not consider the overlap of the years and had to re-evaluate our usage of this data source. We searched online to find the source for the additional dataset we used to try to segregate the data by each year. We called ACS to identify the data source but were not able to find any leads. We then searched their online data repository though their data query engines to reverse engineer the data sources by year and each independent variable. We searched and found the data. We then pulled in secondary data sources to get important attributes like Community Area and Year for our dataset. However, after this initial exercise, we saw that not many data points were there to cover all Community Areas of Chicago (some had 7 community areas).

Despite our efforts of reengineering the data using data engineering concepts we were not able to get sufficient data points for our analysis. We then followed another approach to segregate the overlap of years. We selected the year range, used data manipulation techniques to average the

data and segregated it into a year wise granularity, removed the first and last year (they were counted only once) and obtained the final dataset that could be merged with our initial crime dataset.

The Model That We Used in The Report

Fetch the data set

```
crime<-read.csv("/Users/Jung-yerin/Desktop/R STUDIO/crime_1207.csv")

str(crime)

## 'data.frame':    877091 obs. of  17 variables:
##  $ X.1           : int  3171 3172 3173 3174 3175 3176 3177 3178 317
9 3180 ...
##  $ X             : int  3171 3172 3173 3174 3175 3176 3177 3178 317
9 3180 ...
##  $ Community_Area: int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Year          : int  2009 2009 2009 2009 2009 2009 2009 2009 200
9 2009 ...
##  $ Date          : Factor w/ 426821 levels "01/01/2009 01:00:00 AM"
,..: 256889 186302 253135 3437 50460 324200 337916 289774 212311 37978
6 ...
##  $ Primary_Type  : Factor w/ 3 levels "BATTERY","NARCOTICS",..: 3 1
3 2 3 3 1 1 3 2 ...
##  $ Arrest        : Factor w/ 2 levels "false","true": 1 1 2 2 1 1 2
1 1 2 ...
##  $ Domestic      : Factor w/ 2 levels "false","true": 1 2 1 1 1 1 2
1 1 1 ...
##  $ Sides         : Factor w/ 9 levels "Central","Far_North_Side",..
: 2 2 2 2 2 2 2 2 2 ...
##  $ Edu_Rate      : num  0.41 0.41 0.41 0.41 0.41 0.41 0.41 0.41 0.4
1 0.41 ...
##  $ Foreign_Rate  : num  0.29 0.29 0.29 0.29 0.29 0.29 0.29 0.29 0.2
9 0.29 ...
##  $ Income        : num  40265 40265 40265 40265 40265 ...
##  $ Pov_Rate      : num  0.27 0.27 0.27 0.27 0.27 0.27 0.27 0.27 0.2
7 0.27 ...
##  $ Is_Theft      : int  1 0 1 0 1 1 0 0 1 0 ...
```

```
##  $ Is_Narcotics  : int  0 0 0 1 0 0 0 0 0 1 ...
##  $ Is_Battery    : int  0 1 0 0 0 0 1 1 0 0 ...
##  $ Month         : int  8 6 8 1 2 9 10 8 6 11 ...
```

1. Among 33 kinds of *Primary_Type* in the original cleaned dataset from Deliverable 1, we decided to focus on *THEFT*, *NARCOTICS*, and *BATTERY* for Deliverable 2 as they are amongst the top 5 crimes in Chicago.
2. To make prediction models, we made three other columns *Is_Theft*, *Is_Narcotics*, and *Is_Battery*. Each observation has either 1 or 0. If the crime happened, the value is *1* and if not, the value is *0*.
3. We merged new data set from $AmericanCommunitySurvey$, by adjusting the format to use *"Community_Area"* as key and each *"Year""* and *"Pov_Rate"* as $value$.
4. Description for new variables, such as *Edu_Rate*, *Foreign_Rate*, *Income*, and *Pov_Rate* are as follows:
   1) *Pov_Rate*: Percent income below poverty level
   2) *Edu_Rate*: Percent with a BA or Higher
   3) *Income*: Median household income
   4) *Foreign_Rate*: Percent foreign born

Bring required library

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
library(ggraph)
library(pROC)
```

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

```
##
## Attaching package: 'pROC'
```

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

```r
library(caret)
```

```
## Loading required package: lattice
```

```r
library(ROCR)
```

```
## Loading required package: gplots
```

```
##
## Attaching package: 'gplots'
```

```
## The following object is masked from 'package:stats':
##
##     lowess
```

```r
library(nnet)
library(leaps)
library(C50)
library(partykit)
```

```
## Loading required package: grid
```

```r
library(randomForest)
```

```
## randomForest 4.6-12
```

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

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```r
library(randomForestSRC)
```

```
##
##  randomForestSRC 2.5.1
##
##  Type rfsrc.news() to see new features, changes, and bug fixes.
##
```

```r
library(rpart)
library(rpart.plot)
library(ggpubr)
```

```
## Loading required package: magrittr
```

We decided to use $ Multinomial Logistic Regression Model$, to use categorical variable - *Primary_Type* as a dependent variable.

<Sampling>

```
set.seed(1993)
indA <- sample(which(crime$Primary_Type=="THEFT"), 11000, replace = F)
indB <- sample(which(crime$Primary_Type=="NARCOTICS"), 11000, replace
= F)
indC <- sample(which(crime$Primary_Type=="BATTERY"), 11000, replace =
F)
sample_A <- crime[indA,]
sample_B <- crime[indB,]
sample_C <- crime[indC,]
ABC <- rbind(sample_A, sample_B, sample_C)

str(ABC)

## 'data.frame':    33000 obs. of  17 variables:
##  $ X.1           : int  441996 30230 517255 977707 716887 731653 36
6673 369339 1225089 14356 ...
##  $ X             : int  441996 30230 517255 977707 716887 731653 36
6673 369339 1225089 14356 ...
##  $ Community_Area: int  28 12 3 61 43 43 26 26 71 1 ...
##  $ Year          : int  2012 2009 2012 2009 2009 2012 2010 2010 201
1 2013 ...
##  $ Date          : Factor w/ 426821 levels "01/01/2009 01:00:00 AM"
,..: 365952 208562 357876 230956 238549 401059 372277 382308 222644 25
5247 ...
##  $ Primary_Type  : Factor w/ 3 levels "BATTERY","NARCOTICS",..: 3 3
3 3 3 3 3 3 3 3 ...
##  $ Arrest        : Factor w/ 2 levels "false","true": 1 1 2 2 1 1 1
1 1 1 ...
##  $ Domestic      : Factor w/ 2 levels "false","true": 1 1 1 1 1 1 1
1 1 1 ...
##  $ Sides         : Factor w/ 9 levels "Central","Far_North_Side",..
: 9 2 2 8 7 7 9 9 4 2 ...
##  $ Edu_Rate      : num  0.64 0.58 0.54 0.08 0.23 0.22 0.06 0.06 0.1
3 0.42 ...
##  $ Foreign_Rate  : num  0.03 0.18 0.24 0.4 0.01 0.01 0.01 0.01 0.03
0.29 ...
##  $ Income        : num  69676 92419 44980 33365 30049 ...
##  $ Pov_Rate      : num  0.25 0.06 0.25 0.34 0.32 0.34 0.46 0.46 0.3
```

```
0.27 ...
##  $ Is_Theft      : int  1 1 1 1 1 1 1 1 1 ...
##  $ Is_Narcotics  : int  0 0 0 0 0 0 0 0 0 ...
##  $ Is_Battery    : int  0 0 0 0 0 0 0 0 0 ...
##  $ Month         : int  11 6 10 7 7 12 11 11 7 8 ...

crime_refined<-crime[-c(indA,indB,indC),]
str(crime_refined)

## 'data.frame':    844091 obs. of  17 variables:
##  $ X.1           : int  3171 3172 3173 3174 3175 3176 3177 3178 317
9 3180 ...
##  $ X             : int  3171 3172 3173 3174 3175 3176 3177 3178 317
9 3180 ...
##  $ Community_Area: int  1 1 1 1 1 1 1 1 1 ...
##  $ Year          : int  2009 2009 2009 2009 2009 2009 2009 2009 200
9 2009 ...
##  $ Date          : Factor w/ 426821 levels "01/01/2009 01:00:00 AM"
,..: 256889 186302 253135 3437 50460 324200 337916 289774 212311 37978
6 ...
##  $ Primary_Type  : Factor w/ 3 levels "BATTERY","NARCOTICS",..: 3 1
3 2 3 3 1 1 3 2 ...
##  $ Arrest        : Factor w/ 2 levels "false","true": 1 1 2 2 1 1 2
1 1 2 ...
##  $ Domestic      : Factor w/ 2 levels "false","true": 1 2 1 1 1 1 2
1 1 1 ...
##  $ Sides         : Factor w/ 9 levels "Central","Far_North_Side",..
: 2 2 2 2 2 2 2 2 2 ...
##  $ Edu_Rate      : num  0.41 0.41 0.41 0.41 0.41 0.41 0.41 0.41 0.4
1 0.41 ...
##  $ Foreign_Rate  : num  0.29 0.29 0.29 0.29 0.29 0.29 0.29 0.29 0.2
9 0.29 ...
##  $ Income        : num  40265 40265 40265 40265 40265 ...
##  $ Pov_Rate      : num  0.27 0.27 0.27 0.27 0.27 0.27 0.27 0.27 0.2
7 0.27 ...
##  $ Is_Theft      : int  1 0 1 0 1 1 0 0 1 0 ...
##  $ Is_Narcotics  : int  0 0 0 1 0 0 0 0 0 1 ...
##  $ Is_Battery    : int  0 1 0 0 0 0 1 1 0 0 ...
##  $ Month         : int  8 6 8 1 2 9 10 8 6 11 ...

indT<- sample(1:nrow(crime_refined),10000)
sample_test <- crime_refined[indT,]
```

1. Train dataset <*ABC*>: As our data set is imbalanced, having more *THEFT* than *NARCOTICS* or *BATTERY*, we decided to conditionally sample from the original data set, 11,000 sample from each of the three types of crimes.

2. Test dataset <*sample_test*>: As we removed train dataset from the original dataset, we simply sampled 10,000 cases from the original dataset.

<Variable selection method>

To capture independent variables that would be most useful for our prediction model, we used $VariableSelectionMethod$.

```
null=multinom(Primary_Type ~ 1, data=ABC)

## # weights:  6 (2 variable)
## initial  value 36254.205526
## final  value 36254.205526
## converged

full=multinom(Primary_Type ~
                Community_Area+Domestic+Arrest+Edu_Rate+
                Foreign_Rate+Income+Pov_Rate+Month, data=ABC)

## # weights:  30 (18 variable)
## initial  value 36254.205526
## iter  10 value 29005.845945
## iter  20 value 18756.374710
## iter  30 value 17285.562211
## final  value 17274.314956
## converged

step(null, scope=list(lower=null, upper=full), direction="forward")

## Start:  AIC=72512.41
## Primary_Type ~ 1
##
## trying + Community_Area
## # weights:  9 (4 variable)
## initial  value 36254.205526
## final  value 35953.002826
## converged
## trying + Domestic
## # weights:  9 (4 variable)
## initial  value 36254.205526
## iter  10 value 30368.404041
## iter  20 value 30288.840045
## final  value 30288.726285
## converged
## trying + Arrest
## # weights:  9 (4 variable)
```

```
## initial  value 36254.205526
## iter  10 value 23620.216704
## final  value 23603.417109
## converged
## trying + Edu_Rate
## # weights:  9 (4 variable)
## initial  value 36254.205526
## final  value 34220.138217
## converged
## trying + Foreign_Rate
## # weights:  9 (4 variable)
## initial  value 36254.205526
## final  value 36153.433272
## converged
## trying + Income
## # weights:  9 (4 variable)
## initial  value 36254.205526
## final  value 34112.528403
## converged
## trying + Pov_Rate
## # weights:  9 (4 variable)
## initial  value 36254.205526
## final  value 34526.688787
## converged
## trying + Month
## # weights:  9 (4 variable)
## initial  value 36254.205526
## final  value 36212.418335
## converged
##                   Df      AIC
## + +Arrest          4 47214.83
## + +Domestic        4 60585.45
## + +Income          4 68233.06
## + +Edu_Rate        4 68448.28
## + +Pov_Rate        4 69061.38
## + +Community_Area  4 71914.01
## + +Foreign_Rate    4 72314.87
## + +Month           4 72432.84
## <none>            2 72512.41
## # weights:  9 (4 variable)
## initial  value 36254.205526
## iter  10 value 23620.216704
## final  value 23603.417109
## converged
##
## Step:  AIC=47214.83
```

```
## Primary_Type ~ Arrest
##
## trying + Community_Area
## # weights:  12 (6 variable)
## initial  value 36254.205526
## iter  10 value 23480.898122
## iter  20 value 23327.219399
## final  value 23327.208592
## converged
## trying + Domestic
## # weights:  12 (6 variable)
## initial  value 36254.205526
## iter  10 value 18479.394424
## iter  20 value 18379.502344
## final  value 18379.470348
## converged
## trying + Edu_Rate
## # weights:  12 (6 variable)
## initial  value 36254.205526
## iter  10 value 22286.090325
## final  value 22262.403767
## converged
## trying + Foreign_Rate
## # weights:  12 (6 variable)
## initial  value 36254.205526
## iter  10 value 23598.323909
## iter  20 value 23536.717007
## final  value 23536.656167
## converged
## trying + Income
## # weights:  12 (6 variable)
## initial  value 36254.205526
## iter  10 value 22347.623651
## iter  20 value 22188.247546
## final  value 22188.183187
## converged
## trying + Pov_Rate
## # weights:  12 (6 variable)
## initial  value 36254.205526
## iter  10 value 22600.320991
## iter  20 value 22505.821599
## final  value 22505.738650
## converged
## trying + Month
## # weights:  12 (6 variable)
## initial  value 36254.205526
```

```
## iter  10 value 23781.533046
## iter  20 value 23585.081810
## final  value 23584.983777
## converged
##                    Df      AIC
## + +Domestic         6 36770.94
## + +Income           6 44388.37
## + +Edu_Rate         6 44536.81
## + +Pov_Rate         6 45023.48
## + +Community_Area   6 46666.42
## + +Foreign_Rate     6 47085.31
## + +Month            6 47181.97
## <none>             4 47214.83
## # weights:  12 (6 variable)
## initial  value 36254.205526
## iter  10 value 18479.394424
## iter  20 value 18379.502344
## final  value 18379.470348
## converged
##
## Step:  AIC=36770.94
## Primary_Type ~ Arrest + Domestic
##
## trying + Community_Area
## # weights:  15 (8 variable)
## initial  value 36254.205526
## iter  10 value 19705.383084
## iter  20 value 18234.871732
## final  value 18232.864262
## converged
## trying + Edu_Rate
## # weights:  15 (8 variable)
## initial  value 36254.205526
## iter  10 value 17606.055932
## iter  20 value 17435.468838
## final  value 17435.416835
## converged
## trying + Foreign_Rate
## # weights:  15 (8 variable)
## initial  value 36254.205526
## iter  10 value 18667.111674
## iter  20 value 18336.341966
## final  value 18336.283076
## converged
## trying + Income
## # weights:  15 (8 variable)
```

```
## initial  value 36254.205526
## iter  10 value 17686.256071
## iter  20 value 17357.153031
## final  value 17355.818791
## converged
## trying + Pov_Rate
## # weights:  15 (8 variable)
## initial  value 36254.205526
## iter  10 value 17847.038390
## iter  20 value 17559.323043
## final  value 17558.632483
## converged
## trying + Month
## # weights:  15 (8 variable)
## initial  value 36254.205526
## iter  10 value 20537.253558
## iter  20 value 18364.916673
## iter  30 value 18360.004134
## iter  30 value 18360.004133
## iter  30 value 18360.004133
## final  value 18360.004133
## converged
##                    Df      AIC
## + +Income           8 34727.64
## + +Edu_Rate         8 34886.83
## + +Pov_Rate         8 35133.26
## + +Community_Area   8 36481.73
## + +Foreign_Rate     8 36688.57
## + +Month            8 36736.01
## <none>             6 36770.94
## # weights:  15 (8 variable)
## initial  value 36254.205526
## iter  10 value 17686.256071
## iter  20 value 17357.153031
## final  value 17355.818791
## converged
##
## Step:  AIC=34727.64
## Primary_Type ~ Arrest + Domestic + Income
##
## trying + Community_Area
## # weights:  18 (10 variable)
## initial  value 36254.205526
## iter  10 value 19510.731513
## iter  20 value 17356.737492
## final  value 17355.437681
```

```
## converged
## trying + Edu_Rate
## # weights:  18 (10 variable)
## initial  value 36254.205526
## iter  10 value 19351.524791
## iter  20 value 17328.643597
## iter  30 value 17327.058999
## final  value 17326.963767
## converged
## trying + Foreign_Rate
## # weights:  18 (10 variable)
## initial  value 36254.205526
## iter  10 value 19304.596414
## iter  20 value 17352.979298
## iter  30 value 17351.781452
## final  value 17351.403035
## converged
## trying + Pov_Rate
## # weights:  18 (10 variable)
## initial  value 36254.205526
## iter  10 value 19207.104494
## iter  20 value 17352.311999
## iter  30 value 17350.423858
## final  value 17350.356628
## converged
## trying + Month
## # weights:  18 (10 variable)
## initial  value 36254.205526
## iter  10 value 23862.613642
## iter  20 value 17426.365251
## iter  30 value 17343.879508
## final  value 17340.613152
## converged
##                    Df      AIC
## + +Edu_Rate        10 34673.93
## + +Month           10 34701.23
## + +Pov_Rate        10 34720.71
## + +Foreign_Rate    10 34722.81
## <none>              8 34727.64
## + +Community_Area 10 34730.88
## # weights:  18 (10 variable)
## initial  value 36254.205526
## iter  10 value 19351.524791
## iter  20 value 17328.643597
## iter  30 value 17327.058999
## final  value 17326.963767
```

```
## converged
##
## Step:  AIC=34673.93
## Primary_Type ~ Arrest + Domestic + Income + Edu_Rate
##
## trying + Community_Area
## # weights:  21 (12 variable)
## initial  value 36254.205526
## iter  10 value 19486.887287
## iter  20 value 17322.503712
## final  value 17319.144688
## converged
## trying + Foreign_Rate
## # weights:  21 (12 variable)
## initial  value 36254.205526
## iter  10 value 19336.209834
## iter  20 value 17321.059082
## iter  30 value 17318.818847
## iter  40 value 17318.670879
## iter  40 value 17318.670878
## iter  40 value 17318.670878
## final  value 17318.670878
## converged
## trying + Pov_Rate
## # weights:  21 (12 variable)
## initial  value 36254.205526
## iter  10 value 19203.042157
## iter  20 value 17311.345168
## iter  30 value 17307.688431
## iter  40 value 17307.436669
## final  value 17307.427142
## converged
## trying + Month
## # weights:  21 (12 variable)
## initial  value 36254.205526
## iter  10 value 23847.247956
## iter  20 value 17513.916034
## iter  30 value 17313.961146
## iter  40 value 17312.095362
## final  value 17312.095159
## converged
##                   Df      AIC
## + +Pov_Rate       12 34638.85
## + +Month          12 34648.19
## + +Foreign_Rate   12 34661.34
## + +Community_Area 12 34662.29
```

```
## <none>            10 34673.93
## # weights:  21 (12 variable)
## initial  value 36254.205526
## iter  10 value 19203.042157
## iter  20 value 17311.345168
## iter  30 value 17307.688431
## iter  40 value 17307.436669
## final  value 17307.427142
## converged
##
## Step:  AIC=34638.85
## Primary_Type ~ Arrest + Domestic + Income + Edu_Rate + Pov_Rate
##
## trying + Community_Area
## # weights:  24 (14 variable)
## initial  value 36254.205526
## iter  10 value 19457.244971
## iter  20 value 17353.943428
## iter  30 value 17298.898242
## final  value 17298.897163
## converged
## trying + Foreign_Rate
## # weights:  24 (14 variable)
## initial  value 36254.205526
## iter  10 value 19197.619392
## iter  20 value 17323.046553
## iter  30 value 17301.812798
## final  value 17301.810353
## converged
## trying + Month
## # weights:  24 (14 variable)
## initial  value 36254.205526
## iter  10 value 23847.940871
## iter  20 value 17605.429372
## iter  30 value 17292.578230
## iter  40 value 17292.201178
## final  value 17292.062970
## converged
##                   Df      AIC
## + +Month          14 34612.13
## + +Community_Area 14 34625.79
## + +Foreign_Rate   14 34631.62
## <none>            12 34638.85
## # weights:  24 (14 variable)
## initial  value 36254.205526
## iter  10 value 23847.940871
```

```
## iter  20 value 17605.429372
## iter  30 value 17292.578230
## iter  40 value 17292.201178
## final  value 17292.062970
## converged
##
## Step:  AIC=34612.13
## Primary_Type ~ Arrest + Domestic + Income + Edu_Rate + Pov_Rate +
##     Month
##
## trying + Community_Area
## # weights:  27 (16 variable)
## initial  value 36254.205526
## iter  10 value 29000.491311
## iter  20 value 17729.881395
## iter  30 value 17283.657478
## final  value 17283.553558
## converged
## trying + Foreign_Rate
## # weights:  27 (16 variable)
## initial  value 36254.205526
## iter  10 value 23577.113017
## iter  20 value 17829.178445
## iter  30 value 17287.730899
## final  value 17286.593891
## converged
##                   Df      AIC
## + +Community_Area 16 34599.11
## + +Foreign_Rate   16 34605.19
## <none>            14 34612.13
## # weights:  27 (16 variable)
## initial  value 36254.205526
## iter  10 value 29000.491311
## iter  20 value 17729.881395
## iter  30 value 17283.657478
## final  value 17283.553558
## converged
##
## Step:  AIC=34599.11
## Primary_Type ~ Arrest + Domestic + Income + Edu_Rate + Pov_Rate +
##     Month + Community_Area
##
## trying + Foreign_Rate
## # weights:  30 (18 variable)
## initial  value 36254.205526
## iter  10 value 29005.793043
```

```
## iter  20 value 18753.270834
## iter  30 value 17285.614477
## final  value 17274.314636
## converged
##                 Df      AIC
## + +Foreign_Rate 18 34584.63
## <none>          16 34599.11
## # weights:  30 (18 variable)
## initial  value 36254.205526
## iter  10 value 29005.793043
## iter  20 value 18753.270834
## iter  30 value 17285.614477
## final  value 17274.314636
## converged
##
## Step:  AIC=34584.63
## Primary_Type ~ Arrest + Domestic + Income + Edu_Rate + Pov_Rate +
##     Month + Community_Area + Foreign_Rate

## Call:
## multinom(formula = Primary_Type ~ Arrest + Domestic + Income +
##     Edu_Rate + Pov_Rate + Month + Community_Area + Foreign_Rate,
##     data = ABC)
##
## Coefficients:
##           (Intercept) Arresttrue Domestictrue        Income   Edu_R
ate
## NARCOTICS   -3.830636  6.5688214    -7.645141 -5.188136e-06 -1.7195
586
## THEFT        0.232151 -0.9652246    -3.426348  9.949687e-06  0.8010
404
##            Pov_Rate       Month Community_Area Foreign_Rate
## NARCOTICS  1.379389 -0.01570058  -0.0060393298  -0.75465690
## THEFT     -1.211378  0.02055568   0.0003903971  -0.08316825
##
## Residual Deviance: 34548.63
## AIC: 34584.63
```

