InsNova_Auto_Insurance_Claim_Prediction_MLR_Model

Lohit Marla

2023-11-15

```r
InsNova.data <- read.csv("data/InsNova_data_2023_train.csv")
train.data <- InsNova.data
InsNova.val_data <- read.csv("data/InsNova_data_2023_vh.csv")
test.data <- InsNova.val_data

nrow(train.data)
```

```
## [1] 22619
```

```r
nrow(InsNova.data)
```

```
## [1] 22619
```

```r
nrow(test.data)
```

```
## [1] 22620
```

```r
nrow(InsNova.val_data)
```

```
## [1] 22620
```

```r
column_names <- c(
     "gender", "agecat", "engine_type",
    "veh_color", "marital_status", "e_bill", "time_of_week_driven", "high_education_ind", "veh_body"
)

# Convert the selected columns to factors in your data frame
train.data[, column_names] <- lapply(train.data[, column_names], as.factor)
test.data[, column_names] <- lapply(test.data[, column_names], as.factor)
# Check the data frame structure

train.data$clm <- NULL
train.data$id <- NULL
test.data$id <- NULL
train.data$numclaims <- NULL

str(train.data)
```

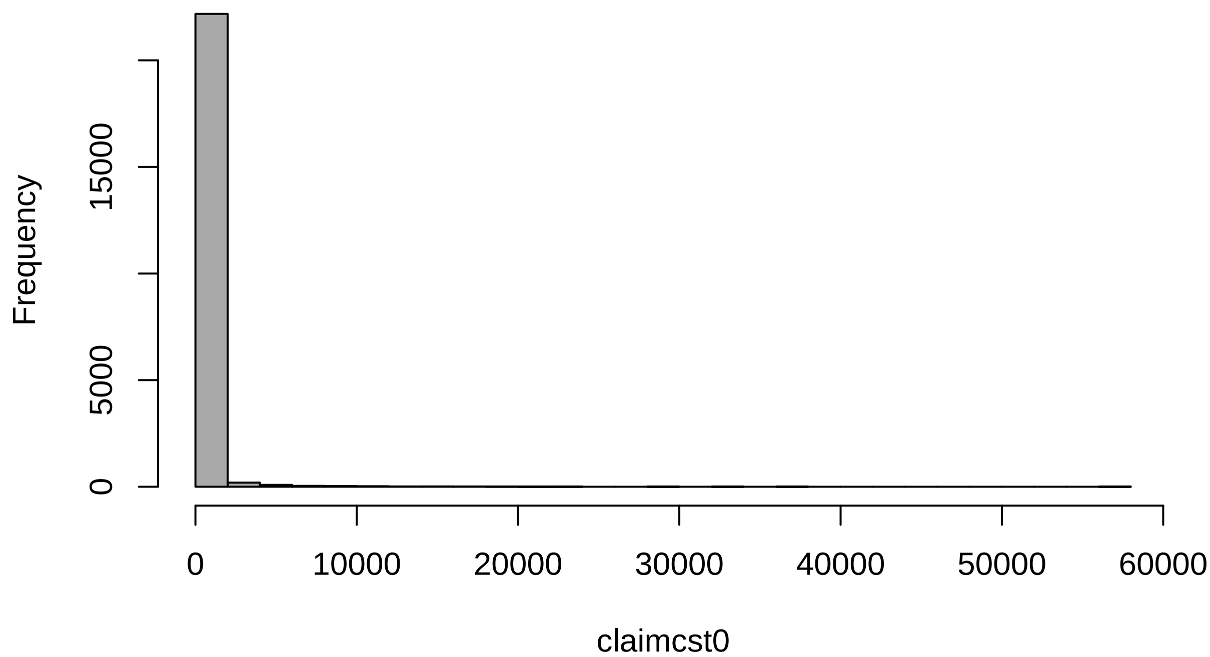
```
## 'data.frame':    22619 obs. of  19 variables:
##  $ veh_value           : num  0.77 4.45 4.9 0.48 0.85 1.37 4.74 0.41 1.41 3.26 ...
##  $ exposure            : num  0.445 0.562 0.465 0.271 0.142 ...
##  $ veh_body            : Factor w/ 13 levels "BUS","CONVT",..: 10 11 11 8 10 10 13 10 10 11 ...
