VamRegression

Kyle Amyx

6/12/2019

Regressing VAM on Logistic Params for years 1954 and 1958

```
params <- read.csv("TractorCoef.csv")
vam <- read.csv("vamFipsData.csv")
params[,1] <- NULL
vam[,1] <- NULL
```

Merge params DF and vam DF

INFO:

The dataset i'm using matches each county and year to their corresponding logistic params.(Slope, Ceiling, Mid) Any counties with a Negative slope have been removed Any counties with a ceiling > 1 have been removed. Any counties with a midpoint before 1900 or after 1980 have been removed Any counties with VAM as NA have been removed Any counties with VAM as 0 have been removed

```
#remove high ceilings
df <- df[df$Ceiling <= 1, ]
df <- df[df$Mid > 1940,]
df <- df[df$Mid < 1960,]
df <- df[!is.na(df$VAM),]
#Remove response variables that are 0
df <- df[which(df$VAM > 0),]
```

The next step necessary for fitting a logistic is to create a binary response

I've set the `response` as a binary value(1 = "Success", 0 = "Failure"), this value was determined by wether a county had VAM of > 250 or not. This is arbitrary and can be played with to see if it changes the result. I chose 250 b/c the range of VAM is (1,999) however majority of these values are below 500 so it seemed like a good starting point.

```
df$response <- ifelse(df$VAM >= 250, 1, 0)
```

Before subsetting the data and get into modeling, I want to remove any N/A's.

```
unique(is.na(df))
```

```
##     fips Slope Ceiling   Mid County.x State   VAM  year response
## 12 FALSE FALSE   FALSE FALSE    FALSE FALSE FALSE FALSE    FALSE
```

As seen above, none of the columns contain any N/A values

Below is a glimpse at the current dataset that will be used in the modeling. Next step is to reduce the observations to only rows that correspond to the years 1954 or 1958

```
head(df)
```

```
##    fips     Slope   Ceiling      Mid       County.x State VAM year
## 12 1009 0.2020403 0.9049963 1956.463         BLOUNT    AL  15 1929
## 13 1009 0.2020403 0.9049963 1956.463         BLOUNT    AL 140 1958
## 14 1009 0.2020403 0.9049963 1956.463         BLOUNT    AL  38 1947
## 21 1015 0.1737442 0.9187005 1953.665 CALHOUN/BENTON    AL 216 1929
## 22 1015 0.1737442 0.9187005 1953.665 CALHOUN/BENTON    AL 751 1958
## 23 1015 0.1737442 0.9187005 1953.665 CALHOUN/BENTON    AL 797 1954
##    response
## 12        0
## 13        0
## 14        0
## 21        0
## 22        1
## 23        1
```

```
df.54 <- subset(df, year == 1954)
rownames(df.54) <- 1:nrow(df.54)
df.58 <- subset(df, year == 1958)
rownames(df.58) <- 1:nrow(df.58)
```

Glimpse into 1954 dataset

```
head(df.54)
```

```
##   fips     Slope   Ceiling      Mid       County.x State VAM year response
## 1 1015 0.1737442 0.9187005 1953.665 CALHOUN/BENTON    AL 797 1954        1
## 2 1043 0.2265686 0.8037223 1954.990        CULLMAN    AL  76 1954        0
## 3 1045 0.1738427 0.9419194 1956.008           DALE    AL 128 1954        0
## 4 1049 0.1862319 0.8949618 1956.816        DE KALB    AL 140 1954        0
## 5 1053 0.1624123 0.9736148 1957.202       ESCAMBIA    AL 298 1954        1
## 6 1061 0.2448450 0.8566386 1953.288         GENEVA    AL 244 1954        0
```

Glimpse into 1958 dataset

```
head(df.58)
```

```
##   fips     Slope   Ceiling      Mid       County.x State VAM year response
## 1 1009 0.2020403 0.9049963 1956.463         BLOUNT    AL 140 1958        0
## 2 1015 0.1737442 0.9187005 1953.665 CALHOUN/BENTON    AL 751 1958        1
## 3 1019 0.1804967 0.8845261 1949.885       CHEROKEE    AL  86 1958        0
## 4 1043 0.2265686 0.8037223 1954.990        CULLMAN    AL 109 1958        0
## 5 1045 0.1738427 0.9419194 1956.008           DALE    AL 289 1958        1
## 6 1049 0.1862319 0.8949618 1956.816        DE KALB    AL 188 1958        0
```

1954 Training and Test Set

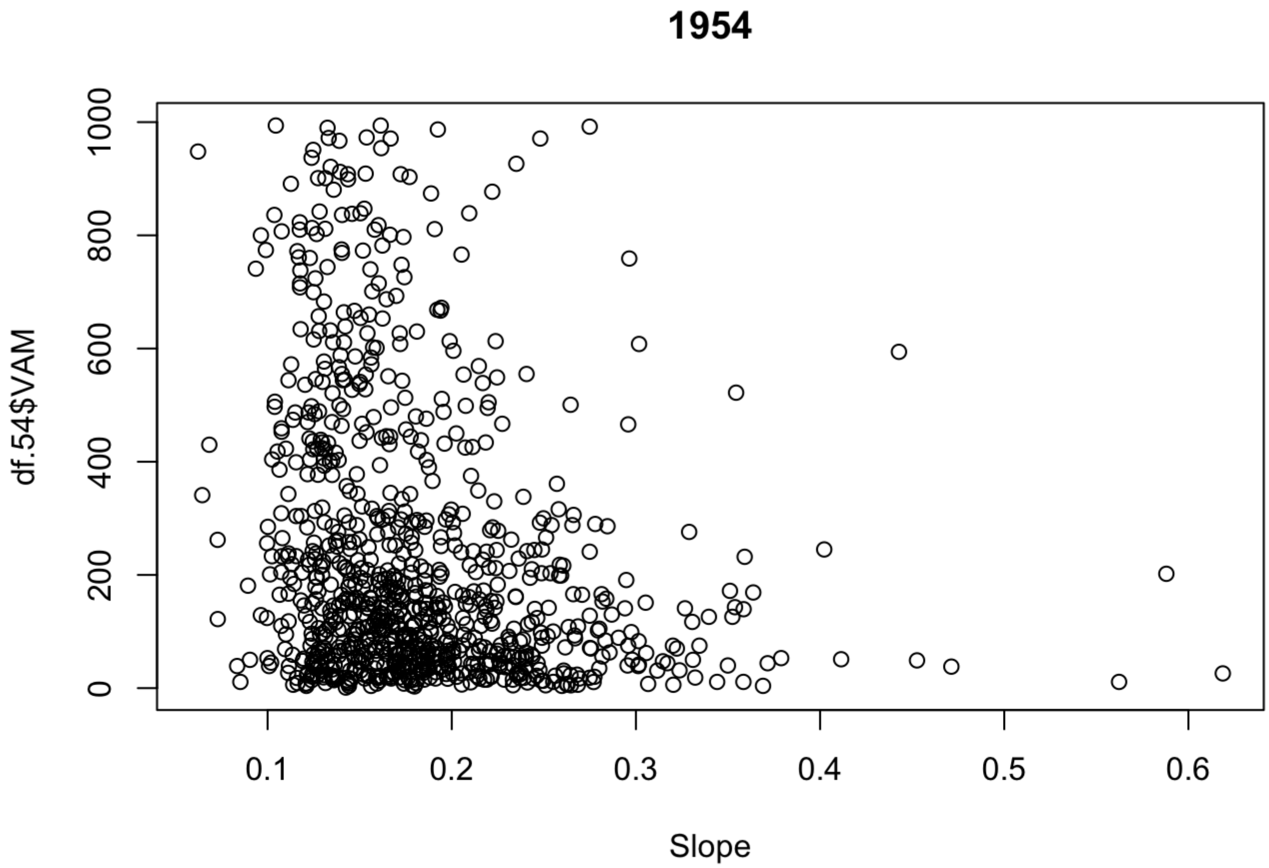
```
smp_size <- floor(.7 * nrow(df.54))
set.seed(123)
train_ind <- sample(seq_len(nrow(df.54)), size = smp_size)

df.54.train <- df.54[train_ind,]

df.54.test <- df.54[-train_ind,]
```

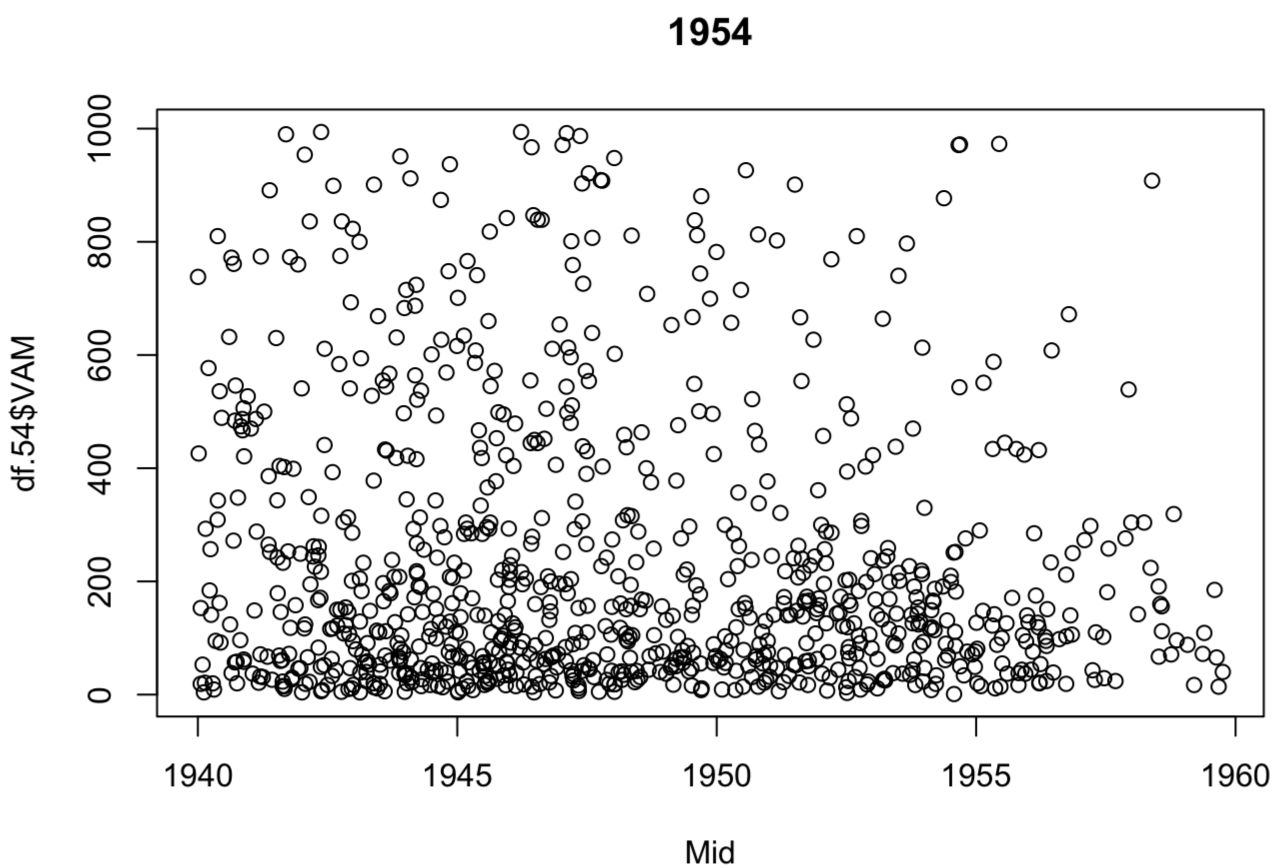
1954 Model w/o Transformations and Plots