As we can see in the result, *Primary_Type ~ Arrest + Domestic + Income + Edu_Rate + Pov_Rate + Month + Community_Area + Foreign_Rate* has the least *AIC*, which is *34584.63*

<Construct a model>

```
logr_pri <-multinom(Primary_Type ~ Arrest + Domestic +
                      Income + Edu_Rate + Pov_Rate + Month +
                      Community_Area + Foreign_Rate, data=ABC)

## # weights:  30 (18 variable)
## initial  value 36254.205526
## iter  10 value 29005.793043
## iter  20 value 18753.270834
## iter  30 value 17285.614477
## final  value 17274.314636
## converged

summary(logr_pri)

## Call:
## multinom(formula = Primary_Type ~ Arrest + Domestic + Income +
##     Edu_Rate + Pov_Rate + Month + Community_Area + Foreign_Rate,
##     data = ABC)
##
## Coefficients:
##           (Intercept) Arresttrue Domestictrue        Income   Edu_R
ate
## NARCOTICS   -3.830636  6.5688214    -7.645141 -5.188136e-06 -1.7195
586
## THEFT        0.232151 -0.9652246    -3.426348  9.949687e-06  0.8010
404
##            Pov_Rate       Month Community_Area Foreign_Rate
## NARCOTICS  1.379389 -0.01570058  -0.0060393298  -0.75465690
## THEFT     -1.211378  0.02055568   0.0003903971  -0.08316825
##
## Std. Errors:
##            (Intercept)   Arresttrue Domestictrue       Income     E
du_Rate
## NARCOTICS 8.102696e-06 7.632870e-06 5.730015e-08 7.516688e-07 2.184
533e-06
## THEFT     5.384062e-06 1.235754e-06 5.897649e-07 4.688727e-07 1.356
330e-06
##               Pov_Rate        Month Community_Area Foreign_Rate
## NARCOTICS 3.799411e-06 5.235321e-05   0.0008151639 4.275542e-07
## THEFT     2.562992e-06 3.551417e-05   0.0005344685 2.663520e-07
##
## Residual Deviance: 34548.63
## AIC: 34584.63

exp(coef(logr_pri))
```

```
##             (Intercept)  Arresttrue Domestictrue    Income  Edu_Rate
## NARCOTICS  0.02169582 712.5295668 0.0004783629 0.9999948 0.1791452
## THEFT       1.26131023   0.3808976 0.0325054439 1.0000099 2.2278575
##              Pov_Rate    Month Community_Area Foreign_Rate
## NARCOTICS 3.9724731 0.984422      0.9939789    0.4701719
## THEFT     0.2977866 1.020768      1.0003905    0.9201963
```

1) ***THEFT***

(1) The relative risk ratio for a one-unit increase in the variable *Edu_Rate* is 2.2278575 for *THEFT* vs. *BATTERY*

(2) The relative risk ratio for a one-unit increase in the variable *Pov_Rate* is 0.2977866 for *THEFT* vs. *BATTERY*

2) ***NARCOTICS***

(1) The relative risk ratio for a one-unit increase in the variable *Edu_Rate* is 0.1791452 for being *NARCOTICS* vs. *BATTERY*

(2) The relative risk ratio for a one-unit increase in the variable *Pov_Rate* is 3.9724731 for *NARCOTICS* vs. *BATTERY*

<Evaluate the model>

```
pre_pri=data.frame(Crime=sample_test$Primary_Type, Pred=predict(logr_p
ri,newdata=sample_test,type="class"))
pre_pri[1:10,]
```

```
##        Crime      Pred
## 1      THEFT     THEFT
## 2      THEFT     THEFT
## 3  NARCOTICS NARCOTICS
## 4      THEFT     THEFT
## 5    BATTERY     THEFT
## 6    BATTERY     THEFT
## 7      THEFT     THEFT
## 8  NARCOTICS NARCOTICS
## 9    BATTERY   BATTERY
## 10     THEFT     THEFT
```

```
confusionMatrix(data =pre_pri$Pred, reference = pre_pri$Crime)
```

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction  BATTERY NARCOTICS THEFT
##   BATTERY      1725         1   152
##   NARCOTICS     441      2042   397
##   THEFT        1431        26  3785
```

```
##
## Overall Statistics
##
##                Accuracy : 0.7552
##                  95% CI : (0.7466, 0.7636)
##     No Information Rate : 0.4334
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6209
##  Mcnemar's Test P-Value : < 2.2e-16
##
## Statistics by Class:
##
##                      Class: BATTERY Class: NARCOTICS Class: THEFT
## Sensitivity                  0.4796           0.9870       0.8733
## Specificity                  0.9761           0.8943       0.7429
## Pos Pred Value               0.9185           0.7090       0.7221
## Neg Pred Value               0.7695           0.9962       0.8846
## Prevalence                   0.3597           0.2069       0.4334
## Detection Rate               0.1725           0.2042       0.3785
## Detection Prevalence         0.1878           0.2880       0.5242
## Balanced Accuracy            0.7278           0.9406       0.8081
```

Additional information for the recommendations

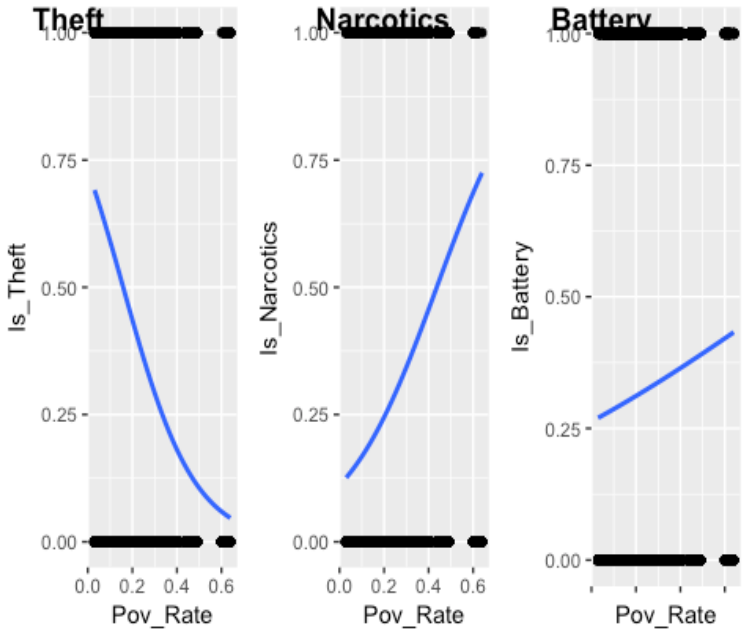
<1. The relationships between independent variables and the three types of crimes>

Plot the graphs

```
Pov_The_gg <- ggplot(ABC, aes(x=Pov_Rate, y=Is_Theft)) + geom_point()
+
  stat_smooth(method="glm", method.args=list(family="binomial"), se=FA
LSE)

Edu_The_gg <- ggplot(ABC, aes(x=Edu_Rate, y=Is_Theft)) + geom_point()
+
  stat_smooth(method="glm", method.args=list(family="binomial"), se=FA
LSE)

Pove_Nar_gg <- ggplot(ABC, aes(x=Pov_Rate, y=Is_Narcotics)) + geom_poi
nt() +
  stat_smooth(method="glm", method.args=list(family="binomial"), se=FA
LSE)

Edu_Nar_gg <- ggplot(ABC, aes(x=Edu_Rate, y=Is_Narcotics)) + geom_poin
t() +
  stat_smooth(method="glm", method.args=list(family="binomial"), se=FA
LSE)

Pove_Bat_gg <- ggplot(ABC, aes(x=Pov_Rate, y=Is_Battery)) + geom_point
() +
  stat_smooth(method="glm", method.args=list(family="binomial"), se=FA
LSE)

Edu_Bat_gg <- ggplot(ABC, aes(x=Edu_Rate, y=Is_Battery)) + geom_point(
) +
  stat_smooth(method="glm", method.args=list(family="binomial"), se=FA
LSE)
```

Plot in one place

```
ggarrange(Pov_The_gg, Pove_Nar_gg, Pove_Bat_gg + rremove("x.text"),
          labels = c("Theft", "Narcotics", "Battery"),
          ncol = 3, nrow = 1)
```



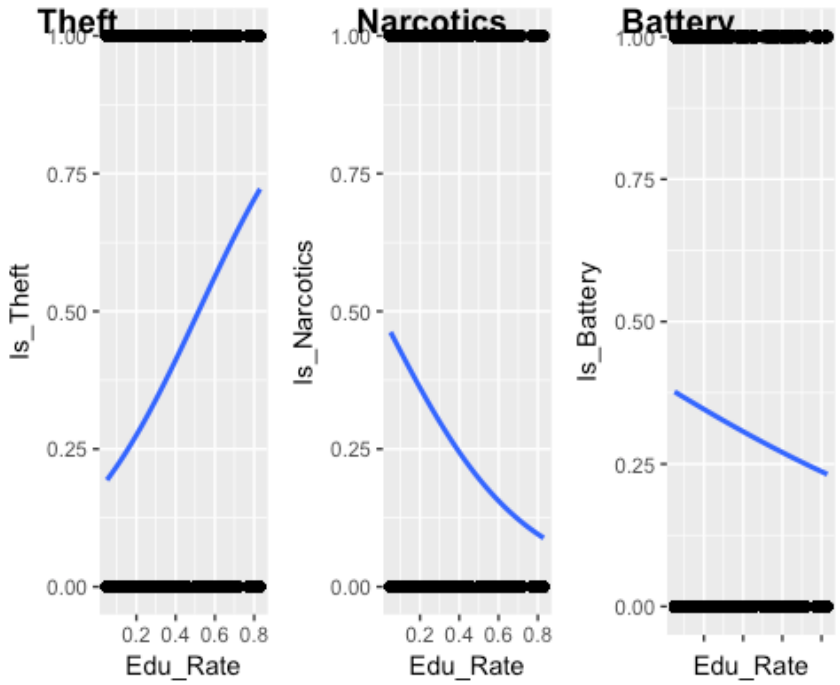
Based on the three types of crimes,

1) The community areas with lower poverty rates have higher theft incidents and vice versa.

2) The community areas with higher poverty rates have higher narcotics incidents and vice versa.

3) The community areas with higher poverty rates have higher battery incidents and vice versa.

```
ggarrange(Edu_The_gg, Edu_Nar_gg, Edu_Bat_gg + rremove("x.text"),
          labels = c("Theft", "Narcotics", "Battery"),
          ncol = 3, nrow = 1)
```



Based on the three types of crimes

1) The community areas with higher education rates have higher theft incidents and vice versa.

2) The community areas with higher education rates have lower narcotics incidents and vice versa.

3) The community areas with higher education rates have lower battery incidents and vice versa.

<2. To identify the top five *Sides* of the crimes>

Fetch the original poverty rate data

```
poverty<-read.csv("/Users/Jung-yerin/Desktop/R STUDIO/poverty_new.csv"
)
```

Arrange the data set to see the top five areas with highest *Pov_Rate* and lowest *Pov_Rate* as well.

```
poverty %>% arrange(desc(Average))
```

```
##    Community_Area X2009 X2010 X2011 X2012 X2013 Average
## 1              54  0.60  0.61  0.62  0.62  0.64   0.618
## 2              68  0.48  0.49  0.48  0.48  0.48   0.482
## 3              40  0.47  0.47  0.48  0.48  0.49   0.478
## 4              26  0.46  0.46  0.47  0.47  0.47   0.466
```

```
## 5              29  0.45  0.45  0.46  0.46  0.46   0.456
## 6              27  0.44  0.44  0.45  0.45  0.45   0.446
## 7              47  0.46  0.41  0.40  0.40  0.37   0.408
## 8              67  0.40  0.40  0.39  0.39  0.39   0.394
## 9              36  0.41  0.40  0.38  0.38  0.38   0.390
## 10             37  0.40  0.39  0.37  0.37  0.36   0.378
## 11             34  0.38  0.37  0.37  0.37  0.37   0.372
## 12             69  0.36  0.37  0.37  0.37  0.38   0.370
## 13             30  0.35  0.35  0.36  0.36  0.37   0.358
## 14             42  0.34  0.35  0.36  0.36  0.38   0.358
## 15             35  0.34  0.35  0.36  0.36  0.37   0.356
## 16             23  0.35  0.34  0.34  0.34  0.34   0.342
## 17             61  0.34  0.34  0.34  0.34  0.35   0.342
## 18             46  0.34  0.34  0.34  0.34  0.34   0.340
## 19             53  0.33  0.34  0.34  0.34  0.35   0.340
## 20             43  0.32  0.33  0.34  0.34  0.35   0.336
## 21             38  0.31  0.32  0.33  0.33  0.34   0.326
## 22             66  0.32  0.32  0.32  0.32  0.32   0.320
## 23             25  0.31  0.31  0.31  0.31  0.31   0.310
## 24             71  0.31  0.31  0.30  0.30  0.30   0.304
## 25             51  0.32  0.30  0.30  0.30  0.29   0.302
## 26             44  0.29  0.28  0.28  0.28  0.28   0.282
## 27             31  0.27  0.28  0.28  0.28  0.28   0.278
## 28              1  0.27  0.27  0.27  0.27  0.27   0.270
## 29             58  0.26  0.27  0.27  0.27  0.27   0.268
## 30             28  0.25  0.26  0.25  0.25  0.26   0.254
## 31              3  0.26  0.25  0.25  0.25  0.25   0.252
## 32             50  0.25  0.26  0.25  0.25  0.25   0.252
## 33             49  0.24  0.25  0.25  0.25  0.26   0.250
## 34             39  0.25  0.25  0.24  0.24  0.24   0.244
## 35             63  0.25  0.25  0.24  0.24  0.24   0.244
## 36             59  0.22  0.23  0.24  0.24  0.25   0.236
## 37             52  0.24  0.23  0.23  0.23  0.23   0.232
## 38             41  0.21  0.21  0.22  0.22  0.22   0.216
## 39             60  0.21  0.21  0.21  0.21  0.21   0.210
## 40             19  0.20  0.21  0.21  0.21  0.21   0.208
## 41             20  0.20  0.21  0.21  0.21  0.21   0.208
## 42             76  0.19  0.20  0.21  0.21  0.22   0.206
## 43             14  0.21  0.21  0.20  0.20  0.20   0.204
## 44              2  0.20  0.20  0.20  0.20  0.21   0.202
## 45             22  0.20  0.20  0.20  0.20  0.19   0.198
## 46             45  0.19  0.19  0.19  0.19  0.19   0.190
## 47             73  0.20  0.19  0.18  0.18  0.18   0.186
## 48             21  0.17  0.18  0.19  0.19  0.19   0.184
## 49             65  0.18  0.18  0.18  0.18  0.18   0.180
## 50             77  0.18  0.18  0.18  0.18  0.18   0.180
```