##  $ veh_age             : int  4 1 1 4 4 3 1 4 3 2 ...
##  $ gender              : Factor w/ 2 levels "F","M": 2 2 1 2 1 2 2 2 1 1 ...
##  $ area                : chr  "D" "A" "A" "A" ...
##  $ agecat              : Factor w/ 6 levels "1","2","3","4",..: 3 3 3 4 5 4 2 2 4 2 ...
```

```
##  $ engine_type          : Factor w/ 4 levels "dissel","electric",..: 4 4 4 4 4 2 4 4 4 4 ...
##  $ max_power            : int  147 158 159 80 126 152 232 106 105 100 ...
##  $ driving_history_score: num  67 76 58 72 91 59 61 37 41 99 ...
##  $ veh_color            : Factor w/ 9 levels "black","blue",..: 1 8 1 8 8 8 4 1 1 8 ...
##  $ marital_status       : Factor w/ 2 levels "M","S": 2 2 1 2 2 2 1 1 2 2 ...
##  $ e_bill               : Factor w/ 2 levels "0","1": 2 2 2 2 1 2 2 1 1 2 ...
##  $ time_of_week_driven  : Factor w/ 2 levels "weekday","weekend": 1 1 1 1 1 1 1 1 2 1 ...
##  $ time_driven          : chr  "6pm - 12am" "6am - 12pm" "6pm - 12am" "12pm - 6pm" ...
##  $ trm_len              : int  6 12 6 12 6 6 6 12 12 6 ...
##  $ credit_score         : num  640 684 654 643 647 ...
##  $ high_education_ind   : Factor w/ 2 levels "0","1": 2 1 2 1 1 1 1 2 2 1 ...
##  $ claimcst0            : num  0 0 0 0 0 0 0 0 0 0 ...
```

```r
str(test.data)
```