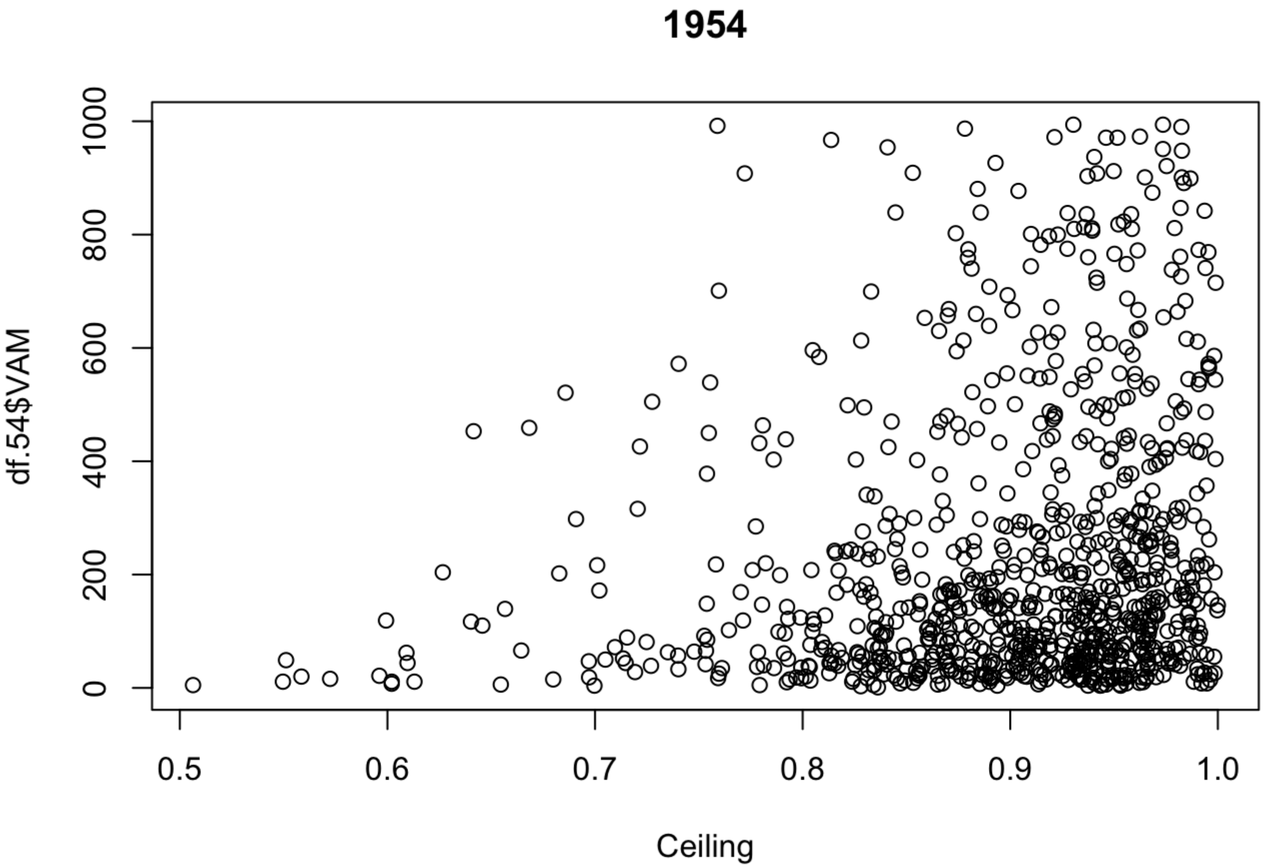
Below are some exploratory plots containing the predictor features for year 1954.

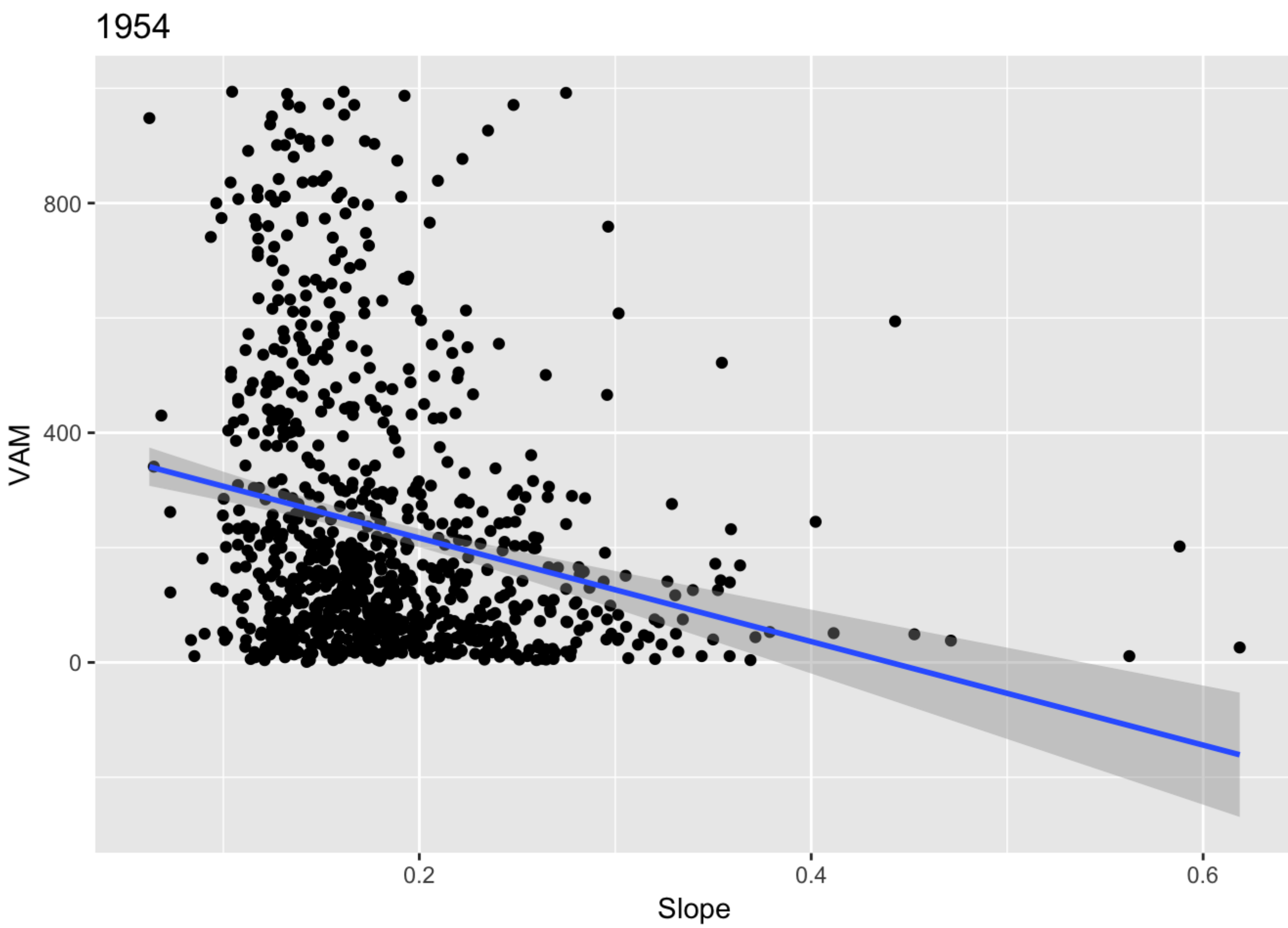
The main one that I want to point out is [VAM vs. Slope], it seems counter-intuitive to me that as the slope increase for the county, the VAM is decreasing.

Maybe you have some other insight into this??