```
## 51              62  0.18  0.17  0.18  0.18  0.18   0.178
## 52              55  0.18  0.17  0.18  0.18  0.17   0.176
## 53              75  0.15  0.17  0.17  0.17  0.18   0.168
## 54              24  0.17  0.16  0.16  0.16  0.16   0.162
## 55              57  0.15  0.15  0.15  0.15  0.16   0.152
## 56              32  0.15  0.14  0.15  0.15  0.15   0.148
## 57              18  0.14  0.14  0.14  0.14  0.15   0.142
## 58              15  0.14  0.14  0.14  0.14  0.14   0.140
## 59              16  0.14  0.14  0.14  0.14  0.14   0.140
## 60              48  0.14  0.14  0.14  0.14  0.14   0.140
## 61               8  0.13  0.13  0.13  0.13  0.13   0.130
## 62               4  0.12  0.13  0.13  0.13  0.13   0.128
## 63              13  0.12  0.12  0.13  0.13  0.13   0.126
## 64              70  0.13  0.13  0.12  0.12  0.13   0.126
## 65               6  0.12  0.12  0.12  0.12  0.12   0.120
## 66               7  0.12  0.12  0.12  0.12  0.12   0.120
## 67              33  0.12  0.11  0.11  0.11  0.10   0.110
## 68              17  0.10  0.10  0.11  0.11  0.11   0.106
## 69              64  0.09  0.09  0.10  0.10  0.10   0.096
## 70              11  0.07  0.08  0.08  0.08  0.09   0.080
## 71              56  0.08  0.08  0.08  0.08  0.08   0.080
## 72               5  0.06  0.06  0.06  0.06  0.06   0.060
## 73              10  0.06  0.06  0.06  0.06  0.06   0.060
## 74              12  0.06  0.06  0.05  0.05  0.05   0.054
## 75              72  0.04  0.04  0.04  0.04  0.04   0.040
## 76              74  0.03  0.03  0.04  0.04  0.04   0.036
## 77               9  0.03  0.03  0.03  0.03  0.03   0.030
```

```
poverty %>% arrange(desc(X2013))
```

```
##    Community_Area X2009 X2010 X2011 X2012 X2013 Average
## 1              54  0.60  0.61  0.62  0.62  0.64   0.618
## 2              40  0.47  0.47  0.48  0.48  0.49   0.478
## 3              68  0.48  0.49  0.48  0.48  0.48   0.482
## 4              26  0.46  0.46  0.47  0.47  0.47   0.466
## 5              29  0.45  0.45  0.46  0.46  0.46   0.456
## 6              27  0.44  0.44  0.45  0.45  0.45   0.446
## 7              67  0.40  0.40  0.39  0.39  0.39   0.394
## 8              36  0.41  0.40  0.38  0.38  0.38   0.390
## 9              42  0.34  0.35  0.36  0.36  0.38   0.358
## 10             69  0.36  0.37  0.37  0.37  0.38   0.370
## 11             30  0.35  0.35  0.36  0.36  0.37   0.358
## 12             34  0.38  0.37  0.37  0.37  0.37   0.372
## 13             35  0.34  0.35  0.36  0.36  0.37   0.356
## 14             47  0.46  0.41  0.40  0.40  0.37   0.408
## 15             37  0.40  0.39  0.37  0.37  0.36   0.378
```

```
## 16             43  0.32  0.33  0.34  0.34  0.35   0.336
## 17             53  0.33  0.34  0.34  0.34  0.35   0.340
## 18             61  0.34  0.34  0.34  0.34  0.35   0.342
## 19             23  0.35  0.34  0.34  0.34  0.34   0.342
## 20             38  0.31  0.32  0.33  0.33  0.34   0.326
## 21             46  0.34  0.34  0.34  0.34  0.34   0.340
## 22             66  0.32  0.32  0.32  0.32  0.32   0.320
## 23             25  0.31  0.31  0.31  0.31  0.31   0.310
## 24             71  0.31  0.31  0.30  0.30  0.30   0.304
## 25             51  0.32  0.30  0.30  0.30  0.29   0.302
## 26             31  0.27  0.28  0.28  0.28  0.28   0.278
## 27             44  0.29  0.28  0.28  0.28  0.28   0.282
## 28              1  0.27  0.27  0.27  0.27  0.27   0.270
## 29             58  0.26  0.27  0.27  0.27  0.27   0.268
## 30             28  0.25  0.26  0.25  0.25  0.26   0.254
## 31             49  0.24  0.25  0.25  0.25  0.26   0.250
## 32              3  0.26  0.25  0.25  0.25  0.25   0.252
## 33             50  0.25  0.26  0.25  0.25  0.25   0.252
## 34             59  0.22  0.23  0.24  0.24  0.25   0.236
## 35             39  0.25  0.25  0.24  0.24  0.24   0.244
## 36             63  0.25  0.25  0.24  0.24  0.24   0.244
## 37             52  0.24  0.23  0.23  0.23  0.23   0.232
## 38             41  0.21  0.21  0.22  0.22  0.22   0.216
## 39             76  0.19  0.20  0.21  0.21  0.22   0.206
## 40              2  0.20  0.20  0.20  0.20  0.21   0.202
## 41             19  0.20  0.21  0.21  0.21  0.21   0.208
## 42             20  0.20  0.21  0.21  0.21  0.21   0.208
## 43             60  0.21  0.21  0.21  0.21  0.21   0.210
## 44             14  0.21  0.21  0.20  0.20  0.20   0.204
## 45             21  0.17  0.18  0.19  0.19  0.19   0.184
## 46             22  0.20  0.20  0.20  0.20  0.19   0.198
## 47             45  0.19  0.19  0.19  0.19  0.19   0.190
## 48             62  0.18  0.17  0.18  0.18  0.18   0.178
## 49             65  0.18  0.18  0.18  0.18  0.18   0.180
## 50             73  0.20  0.19  0.18  0.18  0.18   0.186
## 51             75  0.15  0.17  0.17  0.17  0.18   0.168
## 52             77  0.18  0.18  0.18  0.18  0.18   0.180
## 53             55  0.18  0.17  0.18  0.18  0.17   0.176
## 54             24  0.17  0.16  0.16  0.16  0.16   0.162
## 55             57  0.15  0.15  0.15  0.15  0.16   0.152
## 56             18  0.14  0.14  0.14  0.14  0.15   0.142
## 57             32  0.15  0.14  0.15  0.15  0.15   0.148
## 58             15  0.14  0.14  0.14  0.14  0.14   0.140
## 59             16  0.14  0.14  0.14  0.14  0.14   0.140
## 60             48  0.14  0.14  0.14  0.14  0.14   0.140
## 61              4  0.12  0.13  0.13  0.13  0.13   0.128
```

```
## 62              8  0.13  0.13  0.13  0.13  0.13   0.130
## 63             13  0.12  0.12  0.13  0.13  0.13   0.126
## 64             70  0.13  0.13  0.12  0.12  0.13   0.126
## 65              6  0.12  0.12  0.12  0.12  0.12   0.120
## 66              7  0.12  0.12  0.12  0.12  0.12   0.120
## 67             17  0.10  0.10  0.11  0.11  0.11   0.106
## 68             33  0.12  0.11  0.11  0.11  0.10   0.110
## 69             64  0.09  0.09  0.10  0.10  0.10   0.096
## 70             11  0.07  0.08  0.08  0.08  0.09   0.080
## 71             56  0.08  0.08  0.08  0.08  0.08   0.080
## 72              5  0.06  0.06  0.06  0.06  0.06   0.060
## 73             10  0.06  0.06  0.06  0.06  0.06   0.060
## 74             12  0.06  0.06  0.05  0.05  0.05   0.054
## 75             72  0.04  0.04  0.04  0.04  0.04   0.040
## 76             74  0.03  0.03  0.04  0.04  0.04   0.036
## 77              9  0.03  0.03  0.03  0.03  0.03   0.030

poverty %>% arrange(Average)

##    Community_Area X2009 X2010 X2011 X2012 X2013 Average
## 1               9  0.03  0.03  0.03  0.03  0.03   0.030
## 2              74  0.03  0.03  0.04  0.04  0.04   0.036
## 3              72  0.04  0.04  0.04  0.04  0.04   0.040
## 4              12  0.06  0.06  0.05  0.05  0.05   0.054
## 5               5  0.06  0.06  0.06  0.06  0.06   0.060
## 6              10  0.06  0.06  0.06  0.06  0.06   0.060
## 7              11  0.07  0.08  0.08  0.08  0.09   0.080
## 8              56  0.08  0.08  0.08  0.08  0.08   0.080
## 9              64  0.09  0.09  0.10  0.10  0.10   0.096
## 10             17  0.10  0.10  0.11  0.11  0.11   0.106
## 11             33  0.12  0.11  0.11  0.11  0.10   0.110
## 12              6  0.12  0.12  0.12  0.12  0.12   0.120
## 13              7  0.12  0.12  0.12  0.12  0.12   0.120
## 14             13  0.12  0.12  0.13  0.13  0.13   0.126
## 15             70  0.13  0.13  0.12  0.12  0.13   0.126
## 16              4  0.12  0.13  0.13  0.13  0.13   0.128
## 17              8  0.13  0.13  0.13  0.13  0.13   0.130
## 18             15  0.14  0.14  0.14  0.14  0.14   0.140
## 19             16  0.14  0.14  0.14  0.14  0.14   0.140
## 20             48  0.14  0.14  0.14  0.14  0.14   0.140
## 21             18  0.14  0.14  0.14  0.14  0.15   0.142
## 22             32  0.15  0.14  0.15  0.15  0.15   0.148
## 23             57  0.15  0.15  0.15  0.15  0.16   0.152
## 24             24  0.17  0.16  0.16  0.16  0.16   0.162
## 25             75  0.15  0.17  0.17  0.17  0.18   0.168
## 26             55  0.18  0.17  0.18  0.18  0.17   0.176
```

```
## 27              62  0.18  0.17  0.18  0.18  0.18   0.178
## 28              65  0.18  0.18  0.18  0.18  0.18   0.180
## 29              77  0.18  0.18  0.18  0.18  0.18   0.180
## 30              21  0.17  0.18  0.19  0.19  0.19   0.184
## 31              73  0.20  0.19  0.18  0.18  0.18   0.186
## 32              45  0.19  0.19  0.19  0.19  0.19   0.190
## 33              22  0.20  0.20  0.20  0.20  0.19   0.198
## 34               2  0.20  0.20  0.20  0.20  0.21   0.202
## 35              14  0.21  0.21  0.20  0.20  0.20   0.204
## 36              76  0.19  0.20  0.21  0.21  0.22   0.206
## 37              19  0.20  0.21  0.21  0.21  0.21   0.208
## 38              20  0.20  0.21  0.21  0.21  0.21   0.208
## 39              60  0.21  0.21  0.21  0.21  0.21   0.210
## 40              41  0.21  0.21  0.22  0.22  0.22   0.216
## 41              52  0.24  0.23  0.23  0.23  0.23   0.232
## 42              59  0.22  0.23  0.24  0.24  0.25   0.236
## 43              39  0.25  0.25  0.24  0.24  0.24   0.244
## 44              63  0.25  0.25  0.24  0.24  0.24   0.244
## 45              49  0.24  0.25  0.25  0.25  0.26   0.250
## 46               3  0.26  0.25  0.25  0.25  0.25   0.252
## 47              50  0.25  0.26  0.25  0.25  0.25   0.252
## 48              28  0.25  0.26  0.25  0.25  0.26   0.254
## 49              58  0.26  0.27  0.27  0.27  0.27   0.268
## 50               1  0.27  0.27  0.27  0.27  0.27   0.270
## 51              31  0.27  0.28  0.28  0.28  0.28   0.278
## 52              44  0.29  0.28  0.28  0.28  0.28   0.282
## 53              51  0.32  0.30  0.30  0.30  0.29   0.302
## 54              71  0.31  0.31  0.30  0.30  0.30   0.304
## 55              25  0.31  0.31  0.31  0.31  0.31   0.310
## 56              66  0.32  0.32  0.32  0.32  0.32   0.320
## 57              38  0.31  0.32  0.33  0.33  0.34   0.326
## 58              43  0.32  0.33  0.34  0.34  0.35   0.336
## 59              46  0.34  0.34  0.34  0.34  0.34   0.340
## 60              53  0.33  0.34  0.34  0.34  0.35   0.340
## 61              23  0.35  0.34  0.34  0.34  0.34   0.342
## 62              61  0.34  0.34  0.34  0.34  0.35   0.342
## 63              35  0.34  0.35  0.36  0.36  0.37   0.356
## 64              30  0.35  0.35  0.36  0.36  0.37   0.358
## 65              42  0.34  0.35  0.36  0.36  0.38   0.358
## 66              69  0.36  0.37  0.37  0.37  0.38   0.370
## 67              34  0.38  0.37  0.37  0.37  0.37   0.372
## 68              37  0.40  0.39  0.37  0.37  0.36   0.378
## 69              36  0.41  0.40  0.38  0.38  0.38   0.390
## 70              67  0.40  0.40  0.39  0.39  0.39   0.394
## 71              47  0.46  0.41  0.40  0.40  0.37   0.408
## 72              27  0.44  0.44  0.45  0.45  0.45   0.446
```

```
## 73              29  0.45  0.45  0.46  0.46  0.46   0.456
## 74              26  0.46  0.46  0.47  0.47  0.47   0.466
## 75              40  0.47  0.47  0.48  0.48  0.49   0.478
## 76              68  0.48  0.49  0.48  0.48  0.48   0.482
## 77              54  0.60  0.61  0.62  0.62  0.64   0.618
```

Fetch the original education rate data

```
education<-read.csv("/Users/Jung-yerin/Desktop/R STUDIO/education_new.
csv")
```

Arrange the data set to see the top five areas with highest *Edu_Rate* and lowest *Edu_Rate* as well.

```
education %>% arrange(desc(Average))
```

```
##    Community_Area X2009 X2010 X2011 X2012 X2013 Average
## 1               7  0.82  0.82  0.82  0.82  0.83   0.822
## 2              32  0.79  0.79  0.79  0.79  0.79   0.790
## 3               6  0.78  0.78  0.78  0.78  0.79   0.782
## 4               8  0.77  0.77  0.78  0.78  0.78   0.776
## 5              41  0.70  0.71  0.71  0.71  0.72   0.710
## 6              33  0.69  0.68  0.69  0.69  0.69   0.688
## 7               5  0.67  0.67  0.68  0.68  0.67   0.674
## 8              28  0.62  0.63  0.64  0.64  0.65   0.636
## 9              24  0.60  0.60  0.60  0.60  0.61   0.602
## 10             12  0.58  0.58  0.58  0.58  0.59   0.582
## 11              4  0.56  0.56  0.57  0.57  0.58   0.568
## 12             72  0.56  0.56  0.56  0.56  0.56   0.560
## 13             77  0.54  0.54  0.55  0.55  0.55   0.546
## 14              3  0.53  0.53  0.54  0.54  0.54   0.536
## 15             39  0.50  0.51  0.52  0.52  0.53   0.516
## 16             22  0.45  0.45  0.46  0.46  0.46   0.456
## 17             13  0.44  0.44  0.45  0.45  0.45   0.446
## 18              9  0.41  0.42  0.43  0.43  0.43   0.424
## 19              1  0.41  0.41  0.41  0.41  0.42   0.412
## 20             35  0.39  0.39  0.39  0.39  0.39   0.390
## 21              2  0.37  0.38  0.38  0.38  0.39   0.380
## 22             16  0.33  0.34  0.34  0.34  0.34   0.338
## 23             74  0.33  0.33  0.34  0.34  0.34   0.336
## 24             10  0.31  0.32  0.33  0.33  0.33   0.324
## 25             76  0.31  0.32  0.32  0.32  0.32   0.318
## 26             75  0.31  0.31  0.31  0.31  0.31   0.310
## 27             11  0.30  0.31  0.31  0.31  0.31   0.308
## 28             48  0.29  0.29  0.28  0.28  0.28   0.284
## 29             14  0.27  0.28  0.28  0.28  0.29   0.280
## 30             38  0.26  0.26  0.26  0.26  0.26   0.260
```

```
## 31              42  0.25  0.26  0.26  0.26  0.26   0.258
## 32              36  0.24  0.25  0.26  0.26  0.26   0.254
## 33              60  0.24  0.24  0.24  0.24  0.24   0.240
## 34              44  0.23  0.24  0.24  0.24  0.24   0.238
## 35              15  0.23  0.23  0.24  0.24  0.24   0.236
## 36              21  0.22  0.23  0.24  0.24  0.25   0.236
## 37              17  0.23  0.23  0.23  0.23  0.23   0.230
## 38              45  0.23  0.23  0.23  0.23  0.22   0.228
## 39              43  0.23  0.22  0.22  0.22  0.22   0.222
## 40              34  0.21  0.21  0.21  0.21  0.21   0.210
## 41              50  0.21  0.21  0.21  0.21  0.21   0.210
## 42              31  0.19  0.20  0.21  0.21  0.22   0.206
## 43              56  0.19  0.20  0.20  0.20  0.21   0.200
## 44              70  0.20  0.20  0.20  0.20  0.20   0.200
## 45              55  0.20  0.19  0.19  0.19  0.19   0.192
## 46              73  0.18  0.19  0.19  0.19  0.20   0.190
## 47              49  0.18  0.18  0.19  0.19  0.19   0.186
## 48              18  0.18  0.18  0.17  0.17  0.17   0.174
## 49              64  0.17  0.17  0.17  0.17  0.17   0.170
## 50              69  0.17  0.17  0.17  0.17  0.16   0.168
## 51              59  0.15  0.16  0.17  0.17  0.18   0.166
## 52              40  0.15  0.16  0.16  0.16  0.17   0.160
## 53              46  0.15  0.15  0.15  0.15  0.15   0.150
## 54              53  0.15  0.15  0.15  0.15  0.14   0.148
## 55              27  0.14  0.14  0.14  0.14  0.13   0.138
## 56              71  0.13  0.13  0.13  0.13  0.13   0.130
## 57              52  0.12  0.12  0.12  0.12  0.12   0.120
## 58              23  0.11  0.12  0.12  0.12  0.12   0.118
## 59              25  0.11  0.11  0.12  0.12  0.12   0.116
## 60              37  0.11  0.12  0.12  0.12  0.11   0.116
## 61              51  0.12  0.11  0.11  0.11  0.11   0.112
## 62              19  0.11  0.11  0.11  0.11  0.11   0.110
## 63              47  0.10  0.10  0.10  0.10  0.10   0.100
## 64              62  0.11  0.10  0.10  0.10  0.09   0.100
## 65              29  0.09  0.10  0.10  0.10  0.10   0.098
## 66              57  0.09  0.10  0.10  0.10  0.10   0.098
## 67              65  0.09  0.10  0.10  0.10  0.10   0.098
## 68              66  0.09  0.09  0.09  0.09  0.09   0.090
## 69              20  0.08  0.08  0.09  0.09  0.10   0.088
## 70              58  0.09  0.08  0.09  0.09  0.08   0.086
## 71              61  0.08  0.08  0.08  0.08  0.08   0.080
## 72              67  0.07  0.07  0.07  0.07  0.08   0.072
## 73              30  0.06  0.06  0.06  0.06  0.06   0.060
## 74              54  0.07  0.06  0.06  0.06  0.05   0.060
## 75              63  0.06  0.06  0.06  0.06  0.06   0.060
```

```
## 76              68  0.06  0.06  0.06  0.06  0.06   0.060
## 77              26  0.07  0.06  0.05  0.05  0.05   0.056
```