```
## 'data.frame':    22620 obs. of  18 variables:
##  $ veh_value            : num  3.4 2.55 3.04 2.05 1.93 1.36 1.59 0.84 1.59 4.23 ...
##  $ exposure             : num  0.0763 0.0934 0.1578 0.5607 0.2583 ...
##  $ veh_body             : Factor w/ 13 levels "BUS","CONVT",..: 11 11 11 7 4 13 10 4 10 11 ...
##  $ veh_age              : int  2 2 2 4 2 3 3 4 2 2 ...
##  $ gender               : Factor w/ 2 levels "F","M": 2 1 1 2 2 2 1 2 2 1 ...
##  $ area                 : chr  "B" "A" "E" "C" ...
##  $ agecat               : Factor w/ 6 levels "1","2","3","4",..: 4 3 4 6 4 4 2 2 6 3 ...
##  $ engine_type          : Factor w/ 4 levels "dissel","electric",..: 4 4 4 1 1 4 4 4 3 1 ...
##  $ max_power            : int  174 181 136 164 89 236 178 97 126 143 ...
##  $ driving_history_score: int  83 65 64 82 48 46 59 57 79 56 ...
##  $ veh_color            : Factor w/ 9 levels "black","blue",..: 1 9 8 4 1 1 8 5 8 1 ...
##  $ marital_status       : Factor w/ 2 levels "M","S": 2 1 2 1 2 2 2 1 1 2 ...
##  $ e_bill               : Factor w/ 2 levels "0","1": 2 1 2 2 1 1 2 2 2 2 ...
##  $ time_of_week_driven  : Factor w/ 2 levels "weekday","weekend": 1 1 1 1 1 1 1 1 1 1 ...
##  $ time_driven          : chr  "6pm - 12am" "12am - 6 am" "12pm - 6pm" "6am - 12pm" ...
##  $ trm_len              : int  6 12 12 12 12 12 6 6 12 6 ...
##  $ credit_score         : num  648 638 661 648 640 ...
##  $ high_education_ind   : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
```

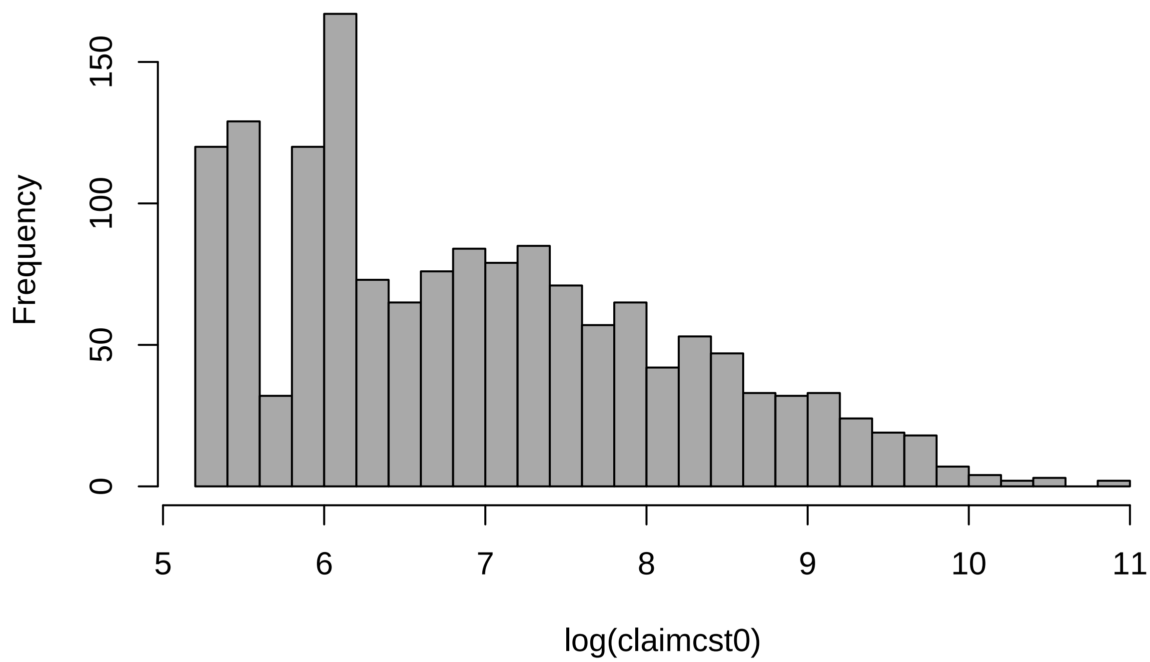
```r
library(e1071)