1954
df.54$VAM
0
200
400
600
800
1000
0.1
0.2
0.3
0.4
0.5
0.6
Slope


1954
df.54$VAM
0
200
400
600
800
1000
1940
1945
1950
1955
1960
Mid


1954
df.54$VAM
0
200
400
600
800
1000
0.5
0.6
0.7
0.8
0.9
1.0
Ceiling


1954
VAM
0
400
800
0.2
0.4
0.6
Slope

```
##       fips      Slope   Ceiling       Mid  County.x     State       VAM
## "integer" "numeric" "numeric" "numeric"  "factor"  "factor" "numeric"
##       year  response
## "integer" "numeric"
```

54 Model Summary

Below is the actual modeling for year 1954 on training set

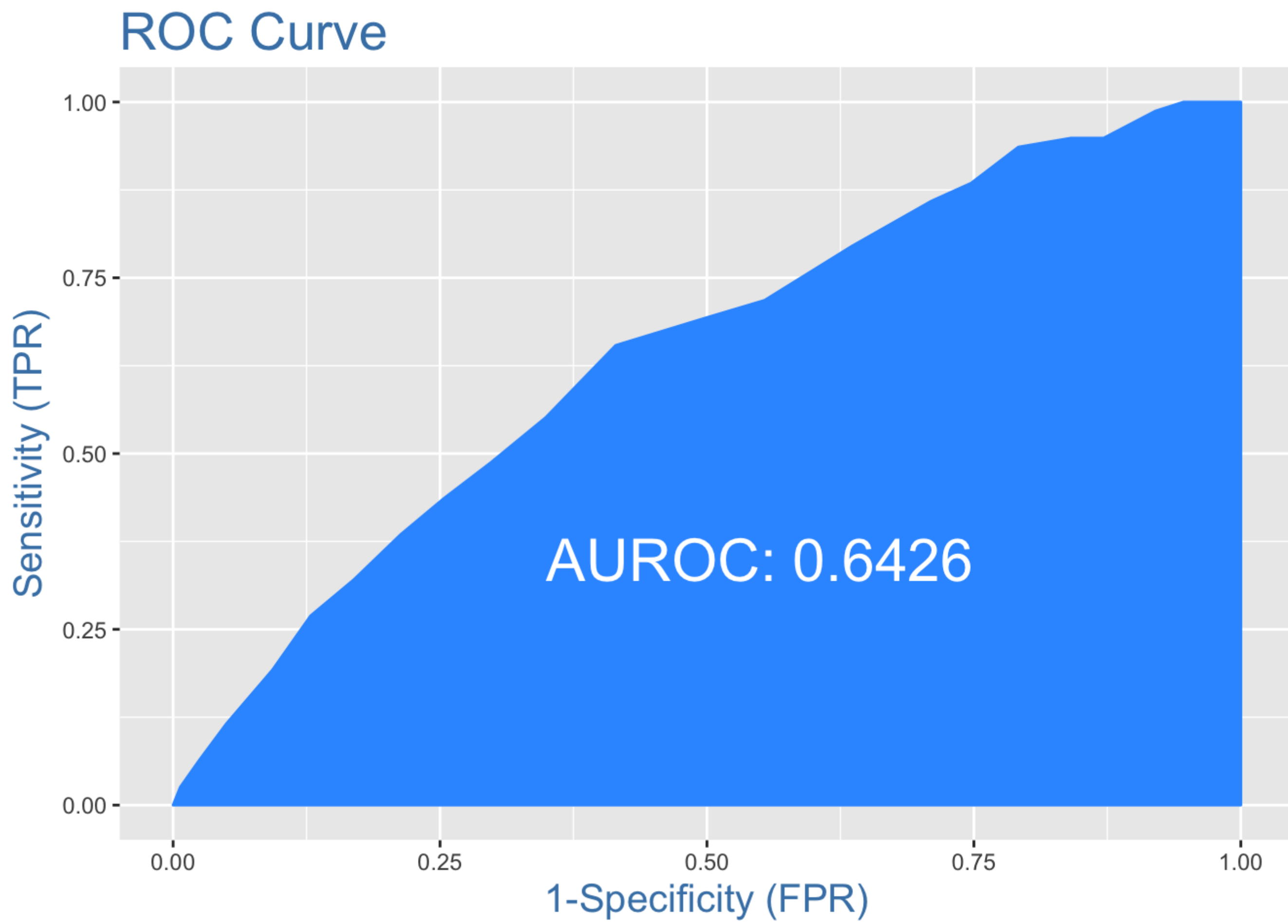
```
model <- glm(response ~ Slope + Ceiling + Mid,  data = df.54.train, family=binomial(link
= "logit"))
summary(model)
```

```
##
## Call:
## glm(formula = response ~ Slope + Ceiling + Mid, family = binomial(link = "logit"),
##     data = df.54.train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.2703  -0.9242  -0.7238   1.2440   2.1916
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) 75.91243   36.67858   2.070 0.038484 *
## Slope       -6.31949    1.91234  -3.305 0.000951 ***
## Ceiling      2.74935    1.37330   2.002 0.045285 *
## Mid         -0.04005    0.01884  -2.126 0.033525 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 797.78  on 629  degrees of freedom
## Residual deviance: 759.48  on 626  degrees of freedom
## AIC: 767.48
##
## Number of Fisher Scoring iterations: 4
```

```
predicted.54 <- plogis(predict(model, df.54.test))
```

Looking at the model summary above ^, this tells me that as both slope and mid increase independent of one another, VAM will decrease. However, as the Ceiling parameter increases, VAM will increase with. I'm not sure how to interpret this, maybe to be discussed??

```
## [1] 0.263
```



This model is able to predict `Success` counties 30% of the time, doesn't seem very good but at the same time we dont have much to work with

```
#Sensitivity(Truth Detection Rate)
sensitivity(df.54.test$response, predicted.54, threshold = optCutOff)
```

```
## [1] 0.2987013
```

```
confusionMatrix(df.54.test$response, predicted.54, threshold = optCutOff)
```

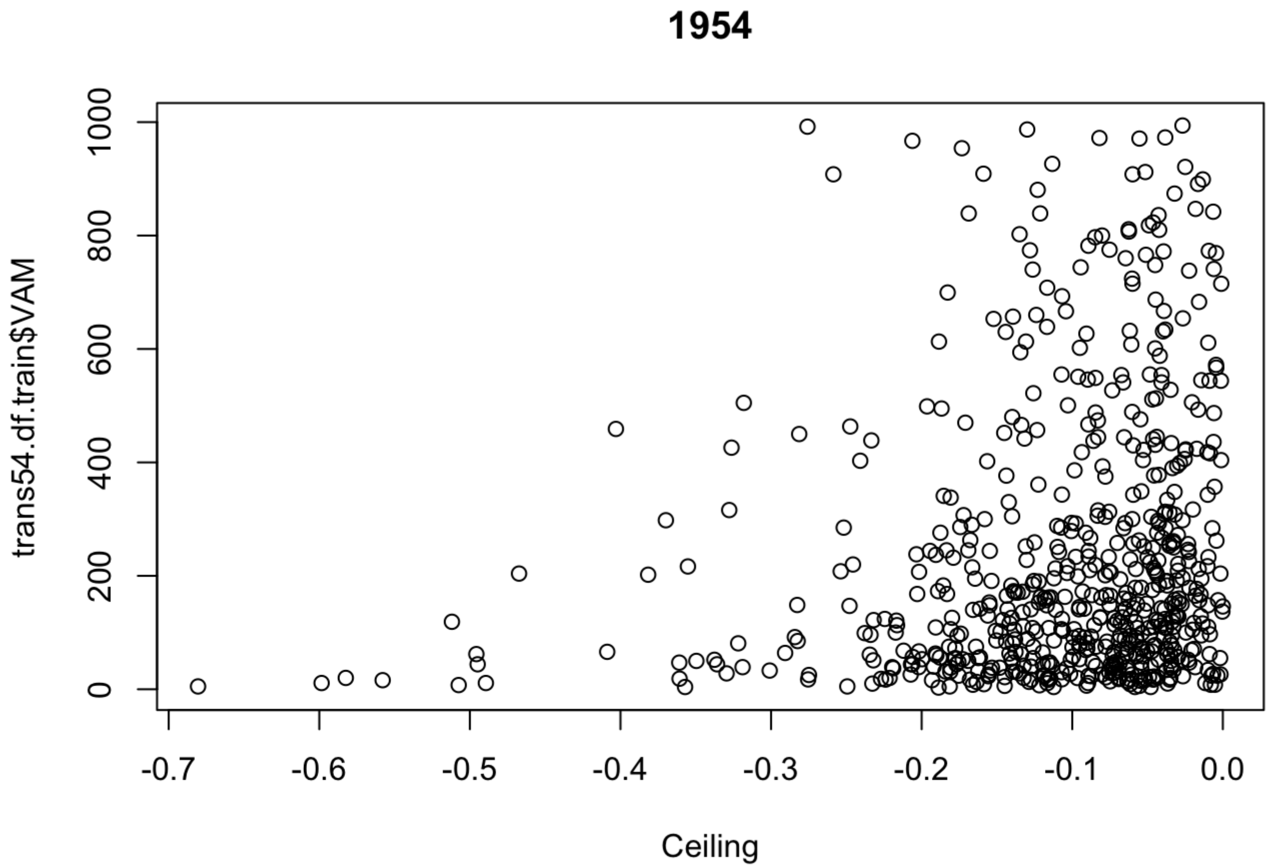
```
##     0  1
## 0 176 54
## 1  17 23
```