```
education %>% arrange(desc(X2013))
```

```
##    Community_Area X2009 X2010 X2011 X2012 X2013 Average
## 1               7  0.82  0.82  0.82  0.82  0.83   0.822
## 2               6  0.78  0.78  0.78  0.78  0.79   0.782
## 3              32  0.79  0.79  0.79  0.79  0.79   0.790
## 4               8  0.77  0.77  0.78  0.78  0.78   0.776
## 5              41  0.70  0.71  0.71  0.71  0.72   0.710
## 6              33  0.69  0.68  0.69  0.69  0.69   0.688
## 7               5  0.67  0.67  0.68  0.68  0.67   0.674
## 8              28  0.62  0.63  0.64  0.64  0.65   0.636
## 9              24  0.60  0.60  0.60  0.60  0.61   0.602
## 10             12  0.58  0.58  0.58  0.58  0.59   0.582
## 11              4  0.56  0.56  0.57  0.57  0.58   0.568
## 12             72  0.56  0.56  0.56  0.56  0.56   0.560
## 13             77  0.54  0.54  0.55  0.55  0.55   0.546
## 14              3  0.53  0.53  0.54  0.54  0.54   0.536
## 15             39  0.50  0.51  0.52  0.52  0.53   0.516
## 16             22  0.45  0.45  0.46  0.46  0.46   0.456
## 17             13  0.44  0.44  0.45  0.45  0.45   0.446
## 18              9  0.41  0.42  0.43  0.43  0.43   0.424
## 19              1  0.41  0.41  0.41  0.41  0.42   0.412
## 20              2  0.37  0.38  0.38  0.38  0.39   0.380
## 21             35  0.39  0.39  0.39  0.39  0.39   0.390
## 22             16  0.33  0.34  0.34  0.34  0.34   0.338
## 23             74  0.33  0.33  0.34  0.34  0.34   0.336
## 24             10  0.31  0.32  0.33  0.33  0.33   0.324
## 25             76  0.31  0.32  0.32  0.32  0.32   0.318
## 26             11  0.30  0.31  0.31  0.31  0.31   0.308
## 27             75  0.31  0.31  0.31  0.31  0.31   0.310
## 28             14  0.27  0.28  0.28  0.28  0.29   0.280
## 29             48  0.29  0.29  0.28  0.28  0.28   0.284
## 30             36  0.24  0.25  0.26  0.26  0.26   0.254
## 31             38  0.26  0.26  0.26  0.26  0.26   0.260
## 32             42  0.25  0.26  0.26  0.26  0.26   0.258
## 33             21  0.22  0.23  0.24  0.24  0.25   0.236
## 34             15  0.23  0.23  0.24  0.24  0.24   0.236
## 35             44  0.23  0.24  0.24  0.24  0.24   0.238
## 36             60  0.24  0.24  0.24  0.24  0.24   0.240
## 37             17  0.23  0.23  0.23  0.23  0.23   0.230
## 38             31  0.19  0.20  0.21  0.21  0.22   0.206
## 39             43  0.23  0.22  0.22  0.22  0.22   0.222
## 40             45  0.23  0.23  0.23  0.23  0.22   0.228
```

```
## 41              34  0.21  0.21  0.21  0.21  0.21   0.210
## 42              50  0.21  0.21  0.21  0.21  0.21   0.210
## 43              56  0.19  0.20  0.20  0.20  0.21   0.200
## 44              70  0.20  0.20  0.20  0.20  0.20   0.200
## 45              73  0.18  0.19  0.19  0.19  0.20   0.190
## 46              49  0.18  0.18  0.19  0.19  0.19   0.186
## 47              55  0.20  0.19  0.19  0.19  0.19   0.192
## 48              59  0.15  0.16  0.17  0.17  0.18   0.166
## 49              18  0.18  0.18  0.17  0.17  0.17   0.174
## 50              40  0.15  0.16  0.16  0.16  0.17   0.160
## 51              64  0.17  0.17  0.17  0.17  0.17   0.170
## 52              69  0.17  0.17  0.17  0.17  0.16   0.168
## 53              46  0.15  0.15  0.15  0.15  0.15   0.150
## 54              53  0.15  0.15  0.15  0.15  0.14   0.148
## 55              27  0.14  0.14  0.14  0.14  0.13   0.138
## 56              71  0.13  0.13  0.13  0.13  0.13   0.130
## 57              23  0.11  0.12  0.12  0.12  0.12   0.118
## 58              25  0.11  0.11  0.12  0.12  0.12   0.116
## 59              52  0.12  0.12  0.12  0.12  0.12   0.120
## 60              19  0.11  0.11  0.11  0.11  0.11   0.110
## 61              37  0.11  0.12  0.12  0.12  0.11   0.116
## 62              51  0.12  0.11  0.11  0.11  0.11   0.112
## 63              20  0.08  0.08  0.09  0.09  0.10   0.088
## 64              29  0.09  0.10  0.10  0.10  0.10   0.098
## 65              47  0.10  0.10  0.10  0.10  0.10   0.100
## 66              57  0.09  0.10  0.10  0.10  0.10   0.098
## 67              65  0.09  0.10  0.10  0.10  0.10   0.098
## 68              62  0.11  0.10  0.10  0.10  0.09   0.100
## 69              66  0.09  0.09  0.09  0.09  0.09   0.090
## 70              58  0.09  0.08  0.09  0.09  0.08   0.086
## 71              61  0.08  0.08  0.08  0.08  0.08   0.080
## 72              67  0.07  0.07  0.07  0.07  0.08   0.072
## 73              30  0.06  0.06  0.06  0.06  0.06   0.060
## 74              63  0.06  0.06  0.06  0.06  0.06   0.060
## 75              68  0.06  0.06  0.06  0.06  0.06   0.060
## 76              26  0.07  0.06  0.05  0.05  0.05   0.056
## 77              54  0.07  0.06  0.06  0.06  0.05   0.060
```

```
education %>% arrange(Average)
```

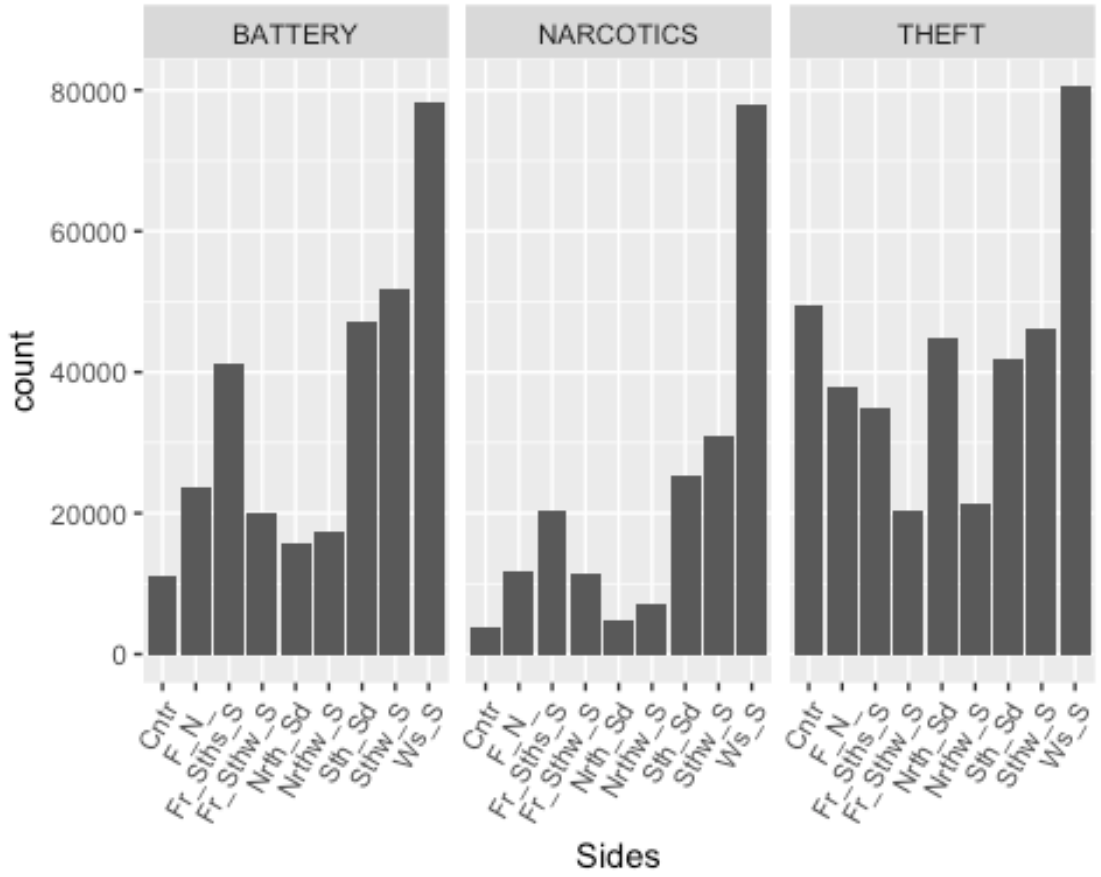
```
##    Community_Area X2009 X2010 X2011 X2012 X2013 Average
## 1              26  0.07  0.06  0.05  0.05  0.05   0.056
## 2              30  0.06  0.06  0.06  0.06  0.06   0.060
## 3              54  0.07  0.06  0.06  0.06  0.05   0.060
## 4              63  0.06  0.06  0.06  0.06  0.06   0.060
## 5              68  0.06  0.06  0.06  0.06  0.06   0.060
```

```
## 6              67  0.07  0.07  0.07  0.07  0.08   0.072
## 7              61  0.08  0.08  0.08  0.08  0.08   0.080
## 8              58  0.09  0.08  0.09  0.09  0.08   0.086
## 9              20  0.08  0.08  0.09  0.09  0.10   0.088
## 10             66  0.09  0.09  0.09  0.09  0.09   0.090
## 11             29  0.09  0.10  0.10  0.10  0.10   0.098
## 12             57  0.09  0.10  0.10  0.10  0.10   0.098
## 13             65  0.09  0.10  0.10  0.10  0.10   0.098
## 14             47  0.10  0.10  0.10  0.10  0.10   0.100
## 15             62  0.11  0.10  0.10  0.10  0.09   0.100
## 16             19  0.11  0.11  0.11  0.11  0.11   0.110
## 17             51  0.12  0.11  0.11  0.11  0.11   0.112
## 18             25  0.11  0.11  0.12  0.12  0.12   0.116
## 19             37  0.11  0.12  0.12  0.12  0.11   0.116
## 20             23  0.11  0.12  0.12  0.12  0.12   0.118
## 21             52  0.12  0.12  0.12  0.12  0.12   0.120
## 22             71  0.13  0.13  0.13  0.13  0.13   0.130
## 23             27  0.14  0.14  0.14  0.14  0.13   0.138
## 24             53  0.15  0.15  0.15  0.15  0.14   0.148
## 25             46  0.15  0.15  0.15  0.15  0.15   0.150
## 26             40  0.15  0.16  0.16  0.16  0.17   0.160
## 27             59  0.15  0.16  0.17  0.17  0.18   0.166
## 28             69  0.17  0.17  0.17  0.17  0.16   0.168
## 29             64  0.17  0.17  0.17  0.17  0.17   0.170
## 30             18  0.18  0.18  0.17  0.17  0.17   0.174
## 31             49  0.18  0.18  0.19  0.19  0.19   0.186
## 32             73  0.18  0.19  0.19  0.19  0.20   0.190
## 33             55  0.20  0.19  0.19  0.19  0.19   0.192
## 34             56  0.19  0.20  0.20  0.20  0.21   0.200
## 35             70  0.20  0.20  0.20  0.20  0.20   0.200
## 36             31  0.19  0.20  0.21  0.21  0.22   0.206
## 37             34  0.21  0.21  0.21  0.21  0.21   0.210
## 38             50  0.21  0.21  0.21  0.21  0.21   0.210
## 39             43  0.23  0.22  0.22  0.22  0.22   0.222
## 40             45  0.23  0.23  0.23  0.23  0.22   0.228
## 41             17  0.23  0.23  0.23  0.23  0.23   0.230
## 42             15  0.23  0.23  0.24  0.24  0.24   0.236
## 43             21  0.22  0.23  0.24  0.24  0.25   0.236
## 44             44  0.23  0.24  0.24  0.24  0.24   0.238
## 45             60  0.24  0.24  0.24  0.24  0.24   0.240
## 46             36  0.24  0.25  0.26  0.26  0.26   0.254
## 47             42  0.25  0.26  0.26  0.26  0.26   0.258
## 48             38  0.26  0.26  0.26  0.26  0.26   0.260
## 49             14  0.27  0.28  0.28  0.28  0.29   0.280
## 50             48  0.29  0.29  0.28  0.28  0.28   0.284
## 51             11  0.30  0.31  0.31  0.31  0.31   0.308
```

```
## 52              75  0.31  0.31  0.31  0.31  0.31   0.310
## 53              76  0.31  0.32  0.32  0.32  0.32   0.318
## 54              10  0.31  0.32  0.33  0.33  0.33   0.324
## 55              74  0.33  0.33  0.34  0.34  0.34   0.336
## 56              16  0.33  0.34  0.34  0.34  0.34   0.338
## 57               2  0.37  0.38  0.38  0.38  0.39   0.380
## 58              35  0.39  0.39  0.39  0.39  0.39   0.390
## 59               1  0.41  0.41  0.41  0.41  0.42   0.412
## 60               9  0.41  0.42  0.43  0.43  0.43   0.424
## 61              13  0.44  0.44  0.45  0.45  0.45   0.446
## 62              22  0.45  0.45  0.46  0.46  0.46   0.456
## 63              39  0.50  0.51  0.52  0.52  0.53   0.516
## 64               3  0.53  0.53  0.54  0.54  0.54   0.536
## 65              77  0.54  0.54  0.55  0.55  0.55   0.546
## 66              72  0.56  0.56  0.56  0.56  0.56   0.560
## 67               4  0.56  0.56  0.57  0.57  0.58   0.568
## 68              12  0.58  0.58  0.58  0.58  0.59   0.582
## 69              24  0.60  0.60  0.60  0.60  0.61   0.602
## 70              28  0.62  0.63  0.64  0.64  0.65   0.636
## 71               5  0.67  0.67  0.68  0.68  0.67   0.674
## 72              33  0.69  0.68  0.69  0.69  0.69   0.688
## 73              41  0.70  0.71  0.71  0.71  0.72   0.710
## 74               8  0.77  0.77  0.78  0.78  0.78   0.776
## 75               6  0.78  0.78  0.78  0.78  0.79   0.782
## 76              32  0.79  0.79  0.79  0.79  0.79   0.790
## 77               7  0.82  0.82  0.82  0.82  0.83   0.822
```

Plot the original crime data to see the top 5 *Sides* for *THEFT* and *NARCOTICS*

```
ggplot(data=crime, aes(x=Sides))+
  geom_bar(stat="count")+
  theme(axis.text.x = element_text(angle = 60, hjust = 1))+
  facet_grid(.~Primary_Type)+
  scale_x_discrete(label=abbreviate)
```



<3. The effect of temperature on theft occurences >

Imported a file which included the average temperature per month in Chicago from 2009 until 2013 from the wunderground weather website

```
Average_Temp <-read.csv("/Users/Jung-yerin/Desktop/R STUDIO/Average_Temp.csv")
```

Extracted the theft type of crime from our original crime file

```
crime_theft<- subset(crime, Primary_Type=="THEFT")
crime_theft <- crime_theft[complete.cases(crime_theft),]
head(crime_theft)
```

```
##     X.1    X Community_Area Year                  Date Primary_Typ
e
## 1  3171 3171              1 2009 08/04/2009 11:00:00 PM        THEF
T
## 3  3173 3173              1 2009 08/01/2009 08:00:00 PM        THEF
T
## 5  3175 3175              1 2009 02/15/2009 07:30:00 PM        THEF
T
## 6  3176 3176              1 2009 09/29/2009 06:00:00 PM        THEF
```

```
T
## 9  3179 3179              1 2009 06/29/2009 12:01:00 AM        THEF
T
## 15 3185 3185              1 2009 05/26/2009 08:00:00 AM        THEF
T
##    Arrest Domestic          Sides Edu_Rate Foreign_Rate   Income Po
v_Rate
## 1   false    false Far_North_Side     0.41         0.29 40265.14
0.27
## 3    true    false Far_North_Side     0.41         0.29 40265.14
0.27
## 5   false    false Far_North_Side     0.41         0.29 40265.14
0.27
## 6   false    false Far_North_Side     0.41         0.29 40265.14
0.27
## 9   false    false Far_North_Side     0.41         0.29 40265.14
0.27
## 15  false    false Far_North_Side     0.41         0.29 40265.14
0.27
##    Is_Theft Is_Narcotics Is_Battery Month
## 1         1            0          0     8
## 3         1            0          0     8
## 5         1            0          0     2
## 6         1            0          0     9
## 9         1            0          0     6
## 15        1            0          0     5
```

Converted *Primary_Type* into character, to count by *Month* and *Year*

```
crime_theft$Primary_Type <- as.character(crime_theft$Primary_Type)
```

Created a new data frame which showed how many occurences of theft happened in each month

```
unique(crime_theft$Month)
```

```
##  [1]  8  2  9  6  5  7 11 12 10  4  3  1
```

```
crime_theft <- transform(crime_theft, count = ave(Primary_Type, Month,
Year, FUN = length))
head(crime_theft)
```

```
##     X.1    X Community_Area Year                  Date Primary_Typ
e
## 1  3171 3171              1 2009 08/04/2009 11:00:00 PM        THEF
T
## 3  3173 3173              1 2009 08/01/2009 08:00:00 PM        THEF
T
```

```
## 5  3175 3175              1 2009 02/15/2009 07:30:00 PM        THEF
T
## 6  3176 3176              1 2009 09/29/2009 06:00:00 PM        THEF
T
## 9  3179 3179              1 2009 06/29/2009 12:01:00 AM        THEF
T
## 15 3185 3185              1 2009 05/26/2009 08:00:00 AM        THEF
T
##    Arrest Domestic          Sides Edu_Rate Foreign_Rate   Income Po
v_Rate
## 1   false    false Far_North_Side     0.41         0.29 40265.14
0.27
## 3    true    false Far_North_Side     0.41         0.29 40265.14
0.27
## 5   false    false Far_North_Side     0.41         0.29 40265.14
0.27
## 6   false    false Far_North_Side     0.41         0.29 40265.14
0.27
## 9   false    false Far_North_Side     0.41         0.29 40265.14
0.27
## 15  false    false Far_North_Side     0.41         0.29 40265.14
0.27
##    Is_Theft Is_Narcotics Is_Battery Month count
## 1         1            0          0     8  7591
## 3         1            0          0     8  7591
## 5         1            0          0     2  5397
## 6         1            0          0     9  7181
## 9         1            0          0     6  6674
## 15        1            0          0     5  6916
```

```
unique(crime_theft$count)
```

```
##  [1] 7591 5397 7181 6674 6916 7708 6431 5805 6874 6435 6397 5838 53
47 7477
## [15] 6125 6077 6970 7162 6876 6063 4793 6889 5961 6715 7401 6600 58
64 5993
## [29] 7349 6130 5409 4361 7016 6627 6663 5635 7254 7065 5933 6184 60
72 5859
## [43] 4883 6553 6460 5698 7019 6392 5553 6507 5745 5202 7100 6152 45
38 6159
## [57] 5328 5456 6266
## 59 Levels: 4361 4538 4793 4883 5202 5328 5347 5397 5409 5456 5553 .
.. 7708
```

```r
crime_theft_summary<- data.frame(crime_theft$Month,crime_theft$Year,cr
ime_theft$count)
head(crime_theft_summary)
```

```
##   crime_theft.Month crime_theft.Year crime_theft.count
## 1                 8             2009              7591
## 2                 8             2009              7591
## 3                 2             2009              5397
## 4                 9             2009              7181
## 5                 6             2009              6674
## 6                 5             2009              6916
```

```r
crime_theft_summary_final<- crime_theft_summary[!duplicated(crime_theft
t_summary),]
crime_theft_summary_final
```

```
##      crime_theft.Month crime_theft.Year crime_theft.count
## 1                    8             2009              7591
## 3                    2             2009              5397
## 4                    9             2009              7181
## 5                    6             2009              6674
## 6                    5             2009              6916
## 7                    7             2009              7708
## 9                   11             2009              6431
## 15                  12             2009              5805
## 17                  10             2009              6874
## 18                   4             2009              6435
## 23                   3             2009              6397
## 27                   1             2009              5838
## 1106                12             2010              5347
## 1107                 8             2010              7477
## 1108                 4             2010              6125
## 1109                 3             2010              6077
## 1110                 9             2010              6970
## 1111                 7             2010              7162
## 1122                 6             2010              6876
## 1126                 1             2010              6063
## 1134                 2             2010              4793
## 1136                10             2010              6889
## 1137                11             2010              5961
## 1142                 5             2010              6715
## 2272                 8             2011              7401
## 2274                 5             2011              6600
## 2275                 4             2011              5864
## 2277                12             2011              5993
## 2280                 7             2011              7349
```