hist( (train.data$claimcst0), breaks=30, main="", xlab="claimcst0", col= "darkgrey")
```



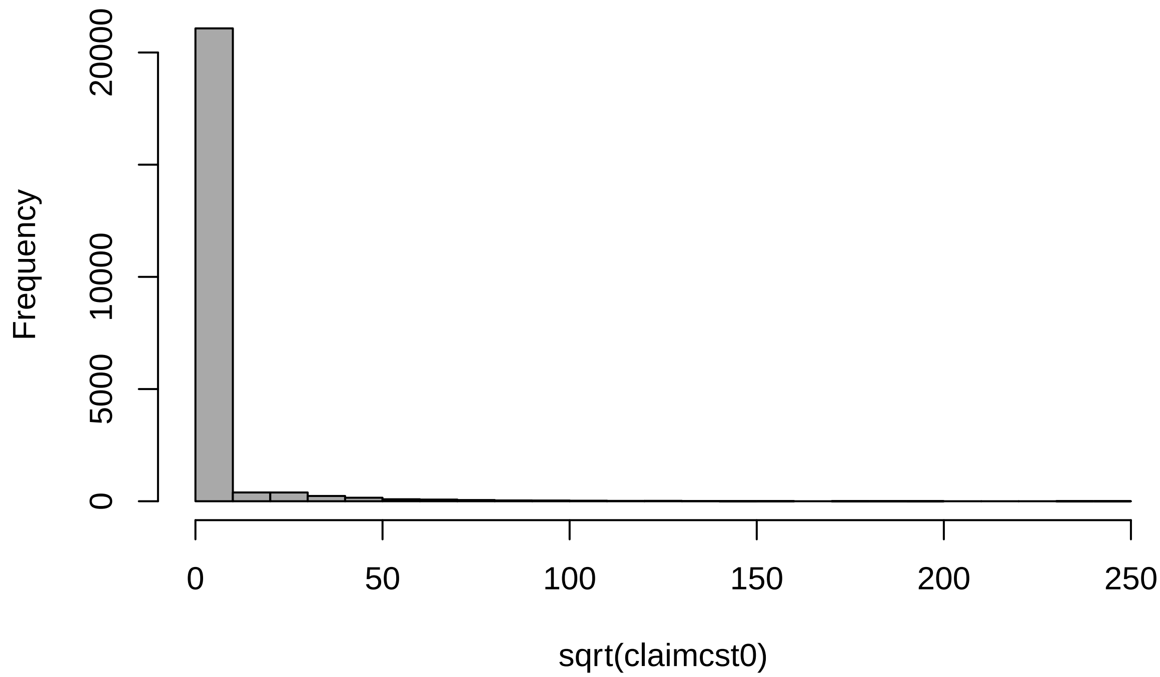

```r
summary(train.data$claimcst0)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0       0       0     163       0   57896
```

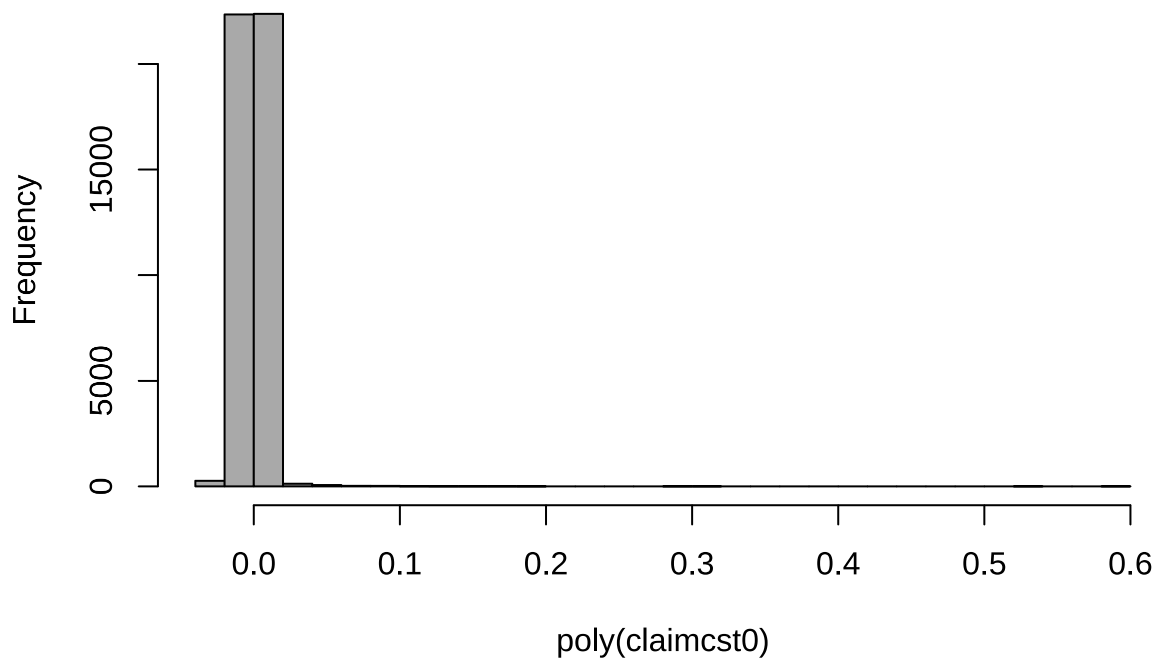
```r
hist( log(train.data$claimcst0), breaks=30, main="", xlab="log(claimcst0)", col= "darkgrey")
```




```r
hist( sqrt(train.data$claimcst0), breaks=30, main="", xlab="sqrt(claimcst0)", col= "darkgrey")
```




```r
hist( poly(train.data$claimcst0, 2), breaks=30, main="", xlab="poly(claimcst0)", col= "darkgrey")
```

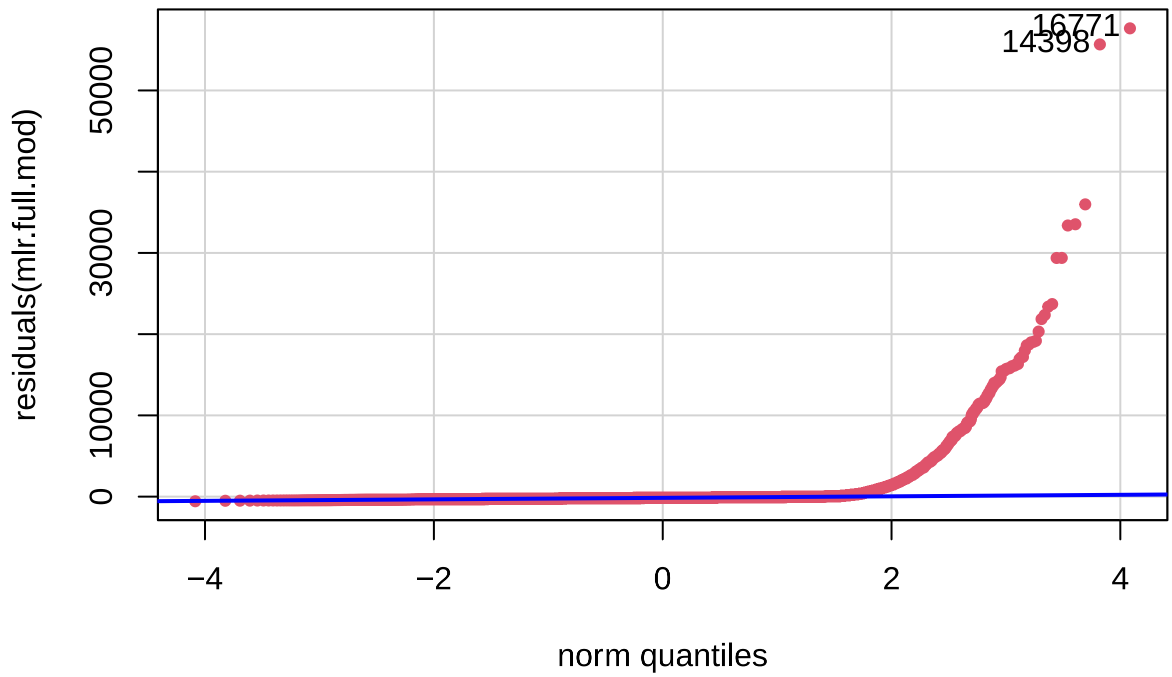



```r
mlr.full.mod <- lm(claimcst0 ~ . , data = train.data)
summary(mlr.full.mod)
```

```
##
## Call:
## lm(formula = claimcst0 ~ ., data = train.data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##   -568   -212   -149    -87  57644
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                171.62717  610.89419   0.281  0.77876
## veh_value                    3.92882   10.19882   0.385  0.70008
## exposure                   226.52663   41.48387   5.461 4.80e-08 ***
## veh_bodyCONVT              -62.01473  383.16812  -0.162  0.87143
## veh_bodyCOUPE              230.95025  302.43506   0.764  0.44509
## veh_bodyHBACK              129.26252  297.13419   0.435  0.66354
## veh_bodyHDTOP               94.71973  297.61747   0.318  0.75029
## veh_bodyMCARA              209.42389  357.25623   0.586  0.55775
## veh_bodyMIBUS              156.26527  