1954 Model w/ Transformations

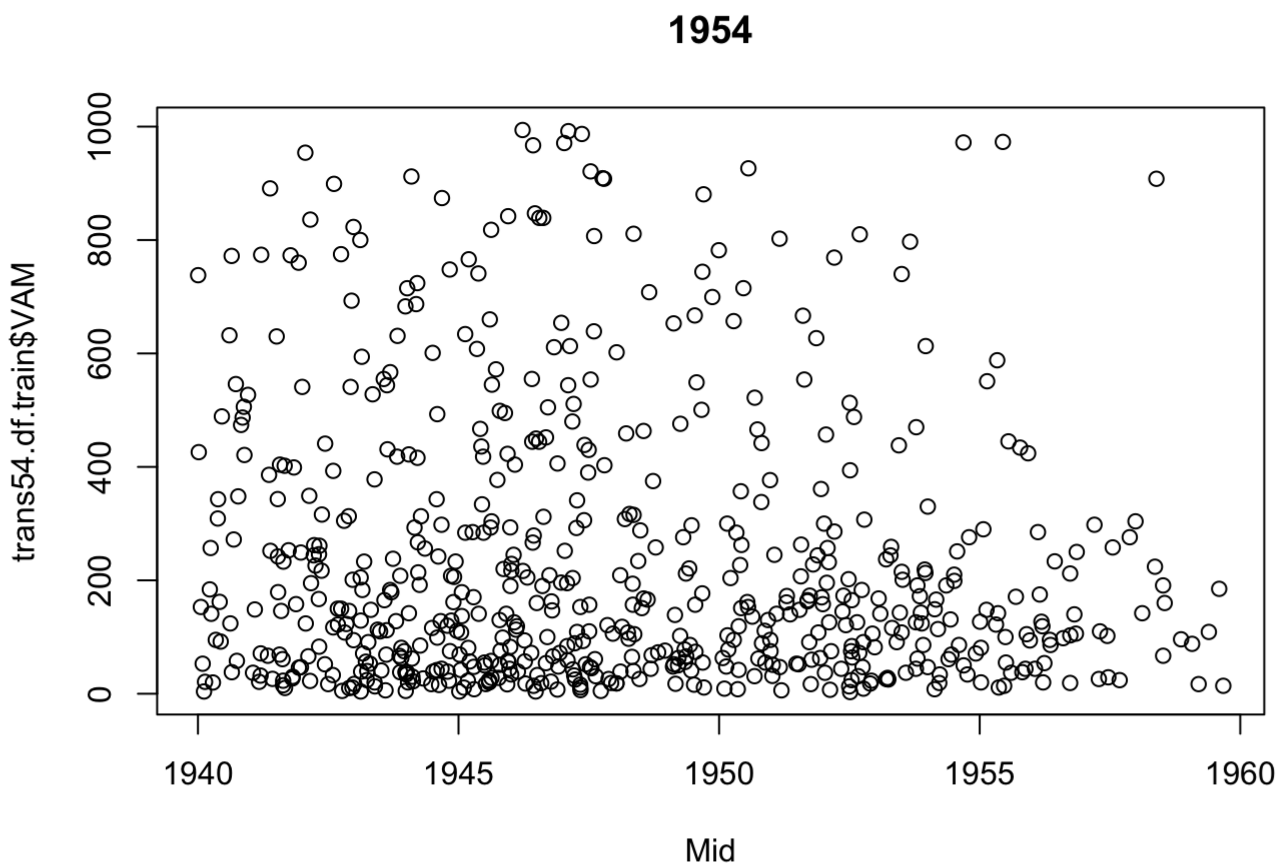
Exploratory plots for predictor features log-transformed

```
trans54.df.train <- df.54.train
trans54.df.test <- df.54.test
trans54.df.train$Ceiling <- log(trans54.df.train$Ceiling)
trans54.df.test$Ceiling <- log(trans54.df.test$Ceiling)
trans54.df.train$Slope <- log(trans54.df.train$Slope)
trans54.df.test$Slope <- log(trans54.df.test$Slope)


#Plots for Transformed Predictors in Year 1954
plot(trans54.df.train$Slope, trans54.df.train$VAM, main = "1954", xlab = "Slope")
```


```
plot(trans54.df.train$Ceiling, trans54.df.train$VAM, main = "1954", xlab = "Ceiling")
```




```
plot(trans54.df.train$Mid, trans54.df.train$VAM, main = "1954", xlab = "Mid")
```



Modeling for transformed predicgtors for year 1954

We see below, same general results, VAM decreases with SLope and Mid but increases with Ceiling

```
model.trans <- glm(response ~ Slope + Ceiling + Mid,  data = trans54.df.train, family=bi
nomial(link="logit"))
summary(model.trans)
```

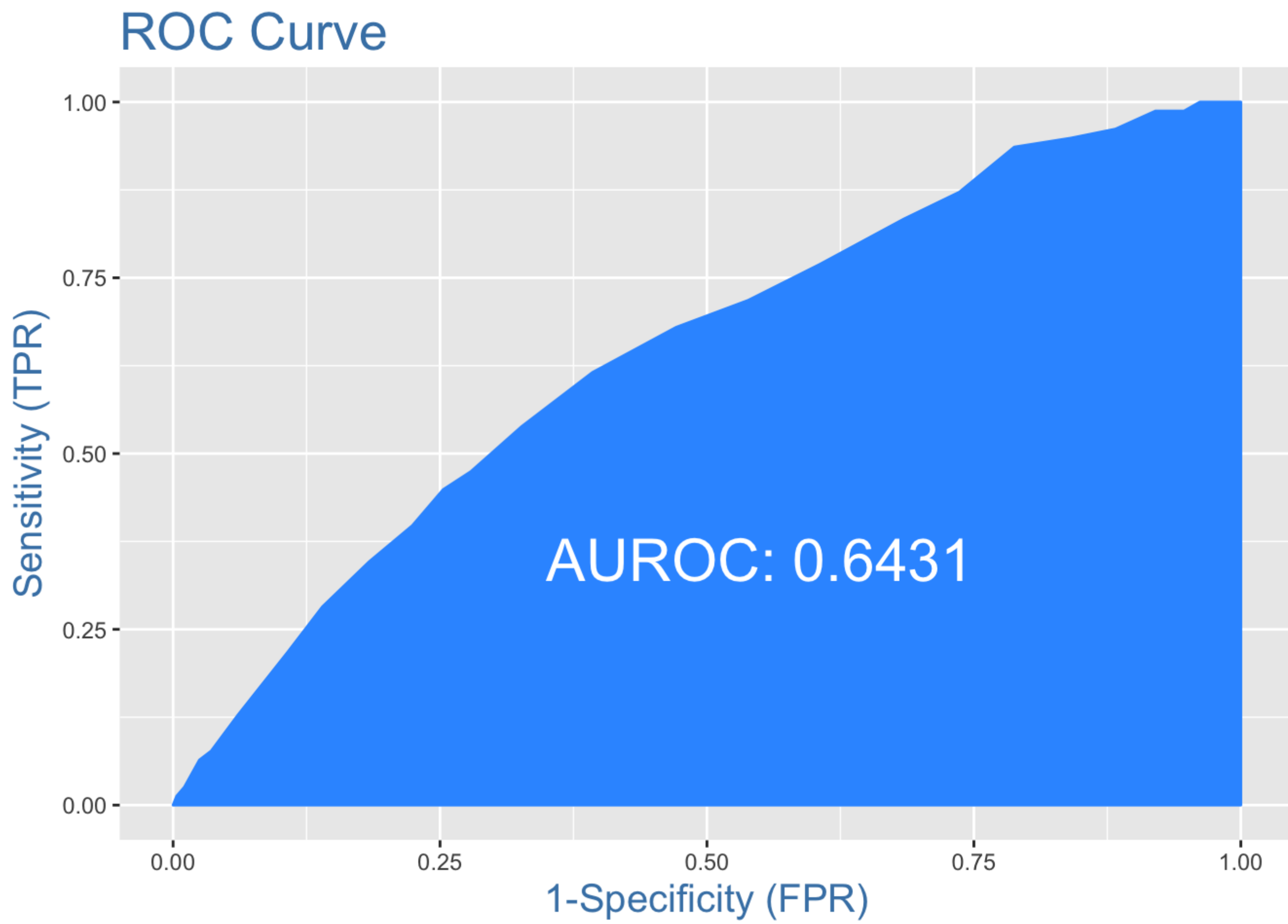
```
##
## Call:
## glm(formula = response ~ Slope + Ceiling + Mid, family = binomial(link = "logit"),
##     data = trans54.df.train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.3122  -0.9168  -0.7177   1.2313   1.9767
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) 70.30520   37.14907   1.893 0.058422 .
## Slope       -1.22026    0.34038  -3.585 0.000337 ***
## Ceiling      2.40608    1.15934   2.075 0.037950 *
## Mid         -0.03745    0.01900  -1.971 0.048677 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 797.78  on 629  degrees of freedom
## Residual deviance: 758.02  on 626  degrees of freedom
## AIC: 766.02
##
## Number of Fisher Scoring iterations: 4
```

Evaluating this transformed model's goodness of fit below

```
predicted.54.trans <- plogis(predict(model.trans, trans54.df.test))

#Find Optimal Prediction cutoff
optCutOff <- optimalCutoff(trans54.df.test$response, predicted.54.trans)

#Misclassification Error
misClassError(trans54.df.test$response, predicted.54.trans, threshold = optCutOff)
```

```
## [1] 0.263
```

```
#ROC Curve
plotROC(trans54.df.test, predicted.54.trans)
```




```
#Sensitivity(Truth Detection Rate)
sensitivity(trans54.df.test$response, predicted.54.trans, threshold = optCutOff)
```

```
## [1] 0.2077922
```

The transformed model above is able to predict `Success` counties 21% of the time so this is worse than the non-transformed model