```
## 2283                11            2011             6130
## 2284                 1            2011             5409
## 2286                 2            2011             4361
## 2288                 6            2011             7016
## 2289                 9            2011             6627
## 2291                10            2011             6663
## 2296                 3            2011             5635
## 3407                 7            2012             7254
## 3408                 6            2012             7065
## 3410                 3            2012             5933
## 3411                 4            2012             6184
## 3412                11            2012             6072
## 3414                12            2012             5859
## 3417                 2            2012             4883
## 3420                 9            2012             6553
## 3422                 5            2012             6460
## 3424                 1            2012             5698
## 3425                 8            2012             7019
## 3430                10            2012             6392
## 4453                 4            2013             5553
## 4454                 8            2013             7162
## 4455                 9            2013             6507
## 4456                11            2013             5745
## 4461                12            2013             5202
## 4464                 7            2013             7100
## 4465                 5            2013             6152
## 4468                 2            2013             4538
## 4469                10            2013             6159
## 4471                 3            2013             5328
## 4480                 1            2013             5456
## 4493                 6            2013             6266
```

```
colnames(crime_theft_summary_final)[colnames(crime_theft_summary_final
)== "crime_theft.Month"] <- "Month"
colnames(crime_theft_summary_final)[colnames(crime_theft_summary_final
)== "crime_theft.count"] <- "Count"
colnames(crime_theft_summary_final)[colnames(crime_theft_summary_final
)== "crime_theft.Year"] <- "Year"
```

Merged the weather data with our crime file

```
crime_theft_summary_final<- within(crime_theft_summary_final,MonthYear
<-paste(Month,Year,sep=""))
head(crime_theft_summary_final)
```

```
##   Month Year Count MonthYear
## 1     8 2009  7591     82009
```

```
## 3     2 2009  5397     22009
## 4     9 2009  7181     92009
## 5     6 2009  6674     62009
## 6     5 2009  6916     52009
## 7     7 2009  7708     72009
```

```
crime_theft_summary_final<- merge.data.frame(crime_theft_summary_final
, Average_Temp,by = "MonthYear")
head(crime_theft_summary_final)
```

```
##   MonthYear Month.x Year.x Count Month.y Year.y Average_Temperature
## 1    102009      10   2009  6874      10   2009                  49
## 2    102010      10   2010  6889      10   2010                  56
## 3    102011      10   2011  6663      10   2011                  55
## 4    102012      10   2012  6392      10   2012                  52
## 5    102013      10   2013  6159      10   2013                  53
## 6    112009      11   2009  6431      11   2009                  46
```

```
crime_theft_summary_final<-data.frame(crime_theft_summary_final$MonthY
ear,crime_theft_summary_final$Month.x,crime_theft_summary_final$Year.x
,crime_theft_summary_final$Count,crime_theft_summary_final$Average_Tem
perature)
head(crime_theft_summary_final)
```

```
##   crime_theft_summary_final.MonthYear crime_theft_summary_final.Mon
th.x
## 1                              102009
10
## 2                              102010
10
## 3                              102011
10
## 4                              102012
10
## 5                              102013
10
## 6                              112009
11
##   crime_theft_summary_final.Year.x crime_theft_summary_final.Count
## 1                             2009                             6874
## 2                             2010                             6889
## 3                             2011                             6663
## 4                             2012                             6392
## 5                             2013                             6159
## 6                             2009                             6431
##   crime_theft_summary_final.Average_Temperature
## 1                                            49
```

```
## 2                                                    56
## 3                                                    55
## 4                                                    52
## 5                                                    53
## 6                                                    46
```

Renamed the columns to be consistent

```
colnames(crime_theft_summary_final)[colnames(crime_theft_summary_final
)== "crime_theft_summary_final.MonthYear"] <- "MonthYear"
colnames(crime_theft_summary_final)[colnames(crime_theft_summary_final
)== "crime_theft_summary_final.Month.x"] <- "Month"
colnames(crime_theft_summary_final)[colnames(crime_theft_summary_final
)== "crime_theft_summary_final.Year.x"] <- "Year"
colnames(crime_theft_summary_final)[colnames(crime_theft_summary_final
)== "crime_theft_summary_final.Count"] <- "Count"
colnames(crime_theft_summary_final)[colnames(crime_theft_summary_final
)== "crime_theft_summary_final.Average_Temperature"] <- "Average_Tempe
rature"
head(crime_theft_summary_final)
```

```
##   MonthYear Month Year Count Average_Temperature
## 1    102009    10 2009  6874                  49
## 2    102010    10 2010  6889                  56
## 3    102011    10 2011  6663                  55
## 4    102012    10 2012  6392                  52
## 5    102013    10 2013  6159                  53
## 6    112009    11 2009  6431                  46
```

```
str(crime_theft_summary_final)
```

```
## 'data.frame':    60 obs. of  5 variables:
##  $ MonthYear          : Factor w/ 60 levels "102009","102010",..: 1
2 3 4 5 6 7 8 9 10 ...
##  $ Month              : int  10 10 10 10 10 11 11 11 11 11 ...
##  $ Year               : int  2009 2010 2011 2012 2013 2009 2010 201
1 2012 2013 ...
##  $ Count              : Factor w/ 59 levels "4361","4538",..: 43 45
40 31 28 33 20 26 23 14 ...
##  $ Average_Temperature: int  49 56 55 52 53 46 42 45 41 38 ...
```

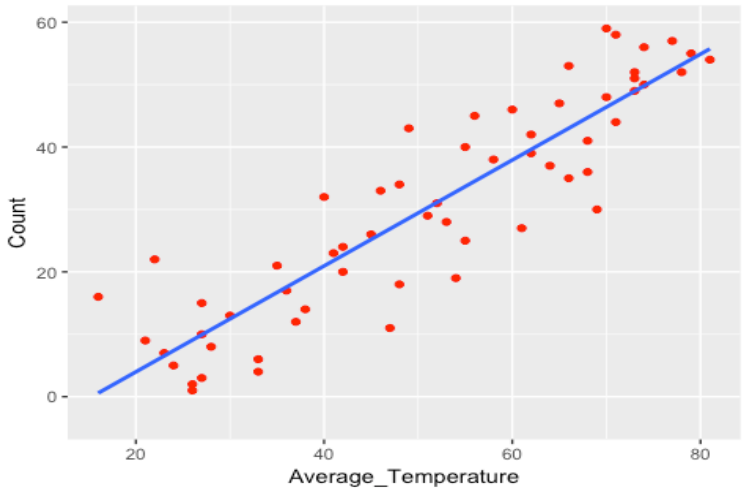
```
crime_theft_summary_final$Count<-as.integer(crime_theft_summary_final$
Count)
```

Ran a linear regression where y is equal to the monthly count of theft and x is equal to the monthly weather data

```r
crime_theft_summary_final<- lm(Count~Average_Temperature, data=crime_theft_summary_final)
summary(crime_theft_summary_final)
```

```
##
## Call:
## lm(formula = Count ~ Average_Temperature, data = crime_theft_summary_final)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.8875  -4.5476   0.1425   4.7008  16.3270
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -12.99578    2.92591  -4.442 4.07e-05 ***
## Average_Temperature   0.84858    0.05395  15.728  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.589 on 58 degrees of freedom
## Multiple R-squared:  0.8101, Adjusted R-squared:  0.8068
## F-statistic: 247.4 on 1 and 58 DF,  p-value: < 2.2e-16
```

```r
ggplot(data= crime_theft_summary_final, aes(x = Average_Temperature, y = Count)) +
  geom_point(colour = "red") +
  geom_smooth(method = "lm", fill = NA)
```



The Models That We Have Tried

(1) Logistic Regression Model

We decided to use categorical variables - *Is_Theft*, *Is_Narcotics*, and *Is_Battery* as the dependent variables for *LogisticRegressionModel*.

This model is using the same train and test sample data set. (Train dataset <*ABC*> and Test dataset <*sample_test*>)

1)THEFT

Variable Selection Method

```
null_A=glm(Is_Theft ~ 1, data=ABC, family=binomial)
full_A=glm(Is_Theft ~ Community_Area+Domestic+Arrest+
             Edu_Rate+Foreign_Rate+Income+Pov_Rate+Month,
           data=ABC, family=binomial)
step(null_A, scope=list(lower=null_A, upper=full_A), direction="forwar
d")
```

```
## Start:  AIC=42011.94
## Is_Theft ~ 1
##
##                  Df Deviance   AIC
## + Arrest          1    33640 33644
## + Income          1    38152 38156
## + Edu_Rate        1    38348 38352
## + Pov_Rate        1    38920 38924
## + Domestic        1    38940 38944
## + Community_Area  1    41422 41426
## + Foreign_Rate    1    41848 41852
## + Month           1    41951 41955
## <none>                 42010 42012
##
## Step:  AIC=33643.87
## Is_Theft ~ Arrest
##
##                  Df Deviance   AIC
## + Domestic        1    27714 27720
## + Income          1    31144 31150
## + Edu_Rate        1    31260 31266
## + Pov_Rate        1    31778 31784
```

```
## + Community_Area  1    33089 33095
## + Foreign_Rate    1    33540 33546
## + Month           1    33614 33620
## <none>                 33640 33644
##
## Step:  AIC=27719.86
## Is_Theft ~ Arrest + Domestic
##
##                  Df Deviance   AIC
## + Income          1    26155 26163
## + Edu_Rate        1    26288 26296
## + Pov_Rate        1    26491 26499
## + Community_Area  1    27442 27450
## + Foreign_Rate    1    27673 27681
## + Month           1    27682 27690
## <none>                 27714 27720
##
## Step:  AIC=26163.32
## Is_Theft ~ Arrest + Domestic + Income
##
##                  Df Deviance   AIC
## + Edu_Rate        1    26110 26120
## + Month           1    26128 26138
## + Pov_Rate        1    26152 26162
## <none>                 26155 26163
## + Foreign_Rate    1    26155 26165
## + Community_Area  1    26155 26165
##
## Step:  AIC=26120.43
## Is_Theft ~ Arrest + Domestic + Income + Edu_Rate
##
##                  Df Deviance   AIC
## + Month           1    26084 26096
## + Pov_Rate        1    26092 26104
## + Community_Area  1    26104 26116
## <none>                 26110 26120
## + Foreign_Rate    1    26109 26121
##
## Step:  AIC=26096.1
## Is_Theft ~ Arrest + Domestic + Income + Edu_Rate + Month
##
##                  Df Deviance   AIC
## + Pov_Rate        1    26066 26080
## + Community_Area  1    26078 26092
## <none>                 26084 26096
## + Foreign_Rate    1    26082 26096
```

```
##
## Step:  AIC=26079.66
## Is_Theft ~ Arrest + Domestic + Income + Edu_Rate + Month + Pov_Rate
##
##                  Df Deviance   AIC
## + Community_Area  1    26059 26075
## <none>                 26066 26080
## + Foreign_Rate    1    26066 26082
##
## Step:  AIC=26075
## Is_Theft ~ Arrest + Domestic + Income + Edu_Rate + Month + Pov_Rate
+
##     Community_Area
##
##                Df Deviance   AIC
## <none>              26059 26075
## + Foreign_Rate  1    26058 26076

##
## Call:  glm(formula = Is_Theft ~ Arrest + Domestic + Income + Edu_Ra
te +
##     Month + Pov_Rate + Community_Area, family = binomial, data = AB
C)
##
## Coefficients:
##    (Intercept)      Arresttrue    Domestictrue          Income
##     -1.191e-01      -2.945e+00      -3.316e+00       1.287e-05
##       Edu_Rate           Month        Pov_Rate  Community_Area
##      1.209e+00       2.427e-02      -1.275e+00       2.160e-03
##
## Degrees of Freedom: 32999 Total (i.e. Null);  32992 Residual
## Null Deviance:       42010
## Residual Deviance: 26060     AIC: 26070
```

As we can see in the result, *Is_Theft ~ Arrest + Domestic + Income + Edu_Rate + Month + Pov_Rate + Community_Area* has the least *AIC*, which is *26075*

Construct a model

```
logr_t <-glm(Is_Theft ~ Arrest + Domestic + Income +
               Edu_Rate + Month + Pov_Rate, data=ABC, family=binomial)
summary(logr_t)

##
## Call:
## glm(formula = Is_Theft ~ Arrest + Domestic + Income + Edu_Rate +
##     Month + Pov_Rate, family = binomial, data = ABC)
```

```
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1437  -0.4204  -0.3277   0.6469   3.5832
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -3.087e-02  1.583e-01  -0.195    0.845
## Arresttrue   -2.948e+00  3.625e-02 -81.334  < 2e-16 ***
## Domestictrue -3.313e+00  6.580e-02 -50.350  < 2e-16 ***
## Income        1.333e-05  2.500e-06   5.333 9.64e-08 ***
## Edu_Rate      1.084e+00  1.412e-01   7.677 1.63e-14 ***
## Month         2.434e-02  4.714e-03   5.163 2.43e-07 ***
## Pov_Rate     -1.247e+00  2.909e-01  -4.288 1.80e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 42010  on 32999  degrees of freedom
## Residual deviance: 26066  on 32993  degrees of freedom
## AIC: 26080
##
## Number of Fisher Scoring iterations: 6
```

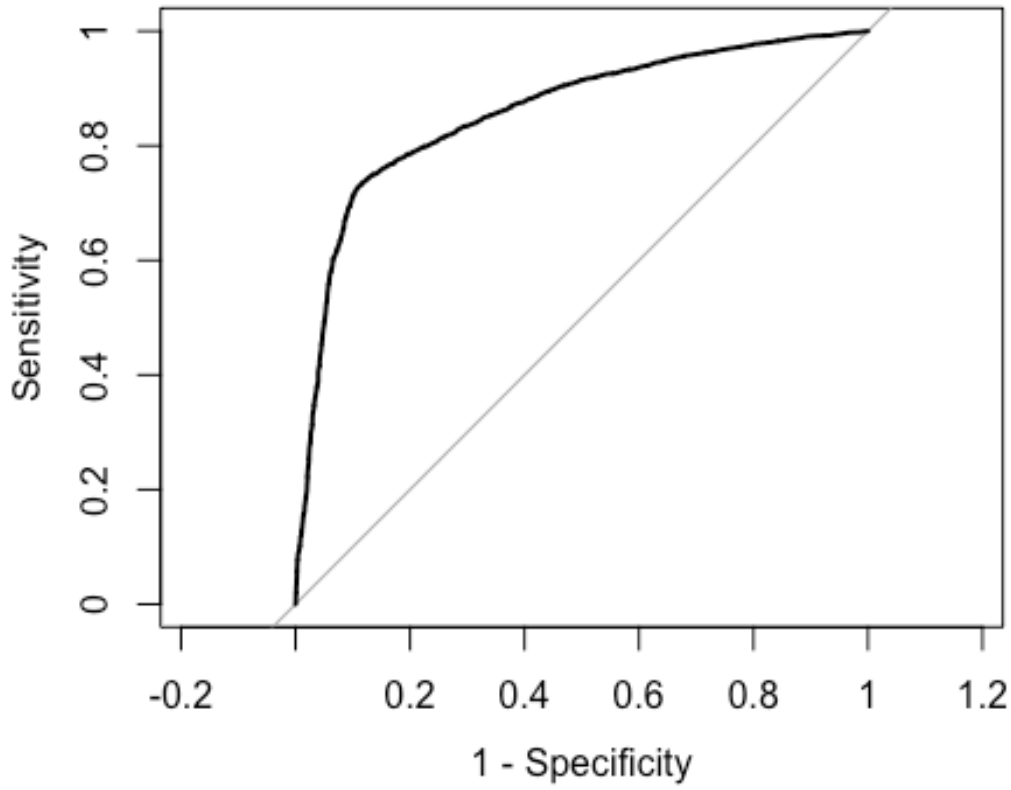
Get a set of predictions

```
pred_t=data.frame(Theft=sample_test$Is_Theft,
                  Pred=predict(logr_t,newdata =sample_test,  type="res
ponse"))
pred_t$Theft<-as.factor(pred_t$Theft)
pred_t$PredClass=ifelse(pred_t$Pred > 0.5, "1","0")
pred_t[1:10,]
```

```
##        Theft       Pred PredClass
## 665214     1 0.65138705         1
## 94312      1 0.76450131         1
## 482534     0 0.06635772         0
## 257399     1 0.52351633         1
## 109670     0 0.78272928         1
## 60992      0 0.66173981         1
## 569749     1 0.61752161         1
## 223696     0 0.06796680         0
## 267165     0 0.03673698         0
## 381814     1 0.88773645         1
```

Evaluate predictions

```
rocCurve_t <- roc(response = pred_t$Theft,
                  predictor = pred_t$Pred, levels = rev(levels(pred_t$
Theft)))
plot(rocCurve_t, legacy.axes = TRUE)
```




```
confusionMatrix(data =pred_t$PredClass, reference = pred_t$Theft, posi
tive = "1")
```

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction    0    1
##          0 4362  747
##          1 1304 3587
##
##                Accuracy : 0.7949
##                  95% CI : (0.7869, 0.8028)
##     No Information Rate : 0.5666
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.5886
```

```
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.8276
##             Specificity : 0.7699
##          Pos Pred Value : 0.7334
##          Neg Pred Value : 0.8538
##              Prevalence : 0.4334
##          Detection Rate : 0.3587
##    Detection Prevalence : 0.4891
##       Balanced Accuracy : 0.7987
##
##        'Positive' Class : 1
##
```

We will see a detailed ROC curve.

First, we made a function to make a plot at once.