303.14960   0.515  0.60623
## veh_bodyPANVN               73.87816  307.97034   0.240  0.81042
## veh_bodyRDSTR              106.19945  515.59199   0.206  0.83681
## veh_bodySEDAN              133.64145  295.29265   0.453  0.65086
## veh_bodySTNWG              141.55740  295.14718   0.480  0.63150
```

```
## veh_bodyTRUCK              171.94401  296.47548   0.580  0.56195
## veh_bodyUTE                118.55497  294.55966   0.402  0.68733
## veh_age                      9.48403   10.76764   0.881  0.37844
## genderM                     26.78876   17.87703   1.499  0.13402
## areaB                        8.68735   25.70658   0.338  0.73541
## areaC                       31.61890   23.03537   1.373  0.16988
## areaD                       22.95740   30.50677   0.753  0.45174
## areaE                        3.55730   34.02466   0.105  0.91673
## areaF                      106.97477   41.84551   2.556  0.01058 *
## agecat2                    -88.95274   35.35423  -2.516  0.01187 *
## agecat3                    -97.27245   34.53729  -2.816  0.00486 **
## agecat4                    -85.70573   34.39117  -2.492  0.01271 *
## agecat5                   -166.78268   36.64673  -4.551 5.36e-06 ***
## agecat6                   -109.09460   40.65250  -2.684  0.00729 **
## engine_typeelectric         33.94124   36.98452   0.918  0.35878
## engine_typehybrid           34.19120   34.59694   0.988  0.32303
## engine_typepetrol           33.11696   23.23494   1.425  0.15408
## max_power                    0.08621    0.27597   0.312  0.75475
## driving_history_score        0.84119    0.44311   1.898  0.05766 .