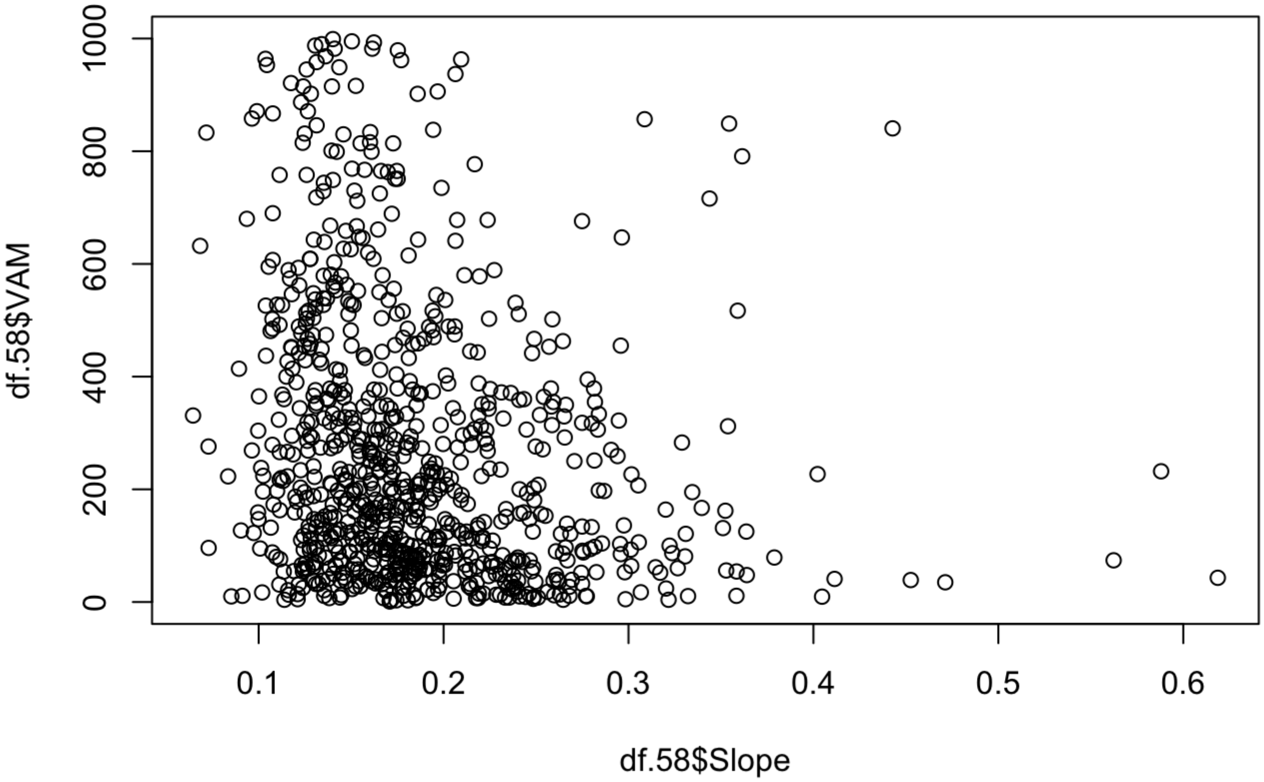
1958

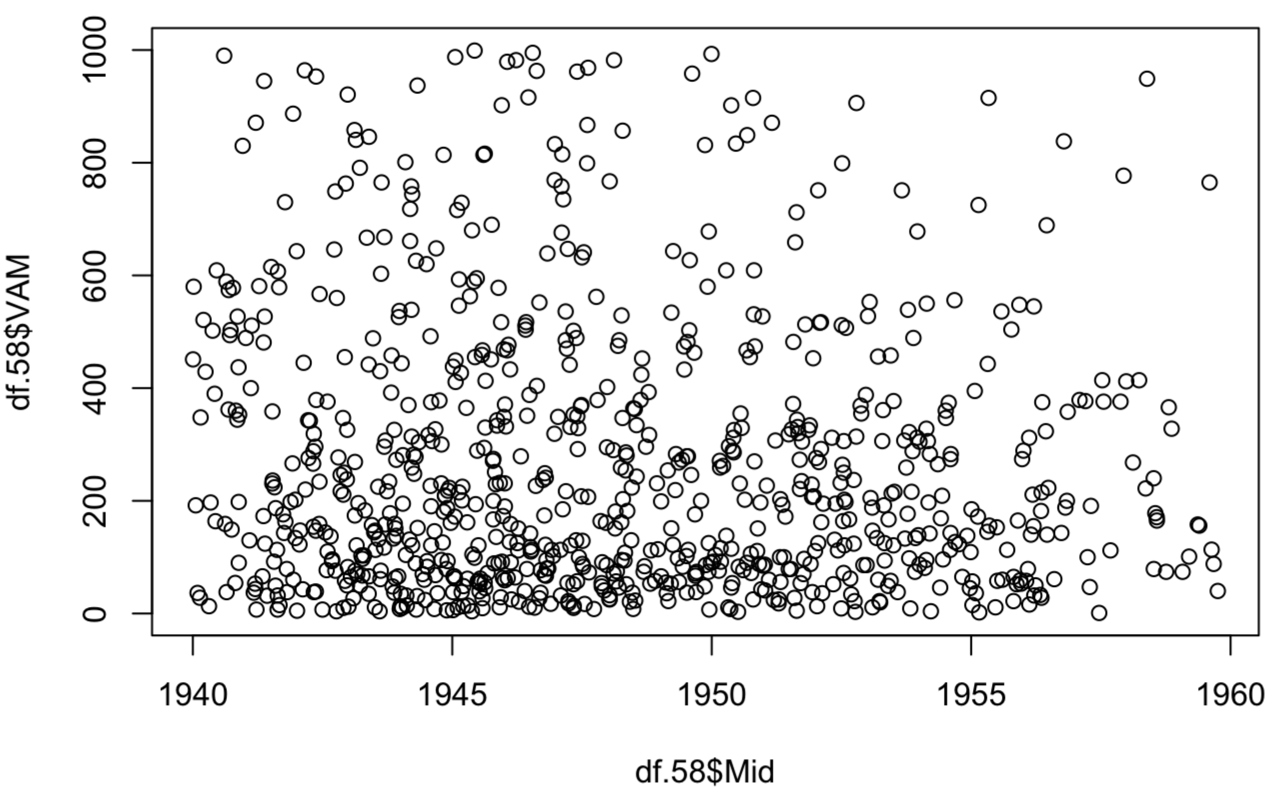
Model for year 1958 below

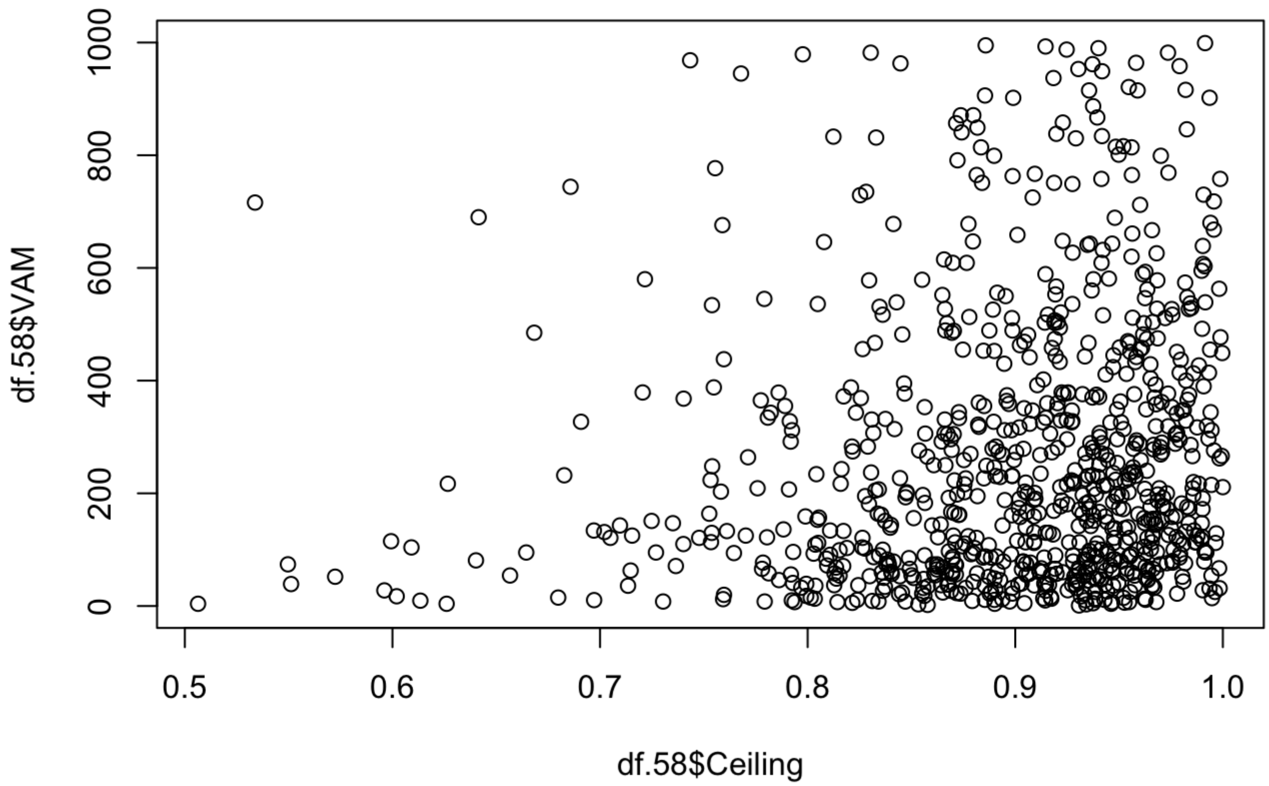
1958 Training and Test Set

```
smp_size <- floor(.7 * nrow(df.58))
set.seed(123)
train_ind <- sample(seq_len(nrow(df.58)), size = smp_size)

df.58.train <- df.58[train_ind,]

df.58.test <- df.58[-train_ind,]
```