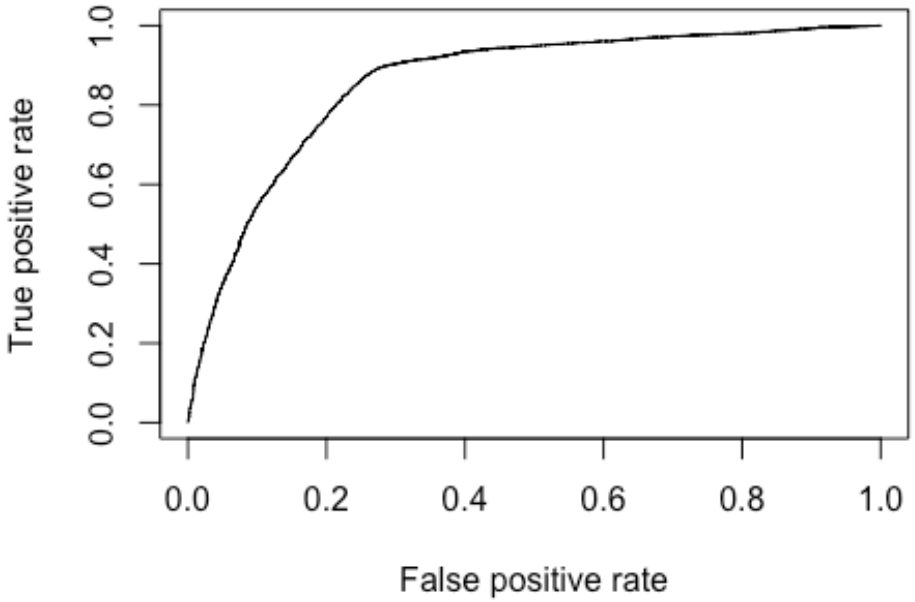
```
Toplot <- function(performance_f){
  plot(performance_f)
  plot(performance_f, xlab="1 - Specificity",ylab="Sensitivity")
  #color coded ROC curve
  plot(performance_f,colorize=TRUE)
  # color coded and annotated ROC curve
  plot(performance_f,colorize=TRUE,print.cutoffs.at=seq(0,1,0.2),text.
adj=c(-0.3,2),
     xlab='1-Specificity',ylab='Sensitivity')
}
```

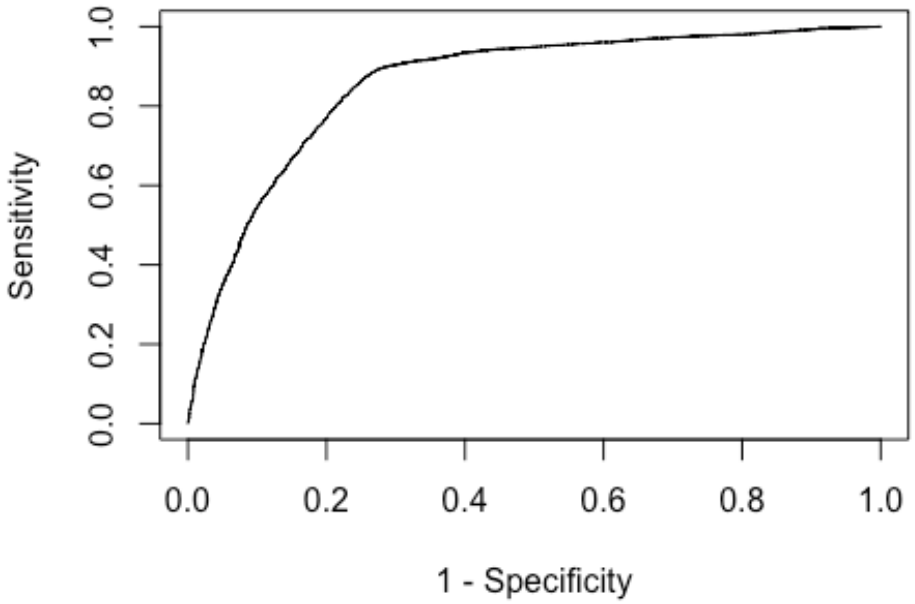
Now, a detailed ROC curve of THEFT

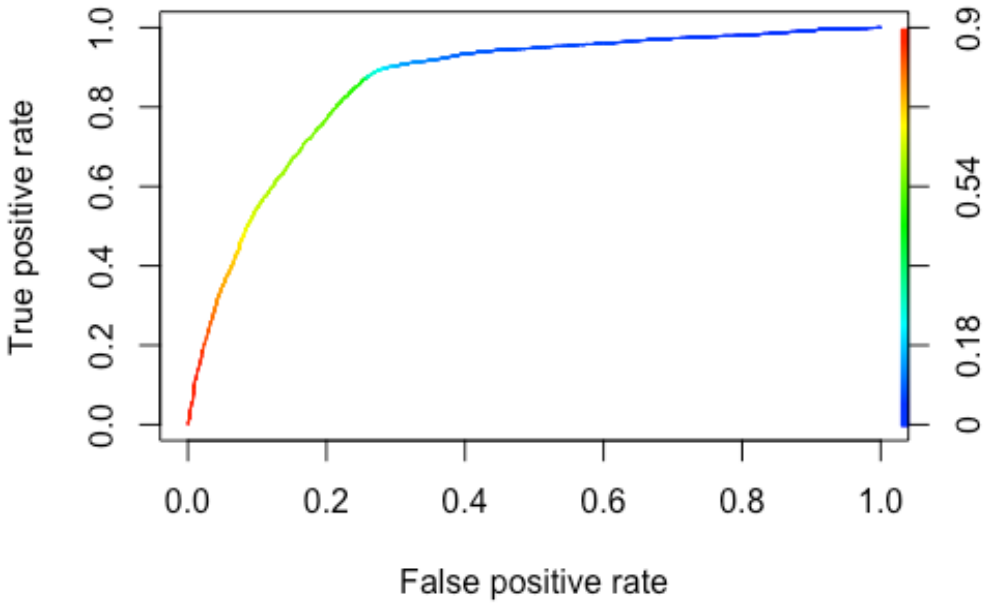
```
pred_t2 = predict(logr_t,newdata=sample_test, type='response')
ROCRpred_the = prediction(pred_t2,sample_test$Is_Theft)
as.numeric(performance(ROCRpred_the,measure='auc')@y.values)
```

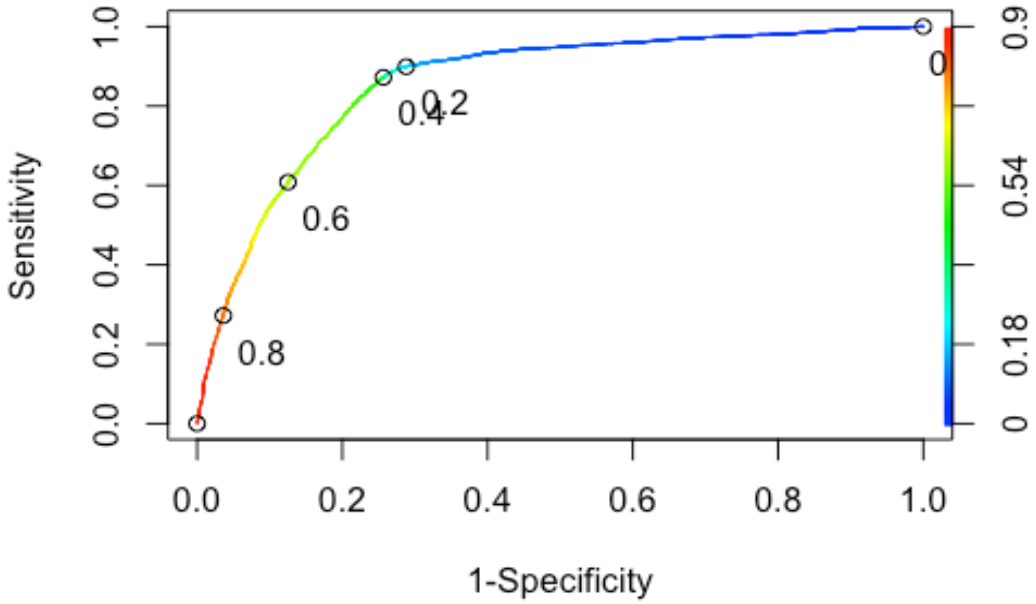
```
## [1] 0.8585327
```

```
ROCRperf_the = performance(ROCRpred_the,'tpr','fpr')
Toplot(ROCRperf_the)
```


True positive rate
0.0
0.2
0.4
0.6
0.8
1.0
False positive rate


Sensitivity
0.0
0.2
0.4
0.6
0.8
1.0
1 - Specificity

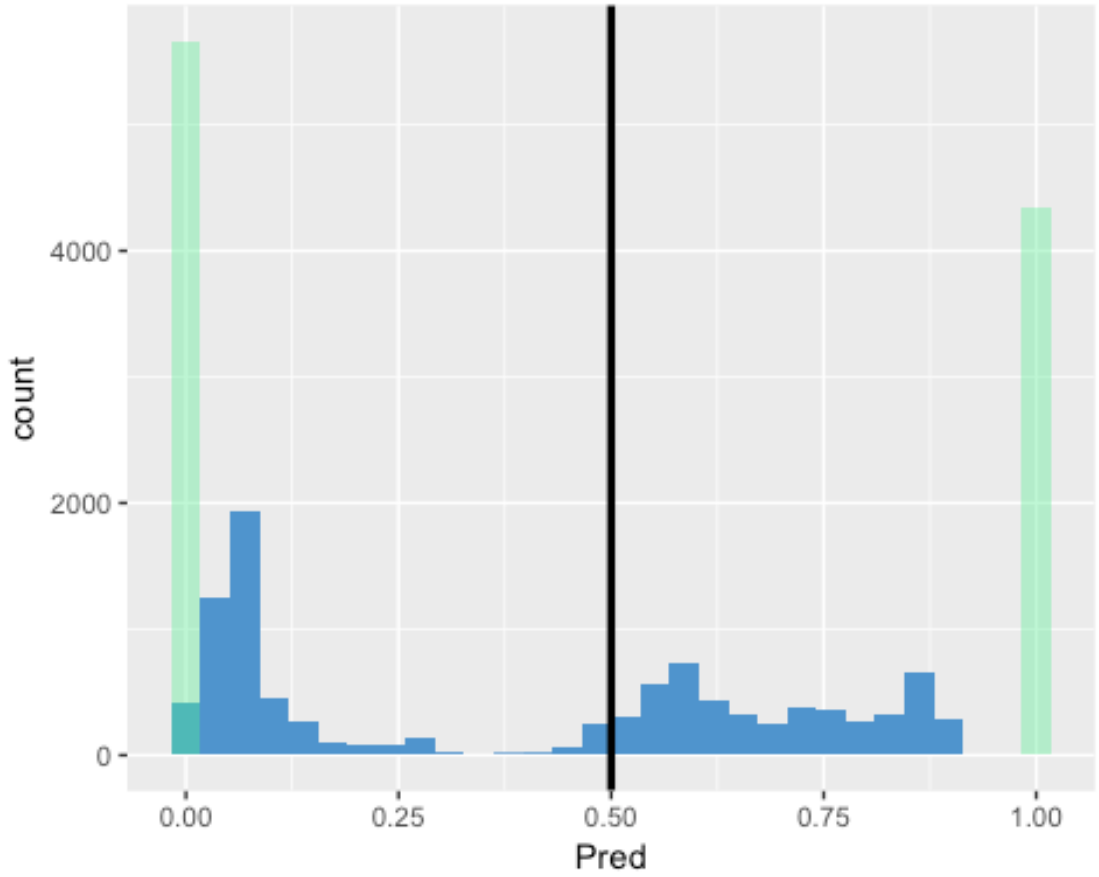




We will see a histogram of all predictions with `Is_Theft` data overlaid.

```
gg_the<- ggplot(data=pred_t,aes(x=Pred))+
  geom_histogram(fill='steelblue3')+
  geom_vline(xintercept=0.5,size=1.2)+
  geom_histogram(data=sample_test,aes(x=Is_Theft),fill='seagreen2',alp
ha=0.4)
gg_the
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



2)NARCOTICS

Variation selection method

```
null_B=glm(Is_Narcotics ~ 1, data=ABC, family=binomial)
full_B=glm(Is_Narcotics ~
             Community_Area+Domestic+Arrest+
             Edu_Rate+Foreign_Rate+Income+Pov_Rate+Month,
           data=ABC, family=binomial)
step(null_B, scope=list(lower=null_B, upper=full_B), direction="forwar
d")
```

```
## Start:  AIC=42011.94
## Is_Narcotics ~ 1
##
##                  Df Deviance   AIC
## + Arrest          1    17274 17278
## + Domestic        1    37069 37073
## + Income          1    39650 39654
## + Edu_Rate        1    39754 39758
## + Pov_Rate        1    40107 40111
## + Foreign_Rate    1    41870 41874
## + Community_Area  1    41933 41937
## + Month           1    41943 41947
## <none>                 42010 42012
##
## Step:  AIC=17277.97
## Is_Narcotics ~ Arrest
##
##                  Df Deviance   AIC
## + Domestic        1    13706 13712
## + Income          1    16377 16383
## + Edu_Rate        1    16438 16444
## + Pov_Rate        1    16537 16543
## + Foreign_Rate    1    17209 17215
## + Community_Area  1    17238 17244
## + Month           1    17254 17260
## <none>                 17274 17278
##
## Step:  AIC=13711.78
## Is_Narcotics ~ Arrest + Domestic
##
##                  Df Deviance   AIC
## + Income          1    12546 12554
## + Edu_Rate        1    12619 12627
## + Pov_Rate        1    12793 12801
## + Community_Area  1    13618 13626
## + Foreign_Rate    1    13633 13641
## + Month           1    13687 13695
## <none>                 13706 13712
##
## Step:  AIC=12554.04
## Is_Narcotics ~ Arrest + Domestic + Income
##
##                  Df Deviance   AIC
## + Edu_Rate        1    12505 12515
## + Month           1    12533 12543
## + Foreign_Rate    1    12539 12549
```

```
## + Pov_Rate        1    12542 12552
## + Community_Area  1    12543 12553
## <none>                 12546 12554
##
## Step:  AIC=12515.17
## Is_Narcotics ~ Arrest + Domestic + Income + Edu_Rate
##
##                  Df Deviance   AIC
## + Pov_Rate        1    12487 12499
## + Community_Area  1    12488 12500
## + Foreign_Rate    1    12490 12502
## + Month           1    12492 12504
## <none>                 12505 12515
##
## Step:  AIC=12498.9
## Is_Narcotics ~ Arrest + Domestic + Income + Edu_Rate + Pov_Rate
##
##                  Df Deviance   AIC
## + Community_Area  1    12470 12484
## + Month           1    12473 12487
## + Foreign_Rate    1    12477 12491
## <none>                 12487 12499
##
## Step:  AIC=12484.32
## Is_Narcotics ~ Arrest + Domestic + Income + Edu_Rate + Pov_Rate +
##     Community_Area
##
##                 Df Deviance   AIC
## + Foreign_Rate  1    12452 12468
## + Month         1    12457 12473
## <none>               12470 12484
##
## Step:  AIC=12468.3
## Is_Narcotics ~ Arrest + Domestic + Income + Edu_Rate + Pov_Rate +
##     Community_Area + Foreign_Rate
##
##         Df Deviance   AIC
## + Month  1    12439 12457
## <none>       12452 12468
##
## Step:  AIC=12457.19
## Is_Narcotics ~ Arrest + Domestic + Income + Edu_Rate + Pov_Rate +
##     Community_Area + Foreign_Rate + Month

##
## Call:  glm(formula = Is_Narcotics ~ Arrest + Domestic + Income + Ed
```

```
u_Rate +
##     Pov_Rate + Community_Area + Foreign_Rate + Month, family = bino
mial,
##     data = ABC)
##
## Coefficients:
##    (Intercept)      Arresttrue    Domestictrue          Income
##     -4.507e+00       7.043e+00      -7.238e+00      -1.207e-05
##       Edu_Rate        Pov_Rate  Community_Area    Foreign_Rate
##     -2.082e+00       1.621e+00      -6.162e-03      -7.120e-01
##          Month
##     -2.455e-02
##
## Degrees of Freedom: 32999 Total (i.e. Null);  32991 Residual
## Null Deviance:       42010
## Residual Deviance: 12440     AIC: 12460
```

As we can see in the result, *Is_Narcotics ~ Arrest + Domestic + Income + Edu_Rate + Pov_Rate + Community_Area + Foreign_Rate + Month* has the least *AIC*, which is *12457.19*

Construct a model

```
logr_n <-glm(Is_Narcotics ~ Arrest + Domestic +
               Income + Edu_Rate + Pov_Rate + Community_Area +
               Foreign_Rate + Month, data=ABC, family=binomial)
summary(logr_n)
```

```
##
## Call:
## glm(formula = Is_Narcotics ~ Arrest + Domestic + Income + Edu_Rate
+
##     Pov_Rate + Community_Area + Foreign_Rate + Month, family = bino
mial,
##     data = ABC)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3927  -0.0963  -0.0367   0.4757   3.8836
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -4.507e+00  2.894e-01 -15.573  < 2e-16 ***
## Arresttrue      7.043e+00  1.402e-01  50.223  < 2e-16 ***
## Domestictrue   -7.238e+00  4.482e-01 -16.149  < 2e-16 ***
## Income         -1.207e-05  3.778e-06  -3.194 0.001402 **
## Edu_Rate       -2.082e+00  2.287e-01  -9.101  < 2e-16 ***
```

```
## Pov_Rate        1.621e+00  4.545e-01   3.566 0.000363 ***
## Community_Area -6.162e-03  1.252e-03  -4.923 8.51e-07 ***
## Foreign_Rate   -7.120e-01  1.686e-01  -4.223 2.41e-05 ***
## Month          -2.455e-02  6.782e-03  -3.619 0.000295 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 42010  on 32999  degrees of freedom
## Residual deviance: 12439  on 32991  degrees of freedom
## AIC: 12457
##
## Number of Fisher Scoring iterations: 9
```

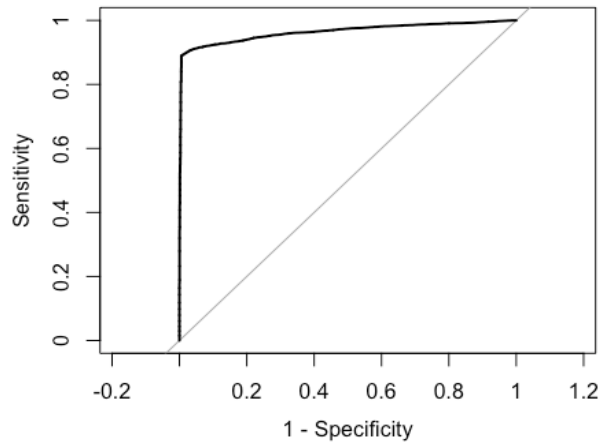
Get a set of predictions

```
pred_n=data.frame(Narcotics=sample_test$Is_Narcotics,
                  Pred=predict(logr_n,newdata =sample_test,  type="res
ponse"))
pred_n$Narcotics<-as.factor(pred_n$Narcotics)
pred_n$PredClass=ifelse(pred_n$Pred > 0.5, "1","0")
pred_n[1:10,]
```

```
##        Narcotics         Pred PredClass
## 665214         0 3.361611e-03         0
## 94312          0 1.818963e-03         0
## 482534         1 8.667606e-01         1
## 257399         0 8.505628e-03         0
## 109670         0 1.644573e-03         0
## 60992          0 3.749186e-03         0
## 569749         0 4.820232e-03         0
## 223696         1 8.835789e-01         1
## 267165         0 6.482723e-06         0
## 381814         0 5.171506e-04         0
```

Evaluate predictions