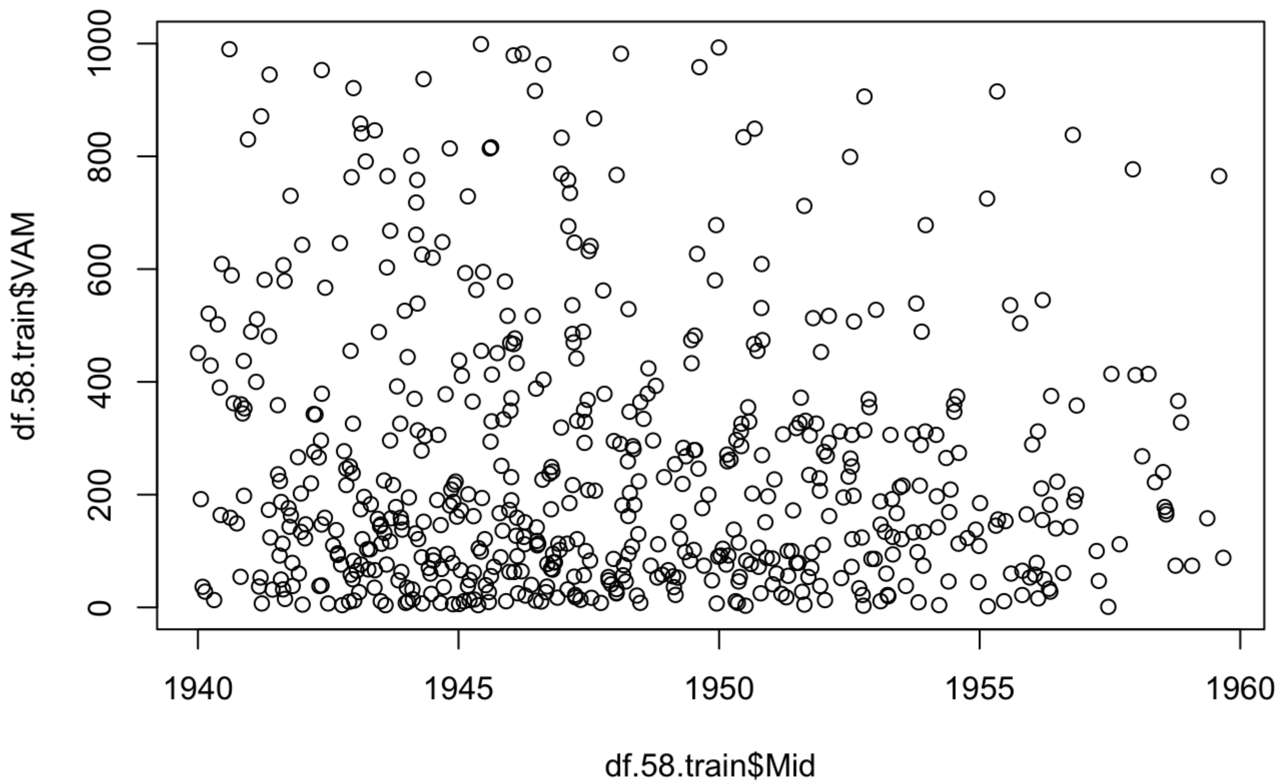
Exploratory plots for year 1958

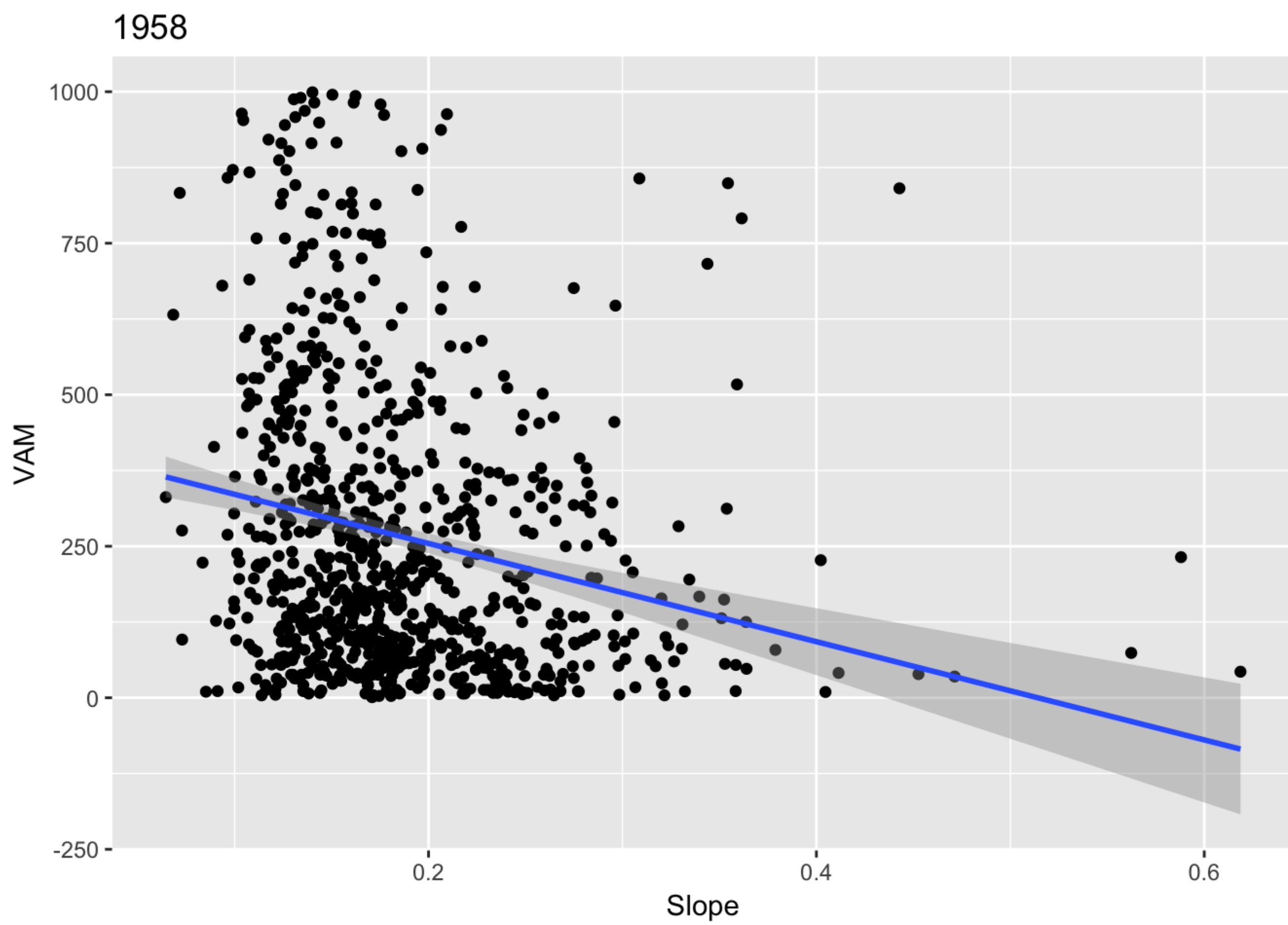

df.58$VAM
0
200
400
600
800
1000
0.1
0.2
0.3
0.4
0.5
0.6
df.58$Slope


df.58$VAM
0
200
400
600
800
1000
1940
1945
1950
1955
1960
df.58$Mid



df.58$VAM
df.58$Ceiling
0
200
400
600
800
1000
0.5
0.6
0.7
0.8
0.9
1.0


df.58.train$VAM
df.58.train$Mid
0
200
400
600
800
1000
1940
1945
1950
1955
1960


1958
VAM
Slope
1000
750
500
250
0
-250
0.2
0.4
0.6

1958 Modeling

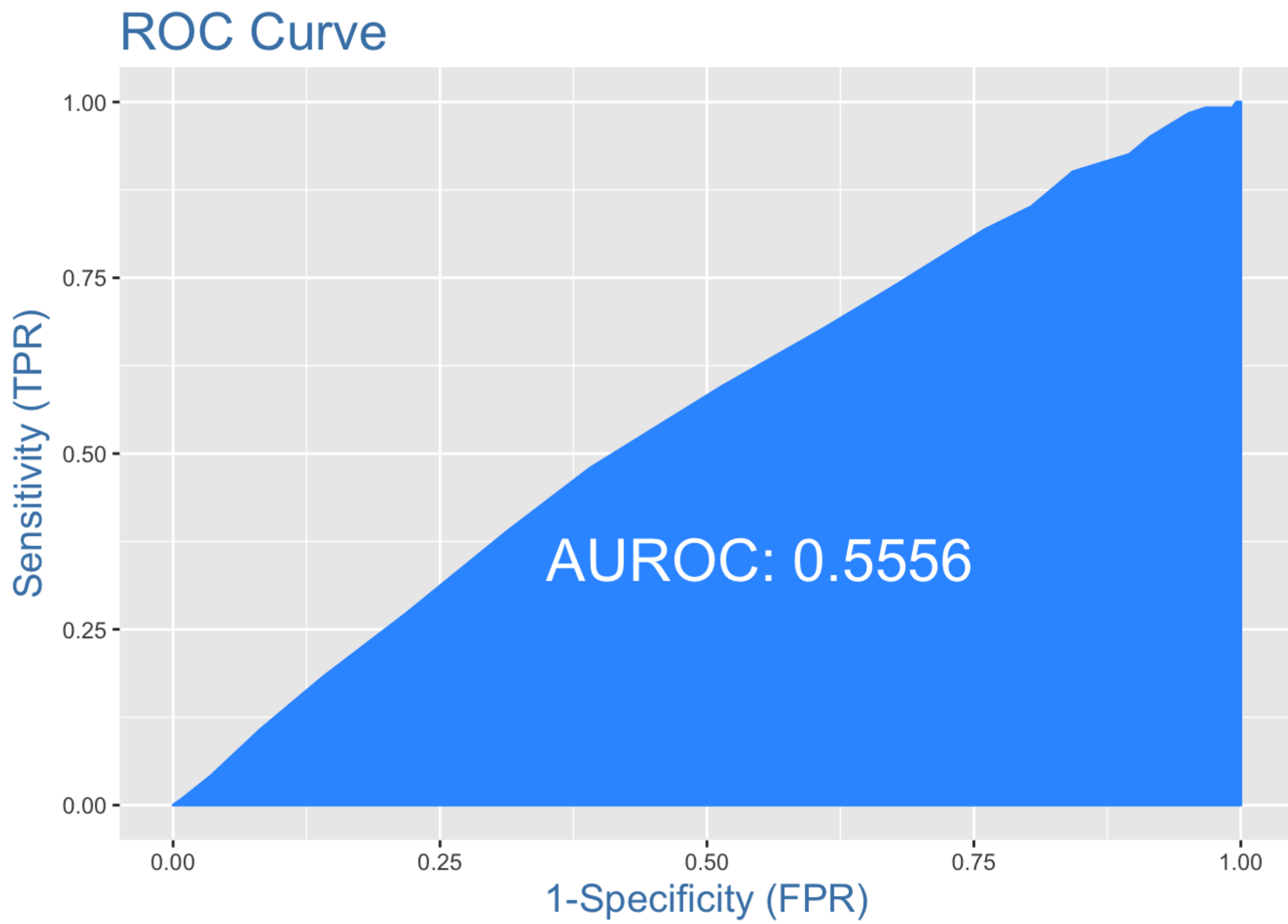
```
model <- glm(response ~ Slope + Ceiling + Mid,  data = df.58.train, family="binomial")
summary(model)
```

```
##
## Call:
## glm(formula = response ~ Slope + Ceiling + Mid, family = "binomial",
##     data = df.58.train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.2438  -1.0184  -0.8056   1.2446   2.0145
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.2190495 35.9641456  -0.090  0.92868
## Slope       -5.1706280  1.6942949  -3.052  0.00227 **
## Ceiling      2.4657006  1.3452594   1.833  0.06682 .
## Mid          0.0007539  0.0184760   0.041  0.96745
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 783.40  on 587  degrees of freedom
## Residual deviance: 758.23  on 584  degrees of freedom
## AIC: 766.23
##
## Number of Fisher Scoring iterations: 4
```

```
predicted.58 <- plogis(predict(model, df.58.test))

#Find Optimal Prediction cutoff
optCutOff <- optimalCutoff(df.58.test$response, predicted.58)

#Misclassification Error
misClassError(df.58.test$response, predicted.58, threshold = optCutOff)
```

```
## [1] 0.4071
```

```
#ROC Curve
plotROC(df.58.test, predicted.58)
```




```
#Sensitivity(Truth Detection Rate)
sensitivity(df.58.test$response, predicted.58, threshold = optCutOff)
```

```
## [1] 0.4710744
```

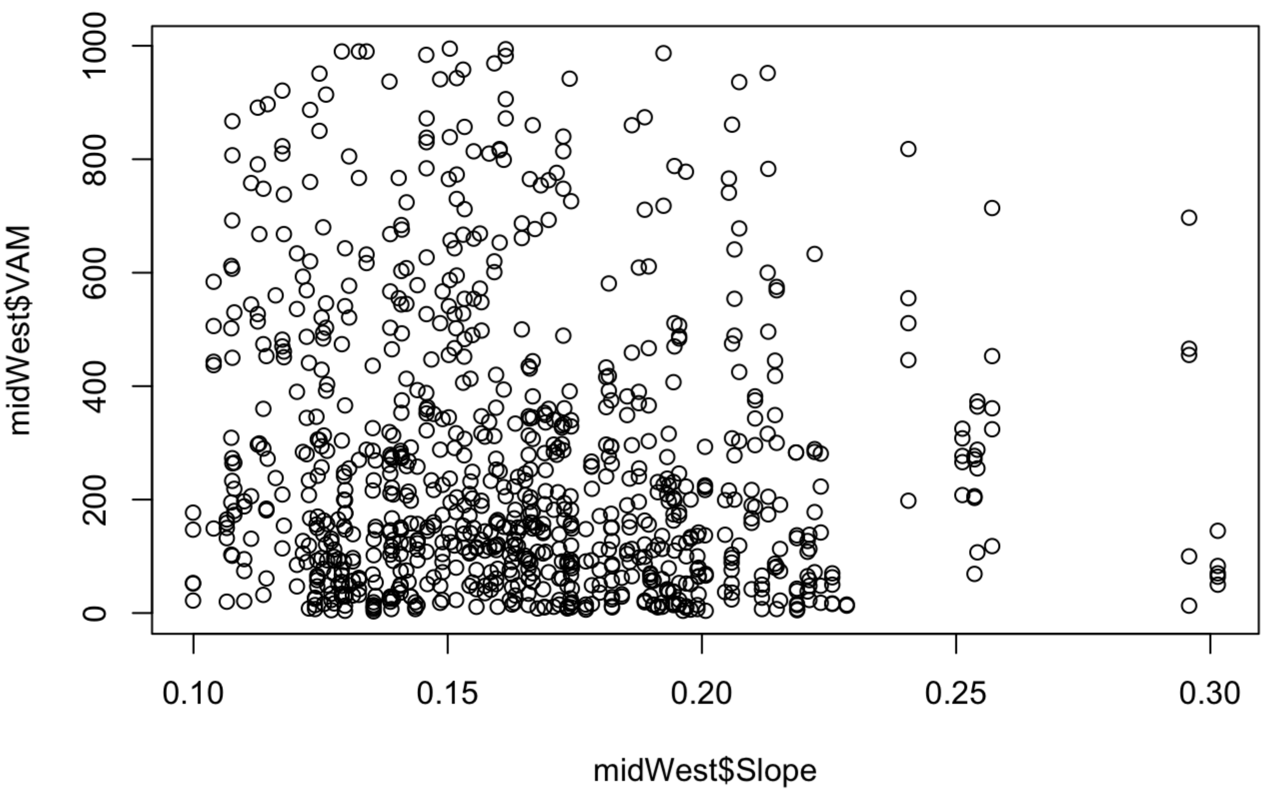
Interestingly enough thoughm the 1958 model is able to predict `Success` counties 47% of the time, this is by far the best.

MidWestern States training and test

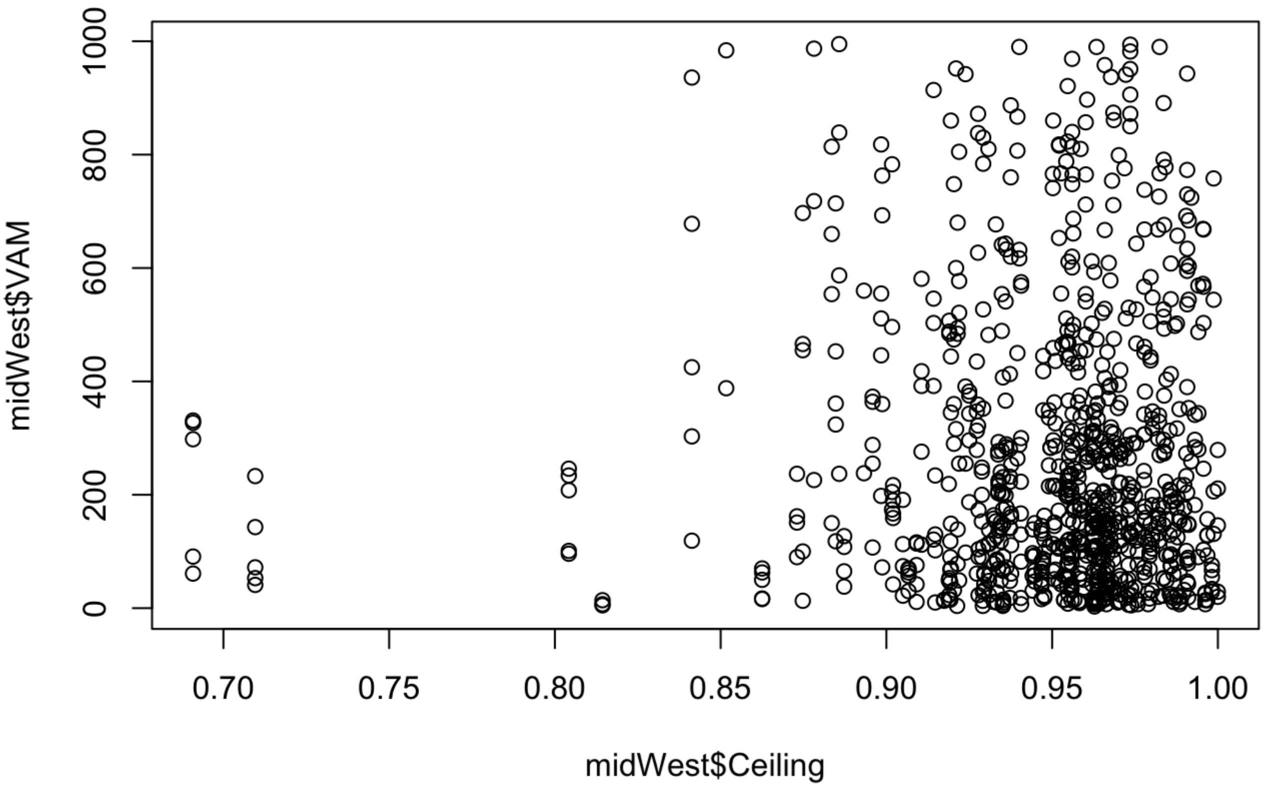
```
#Oh, MI, MN, IL, IN, WI
midWest <- subset(df, State == "OH" | State == "MI" | State == "MN" | State == "IL" | St
ate == "IN" | State == "WI")
smp_size <- floor(.7 * nrow(midWest))
set.seed(123)
train_ind <- sample(seq_len(nrow(midWest)), size = smp_size)

midWest.train <- midWest[train_ind,]

midWest.test <- midWest[-train_ind,]
```

Exploring only midwestern states

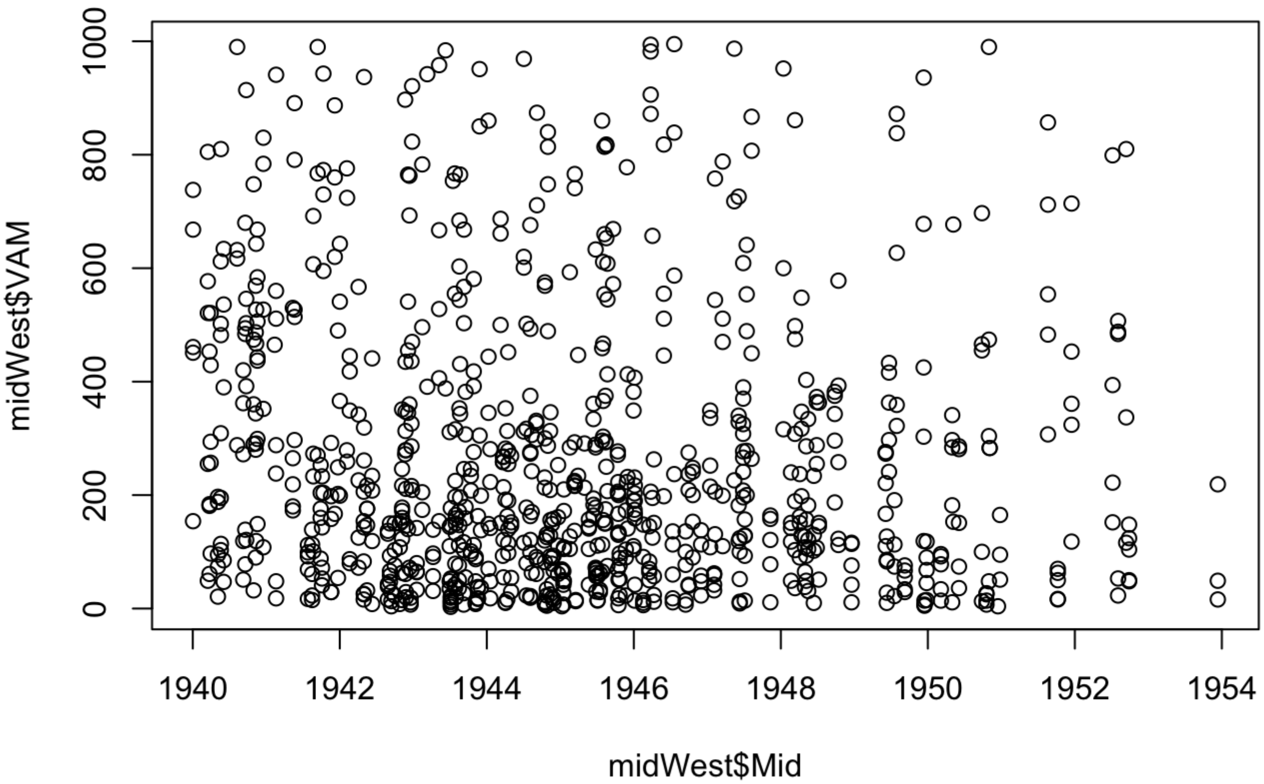
```
#Plots for Transformed Predictors in Year 1954
plot(midWest$Slope, midWest$VAM)
```