```
rocCurve_n <- roc(response = pred_n$Narcotics,
                  predictor = pred_n$Pred, levels = rev(levels(pred_n$
Narcotics)))
plot(rocCurve_n, legacy.axes = TRUE)
```

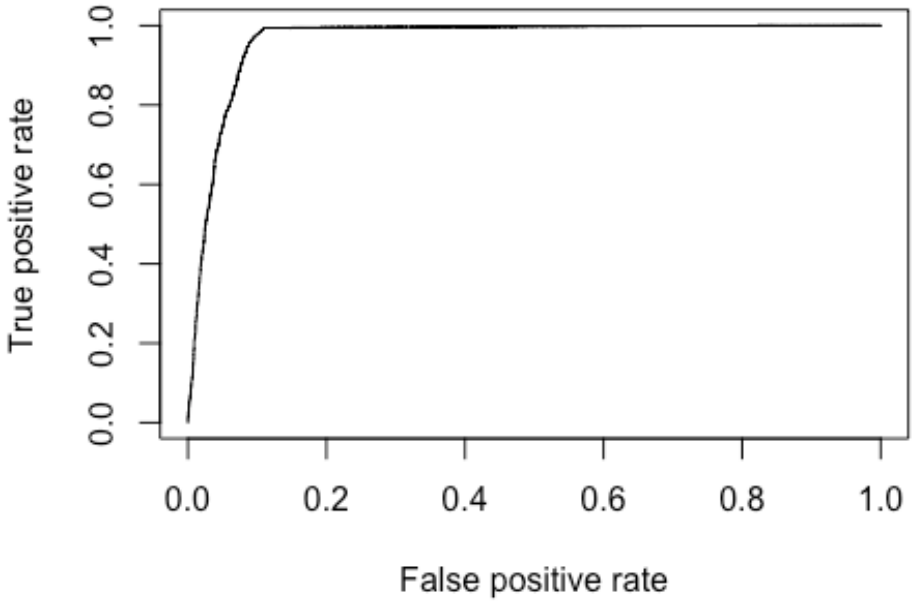


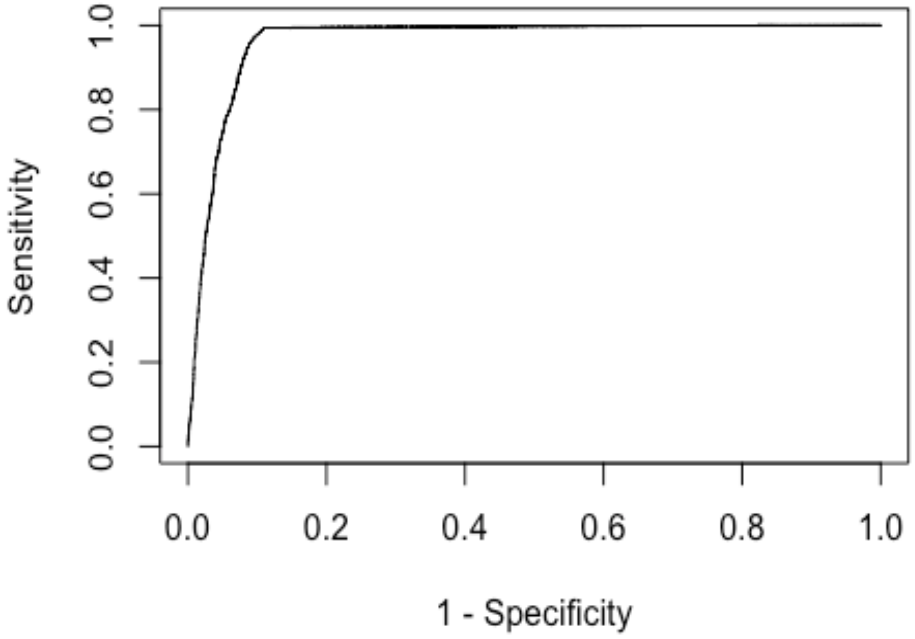

```
confusionMatrix(data =pred_n$PredClass, reference = pred_n$Narcotics,
positive = "1")
```

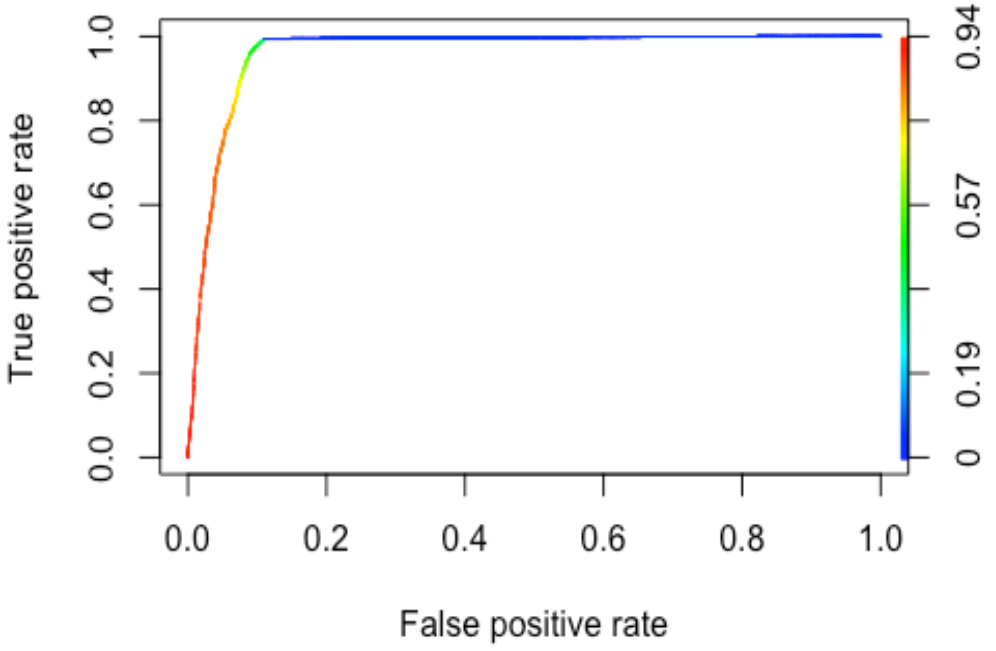
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 7215   83
##          1  716 1986
##
##                Accuracy : 0.9201
##                  95% CI : (0.9146, 0.9253)
##     No Information Rate : 0.7931
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.7813
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9599
##             Specificity : 0.9097
##          Pos Pred Value : 0.7350
##          Neg Pred Value : 0.9886
##              Prevalence : 0.2069
##          Detection Rate : 0.1986
##    Detection Prevalence : 0.2702
##       Balanced Accuracy : 0.9348
##
##        'Positive' Class : 1
##
```

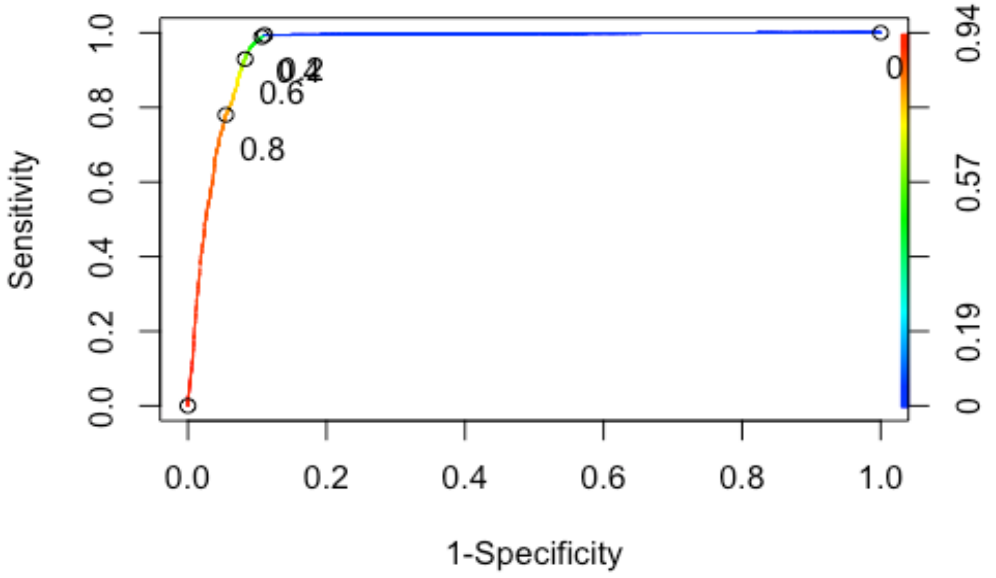
We will see a detailed ROC curve.

```
pred_n2 = predict(logr_n,newdata=sample_test, type='response')
ROCRpred_n = prediction(pred_n2,sample_test$Is_Narcotics)
as.numeric(performance(ROCRpred_n,measure='auc')@y.values)

## [1] 0.9643026

ROCRperf_nar = performance(ROCRpred_n,'tpr','fpr')
Toplot(ROCRperf_nar)
```


True positive rate
0.0
0.2
0.4
0.6
0.8
1.0
False positive rate


Sensitivity
0.0
0.2
0.4
0.6
0.8
1.0
1 - Specificity

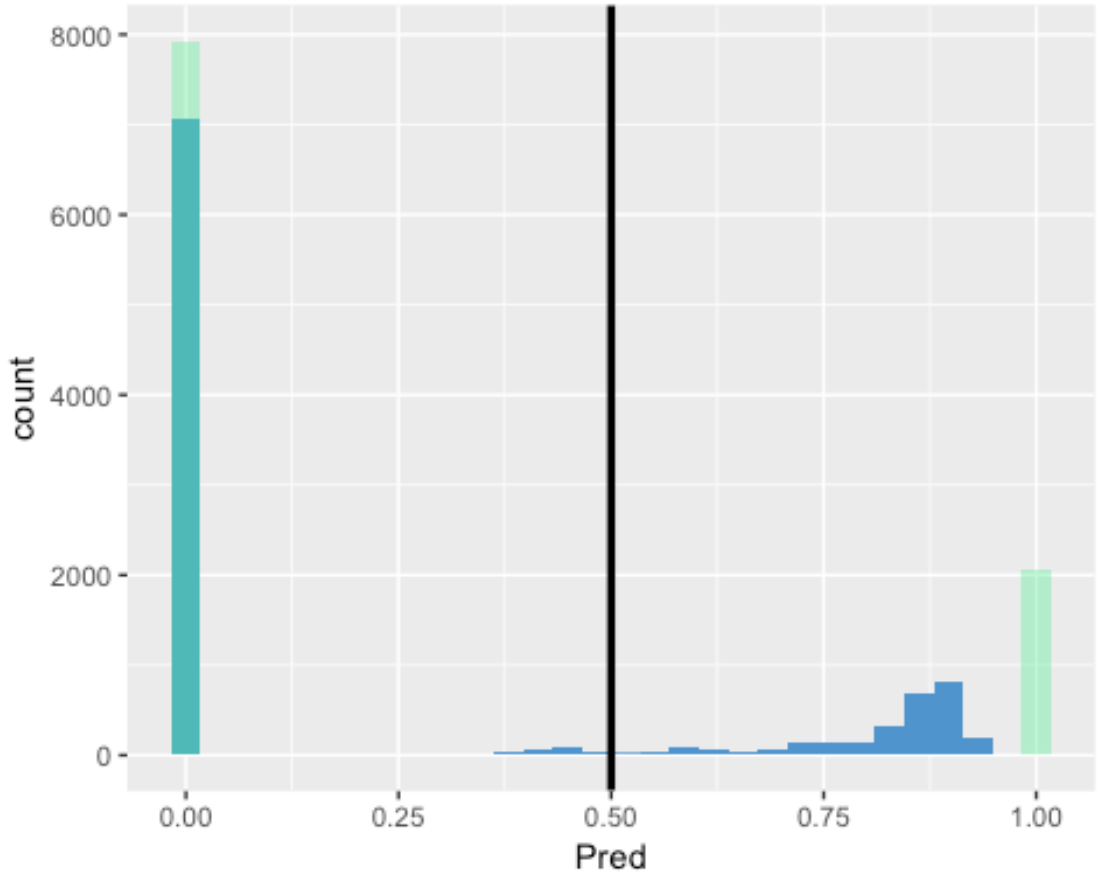




We will see a histogram of all predictions with *Is_Narcotics* data overlaid.

```
gg_nar<- ggplot(data=pred_n,aes(x=Pred))+
  geom_histogram(fill='steelblue3')+
  geom_vline(xintercept=0.5,size=1.2)+
  geom_histogram(data=sample_test,aes(x=Is_Narcotics),fill='seagreen2'
,alpha=0.4)
gg_nar
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



3)BATTERY

Variable Selection Method

```
null_C=glm(Is_Battery ~ 1, data=ABC, family=binomial)
full_C=glm(Is_Battery ~
             Community_Area+Domestic+Arrest+
             Edu_Rate+Foreign_Rate+Income+Pov_Rate+Month,
           data=ABC, family=binomial)
step(null_C, scope=list(lower=null_C, upper=full_C), direction="forwar
d")
```

```
## Start:  AIC=42011.94
## Is_Battery ~ 1
##
##                  Df Deviance   AIC
## + Domestic        1    30419 30423
## + Arrest          1    38738 38742
## + Income          1    41715 41719
## + Edu_Rate        1    41720 41724
## + Community_Area  1    41776 41780
## + Pov_Rate        1    41884 41888
## <none>                  42010 42012
## + Foreign_Rate    1    42009 42013
## + Month           1    42010 42014
##
## Step:  AIC=30422.96
## Is_Battery ~ Domestic
##
##                  Df Deviance   AIC
## + Arrest          1    28589 28595
## + Community_Area  1    30370 30376
## + Income          1    30382 30388
## + Edu_Rate        1    30391 30397
## + Pov_Rate        1    30405 30411
## + Foreign_Rate    1    30411 30417
## <none>                  30419 30423
## + Month           1    30418 30424
##
## Step:  AIC=28595.05
## Is_Battery ~ Domestic + Arrest
##
##                  Df Deviance   AIC
## + Income          1    28288 28296
## + Edu_Rate        1    28324 28332
## + Pov_Rate        1    28386 28394
## + Community_Area  1    28492 28500
## + Month           1    28583 28591
## <none>                  28589 28595
## + Foreign_Rate    1    28589 28597
##
## Step:  AIC=28295.94
## Is_Battery ~ Domestic + Arrest + Income
##
##                  Df Deviance   AIC
## + Foreign_Rate    1    28279 28289
## + Community_Area  1    28280 28290
## + Edu_Rate        1    28282 28292
```

```
## + Month           1    28283 28293
## <none>                 28288 28296
## + Pov_Rate        1    28286 28296
##
## Step:  AIC=28288.98
## Is_Battery ~ Domestic + Arrest + Income + Foreign_Rate
##
##                  Df Deviance   AIC
## + Community_Area  1    28267 28279
## + Month           1    28274 28286
## + Edu_Rate        1    28276 28288
## <none>                 28279 28289
## + Pov_Rate        1    28279 28291
##
## Step:  AIC=28278.69
## Is_Battery ~ Domestic + Arrest + Income + Foreign_Rate + Community_
Area
##
##            Df Deviance   AIC
## + Month     1    28262 28276
## <none>           28267 28279
## + Edu_Rate  1    28266 28280
## + Pov_Rate  1    28267 28281
##
## Step:  AIC=28275.59
## Is_Battery ~ Domestic + Arrest + Income + Foreign_Rate + Community_
Area +
##     Month
##
##            Df Deviance   AIC
## <none>           28262 28276
## + Edu_Rate  1    28261 28277
## + Pov_Rate  1    28262 28278

##
## Call:  glm(formula = Is_Battery ~ Domestic + Arrest + Income + Fore
ign_Rate +
##     Community_Area + Month, family = binomial, data = ABC)
##
## Coefficients:
##    (Intercept)    Domestictrue      Arresttrue          Income
##     -2.993e-01       4.131e+00      -1.436e+00      -1.286e-05
##   Foreign_Rate  Community_Area           Month
##      3.755e-01       2.720e-03      -1.017e-02
##
## Degrees of Freedom: 32999 Total (i.e. Null);  32993 Residual
```

```
## Null Deviance:       42010
## Residual Deviance: 28260     AIC: 28280
```

As we can see in the result, *Is_Battery ~ Domestic + Arrest + Income + Foreign_Rate + Community_Area + Month* has the least *AIC*, which is *28275.59*

Construct a model

```
logr_b <-glm(Is_Battery ~ Domestic + Arrest +
               Income + Foreign_Rate + Community_Area + Month,
             data=ABC, family=binomial)
summary(logr_b)

##
## Call:
## glm(formula = Is_Battery ~ Domestic + Arrest + Income + Foreign_Rat
e +
##     Community_Area + Month, family = binomial, data = ABC)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7651  -0.6984  -0.4823   0.2530   2.4461
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -2.993e-01  6.936e-02  -4.314  1.6e-05 ***
## Domestictrue    4.131e+00  6.455e-02  64.000  < 2e-16 ***
## Arresttrue     -1.436e+00  3.329e-02 -43.145  < 2e-16 ***
## Income         -1.286e-05  8.854e-07 -14.528  < 2e-16 ***
## Foreign_Rate    3.755e-01  1.043e-01   3.601 0.000316 ***
## Community_Area  2.720e-03  7.743e-04   3.513 0.000444 ***
## Month          -1.017e-02  4.500e-03  -2.259 0.023871 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 42010  on 32999  degrees of freedom
## Residual deviance: 28262  on 32993  degrees of freedom
## AIC: 28276
##
## Number of Fisher Scoring iterations: 5
```

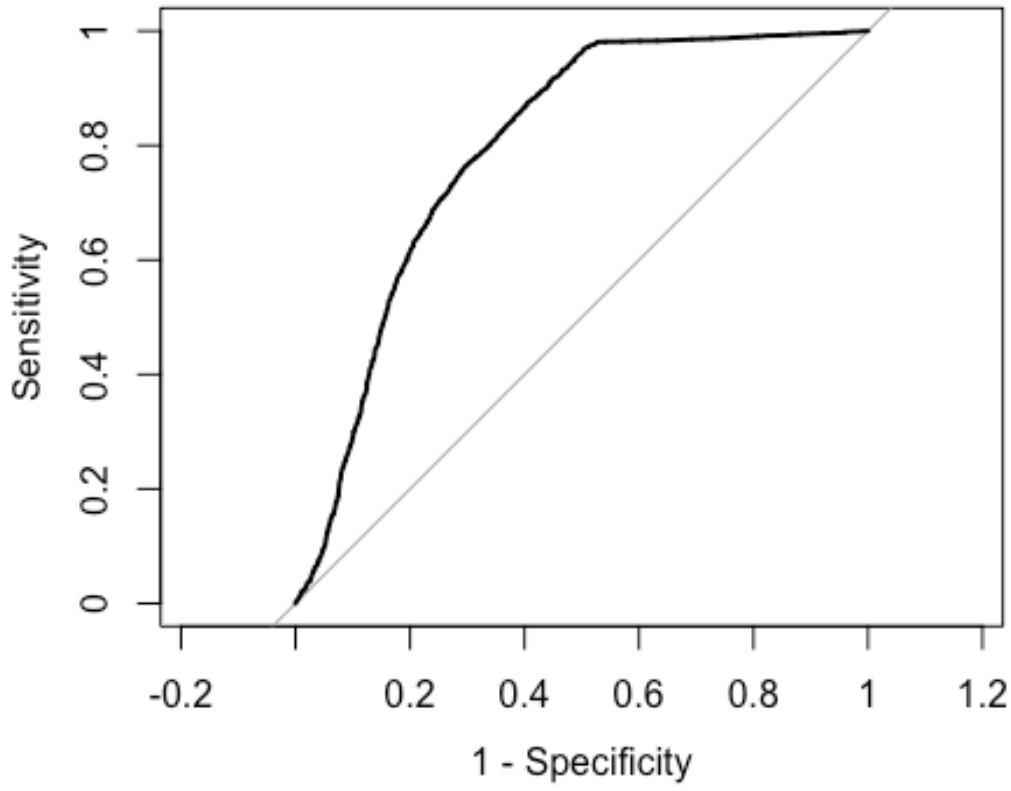
Get a set of predictions

```
pred_b=data.frame(Battery=sample_test$Is_Battery,
                  Pred=predict(logr_b,newdata =sample_test,  type="res
ponse"))
pred_b$Battery<-as.factor(pred_b$Battery)
pred_b$PredClass=ifelse(pred_b$Pred > 0.5, "1","0")
pred_b[1:10,]
```

```
##        Battery      Pred PredClass
## 665214       0 0.3278079         0
## 94312        0 0.2762504         0
## 482534       0 0.1154571         0
## 257399       0 0.3508318         0
## 109670       1 0.2646419         0
## 60992        1 0.3165928         0
## 569749       0 0.3207569         0
## 223696       0 0.1051005         0
## 267165       1 0.9717067         1
## 381814       0 0.2005638         0
```

Evaluate predictions

```
rocCurve_b <- roc(response = pred_b$Battery,
                  predictor = pred_b$Pred, levels = rev(levels(pred_b$
Battery)))
plot(rocCurve_b, legacy.axes = TRUE)
```