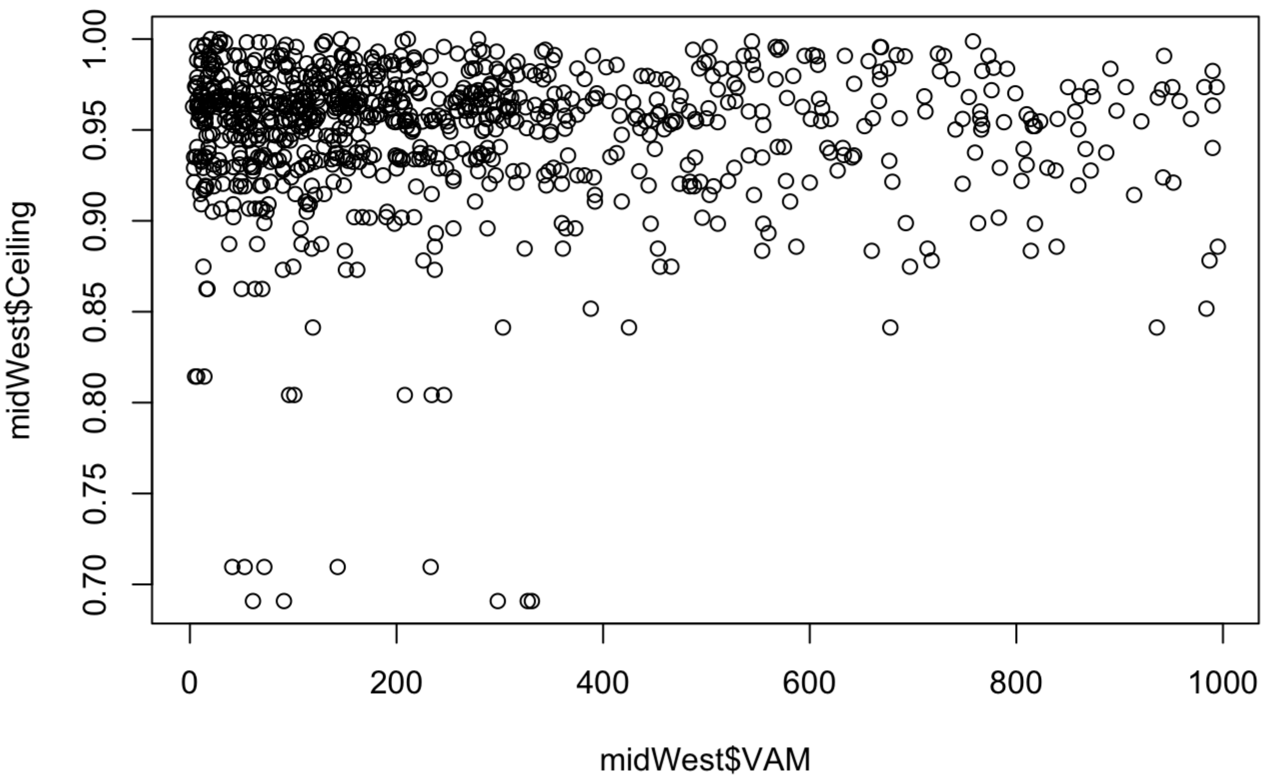

```
plot(midWest$Ceiling, midWest$VAM)
```




```
plot(midWest$Mid, midWest$VAM)
```




```
#Response on X-Axis
plot(midWest$VAM, midWest$Ceiling)
```



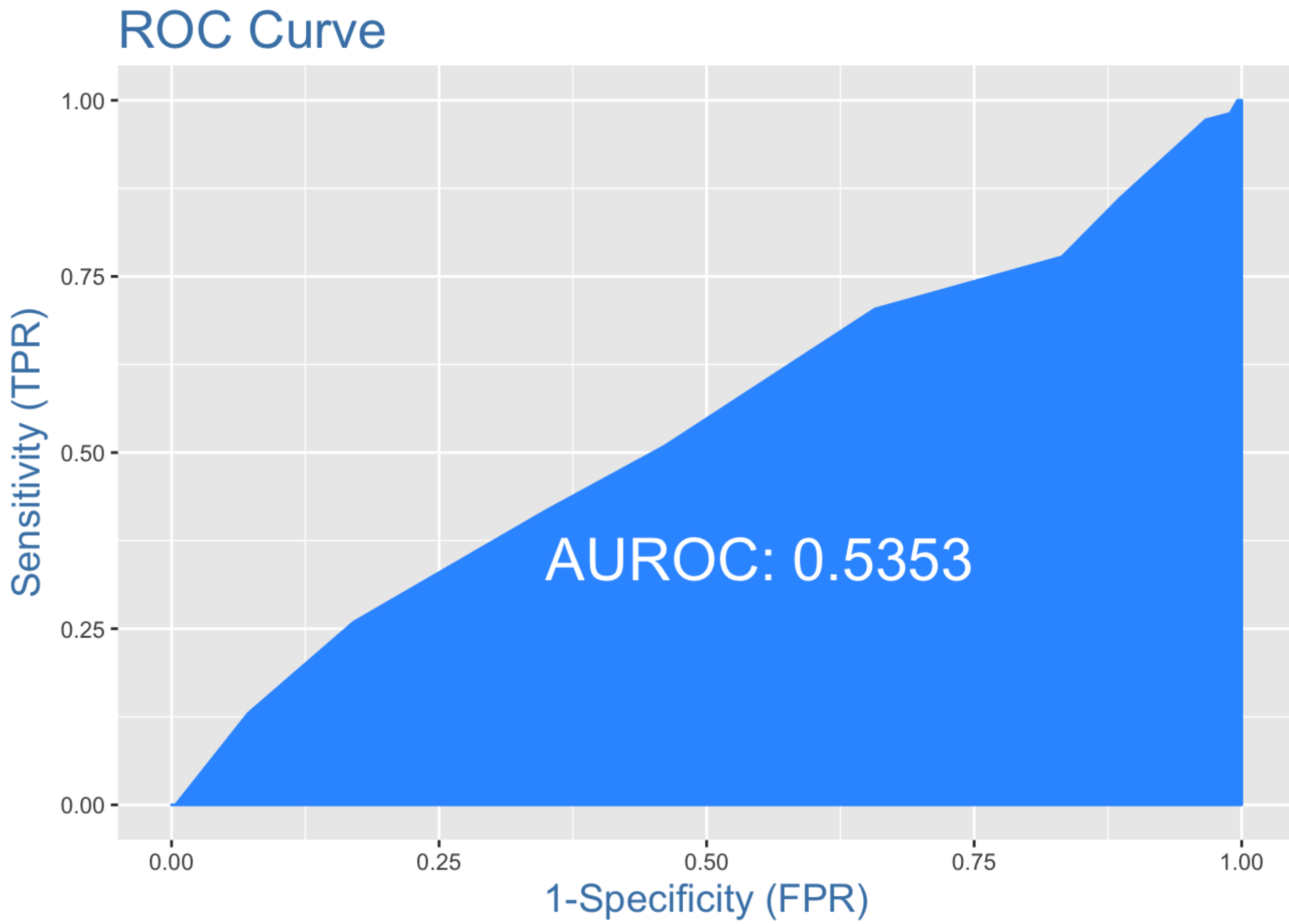

```
model_midWest <- glm(response ~ Slope + Ceiling + Mid,  data = midWest.train, family=bin
omial(link = "logit"))
summary(model_midWest)
```

```
##
## Call:
## glm(formula = response ~ Slope + Ceiling + Mid, family = binomial(link = "logit"),
##     data = midWest.train)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.151  -1.002  -0.938   1.308   1.665
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) 63.78637   55.45625   1.150    0.250
## Slope       -3.08665    2.58071  -1.196    0.232
## Ceiling      1.04119    2.19220   0.475    0.635
## Mid         -0.03326    0.02854  -1.165    0.244
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 840.78  on 626  degrees of freedom
## Residual deviance: 835.06  on 623  degrees of freedom
## AIC: 843.06
##
## Number of Fisher Scoring iterations: 4
```

```
predicted.midWest <- plogis(predict(model_midWest, midWest.test))

#Find Optimal Prediction cutoff
optCutOff <- optimalCutoff(midWest.test$response, predicted.midWest)

#Misclassification Error
misClassError(midWest.test$response, predicted.midWest, threshold = optCutOff)
```

```
## [1] 0.3741
```

```
#ROC Curve
plotROC(midWest.test, predicted.midWest)
```




```
#Sensitivity(Truth Detection Rate)
sensitivity(midWest.test$response, predicted.midWest, threshold = optCutOff)
```

```
## [1] 0.1203704
```

Midwest model is no good, 12% sucess in prediction, however there aren’t many observations so this was to be expected.