```
confusionMatrix(data =pred_b$PredClass, reference = pred_b$Battery, po
sitive = "1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 6280 1900
##          1  123 1697
##
##                Accuracy : 0.7977
##                  95% CI : (0.7897, 0.8055)
##     No Information Rate : 0.6403
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.5075
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.4718
##             Specificity : 0.9808
##          Pos Pred Value : 0.9324
```

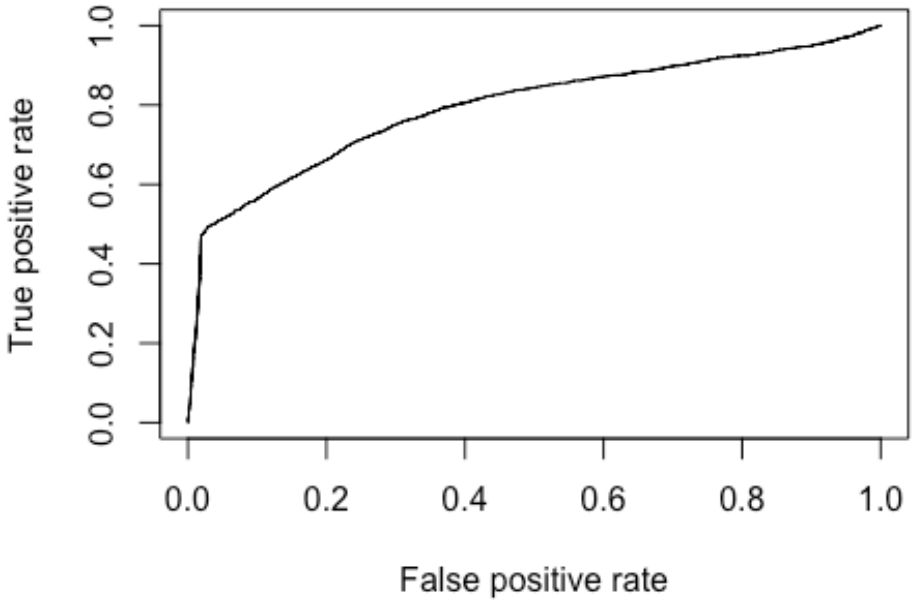
```
##          Neg Pred Value : 0.7677
##              Prevalence : 0.3597
##          Detection Rate : 0.1697
##    Detection Prevalence : 0.1820
##       Balanced Accuracy : 0.7263
##
##        'Positive' Class : 1
##
```

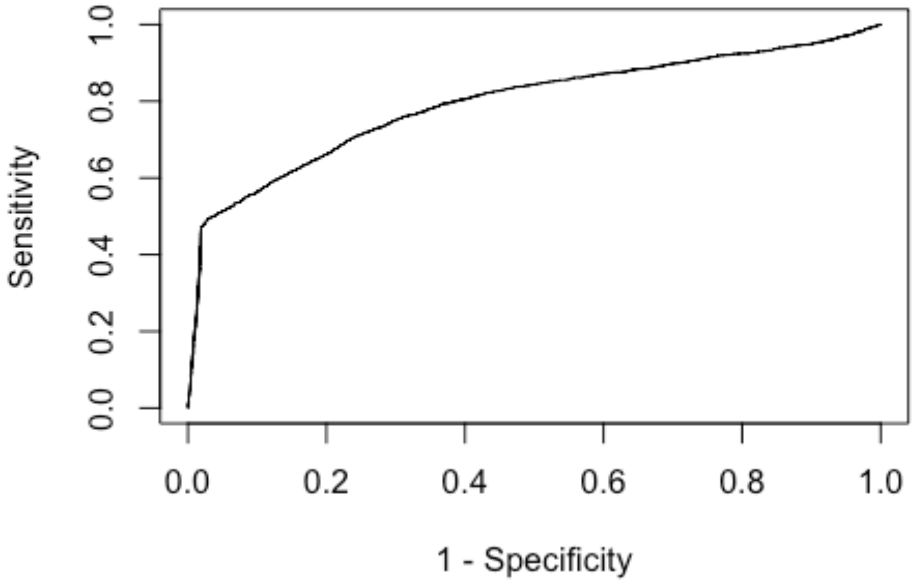
We will see a detailed ROC curve.

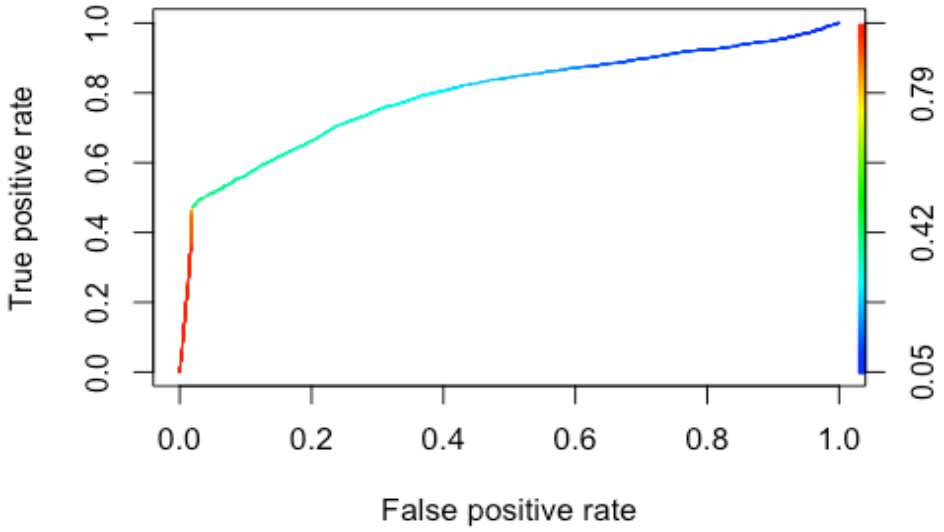
```
pred_b2 = predict(logr_b,newdata=sample_test, type='response')
ROCRpred_b = prediction(pred_b2,sample_test$Is_Battery)
as.numeric(performance(ROCRpred_b,measure='auc')@y.values)
```

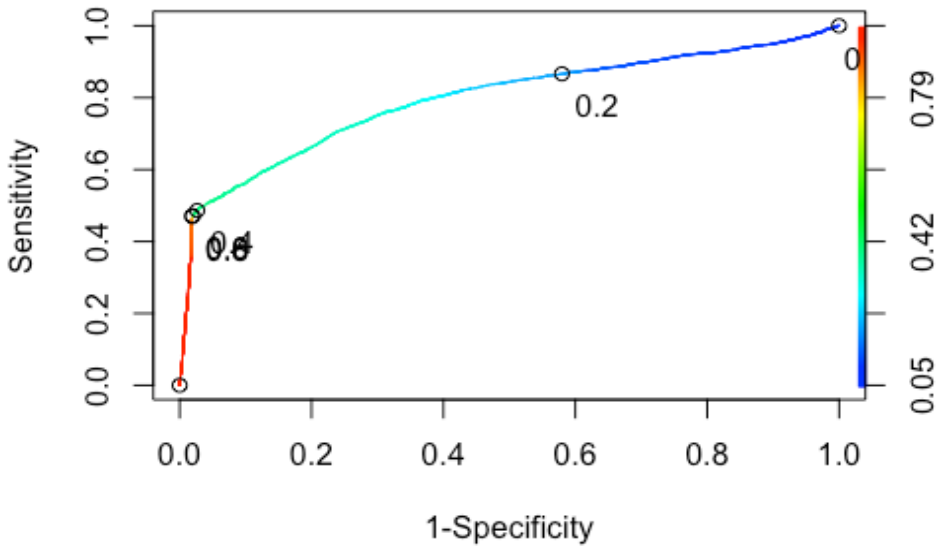
```
## [1] 0.7954997
```

```
ROCRperf_bat = performance(ROCRpred_b,'tpr','fpr')
Toplot(ROCRperf_bat)
```


True positive rate
0.0
0.2
0.4
0.6
0.8
1.0
False positive rate


Sensitivity
0.0
0.2
0.4
0.6
0.8
1.0
1 - Specificity

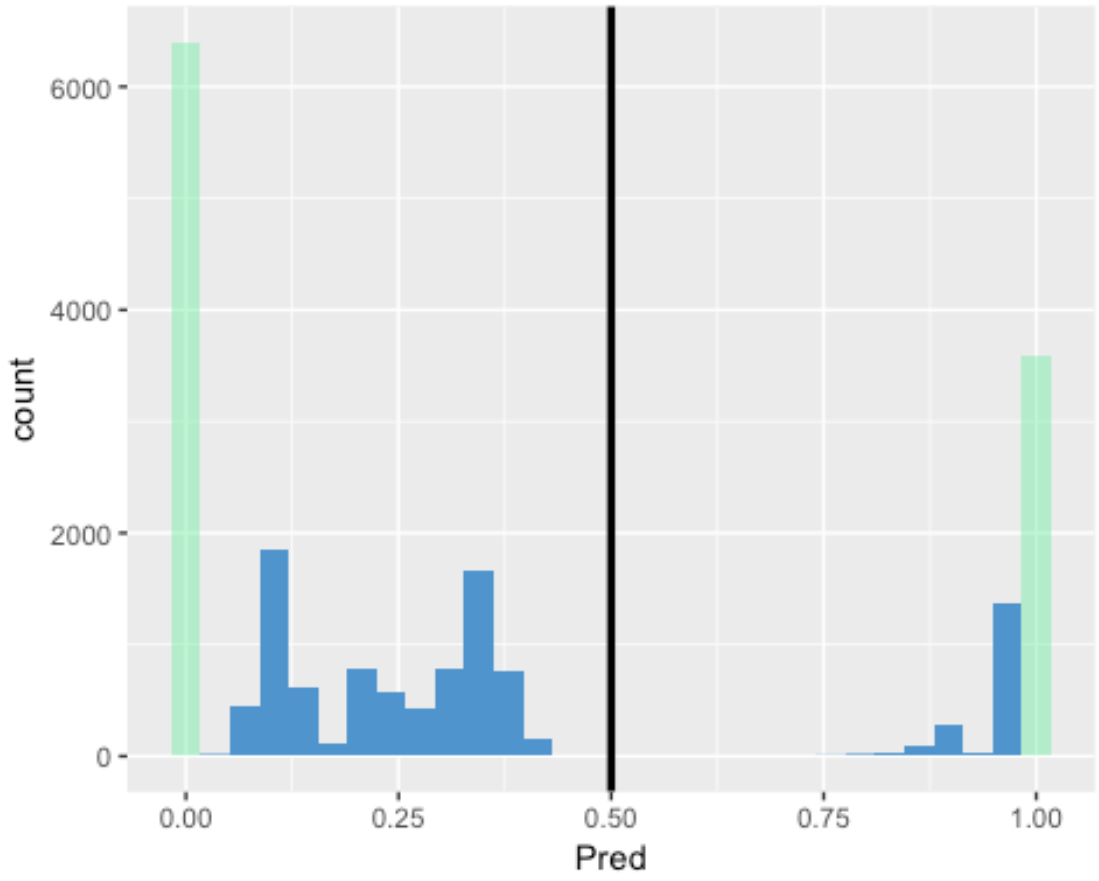




We will see a histogram of all predictions with *Is_Battery* data overlaid.

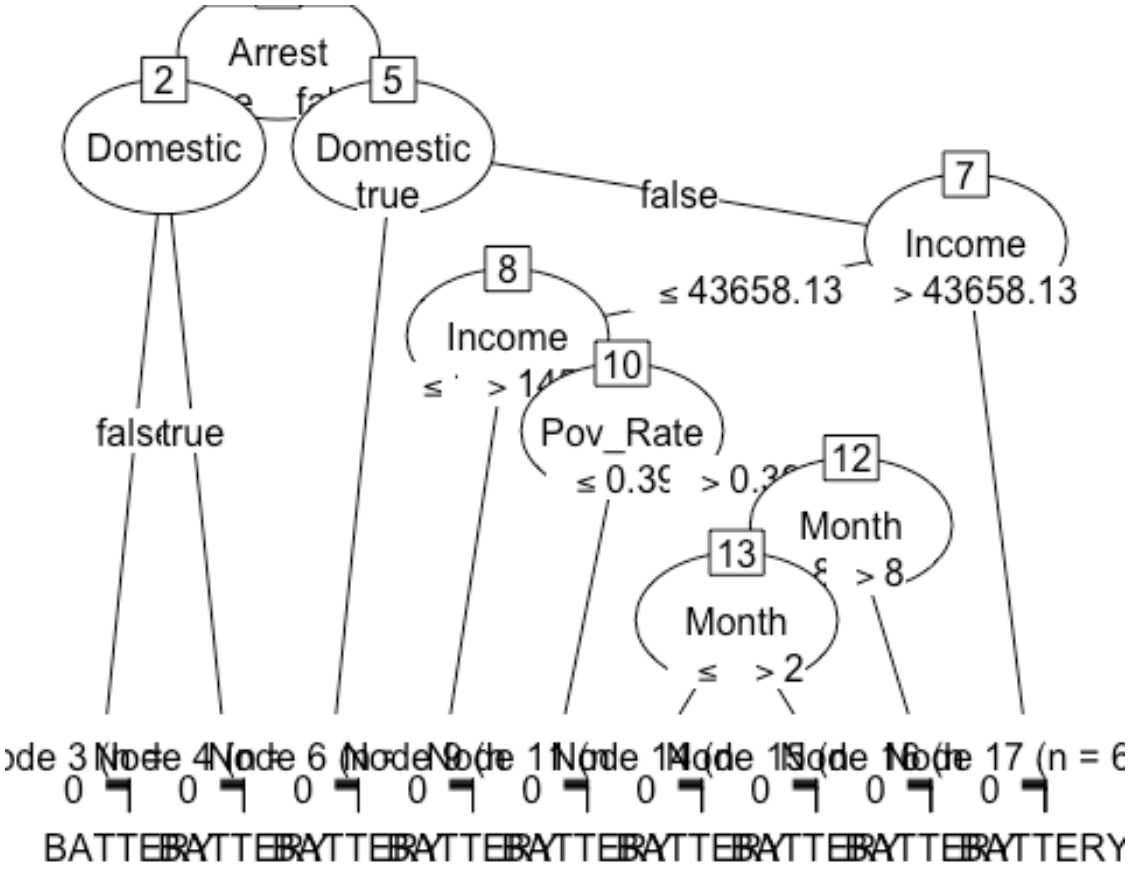
```
gg_bat<- ggplot(data=pred_b,aes(x=Pred))+
  geom_histogram(fill='steelblue3')+
  geom_vline(xintercept=0.5,size=1.2)+
  geom_histogram(data=sample_test,aes(x=Is_Battery),fill='seagreen2',a
lpha=0.4)
gg_bat
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



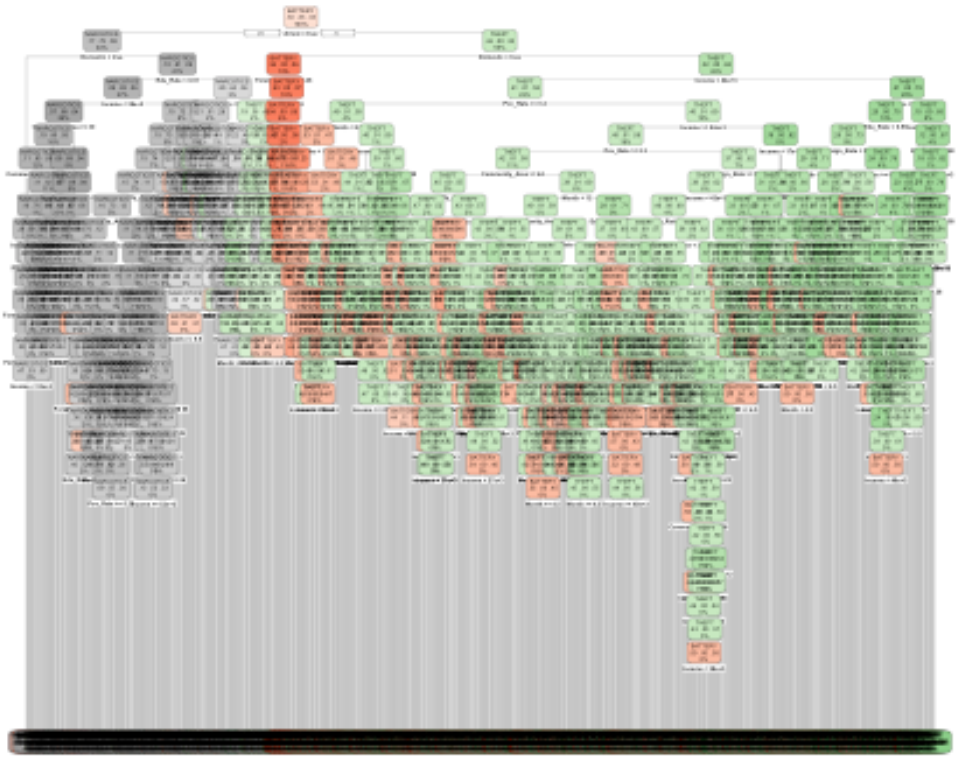
(2) C5.0 Decision Tree

```
c_model <- C5.0(Primary_Type ~ Arrest + Domestic +
                      Income + Edu_Rate + Pov_Rate + Month +
                      Community_Area + Foreign_Rate, data= ABC)
c_results <- predict(object = c_model, newdata = sample_test, type = "
class")
table(c_results, sample_test$Primary_Type)

##
## c_results   BATTERY NARCOTICS THEFT
##   BATTERY      1848         3   276
##   NARCOTICS     451      2057   427
##   THEFT        1298         9  3631

plot(c_model)
```



(3) Rpart Deecision Tree

```
rpartContr=rpart.control(minsplit = 10, cp=1e-09, minbucket = 4)
r_model <- rpart(Primary_Type ~ Arrest + Domestic +
                      Income + Edu_Rate + Pov_Rate + Month +
                      Community_Area + Foreign_Rate,  control=rpartCon
tr, data= ABC)
r_results <- predict(object = r_model, newdata = sample_test, type = "
class")
table(c_results, sample_test$Primary_Type)

##
## c_results   BATTERY NARCOTICS THEFT
##   BATTERY      1848         3   276
##   NARCOTICS     451      2057   427
##   THEFT        1298         9  3631

rpart.plot(r_model)

## Warning: labs do not fit even at cex 0.15, there may be some overpl
otting
```



Since we tried to PREDICT which type of crimes is highly likely to happen in the future, we thought that both decision tree models were not suitable in our case.

(4) Radom Forest Test

```
randomTest <- rfsrc(Primary_Type ~ Arrest + Domestic +
                      Income + Edu_Rate + Pov_Rate + Month +
                      Community_Area + Foreign_Rate, data= ABC)
randomTest
```

```
##                          Sample size: 33000
##            Frequency of class labels: 11000, 11000, 11000
##                      Number of trees: 1000
##            Forest terminal node size: 1
##        Average no. of terminal nodes: 3478.872
## No. of variables tried at each split: 3
##               Total no. of variables: 8
##                                Analysis: RF-C
##                                  Family: class
##                        Splitting rule: gini
##               Normalized Brier score: 51.93
##                            Error rate: 0.24, 0.42, 0.06, 0.25
```

```
##
## Confusion matrix:
##
##            predicted
##   observed  BATTERY NARCOTICS THEFT class.error
##   BATTERY      6333      1162  3505      0.4243
##   NARCOTICS     268     10383   349      0.0561
##   THEFT        1875       903  8222      0.2525
##
##  Overall error rate: 24.43%
```