## veh_colorblue              -35.80105   34.32709  -1.043  0.29699
## veh_colorbrown               7.34995   42.26492   0.174  0.86194
## veh_colorgray               18.74746   26.64722   0.704  0.48172
## veh_colorgreen             -43.03595   39.98577  -1.076  0.28181
## veh_colorred                17.48322   39.49275   0.443  0.65799
## veh_colorsilver            -11.77303   35.50630  -0.332  0.74021
## veh_colorwhite              -3.36337   26.50235  -0.127  0.89901
## veh_coloryellow             16.57037   44.90878   0.369  0.71215
## marital_statusS            -26.38471   16.99906  -1.552  0.12065
## e_bill1                     -2.53753   17.84235  -0.142  0.88691
## time_of_week_drivenweekend  46.94980   21.15956   2.219  0.02651 *
## time_driven12pm - 6pm      -17.31900   40.08717  -0.432  0.66572
## time_driven6am - 12pm       -7.36786   40.11014  -0.184  0.85426
## time_driven6pm - 12am      -12.89261   43.66031  -0.295  0.76777
## trm_len                    -10.00309    3.68560  -2.714  0.00665 **
## credit_score                -0.28434    0.80714  -0.352  0.72463
## high_education_ind1         -1.93757   32.11506  -0.060  0.95189
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1270 on 22570 degrees of freedom
## Multiple R-squared:  0.004498,   Adjusted R-squared:  0.002381
## F-statistic: 2.124 on 48 and 22570 DF,  p-value: 9.567e-06
```

```r
car::qqPlot(residuals(mlr.full.mod), main = NA, pch = 19, col = 2, cex = 0.7)
```

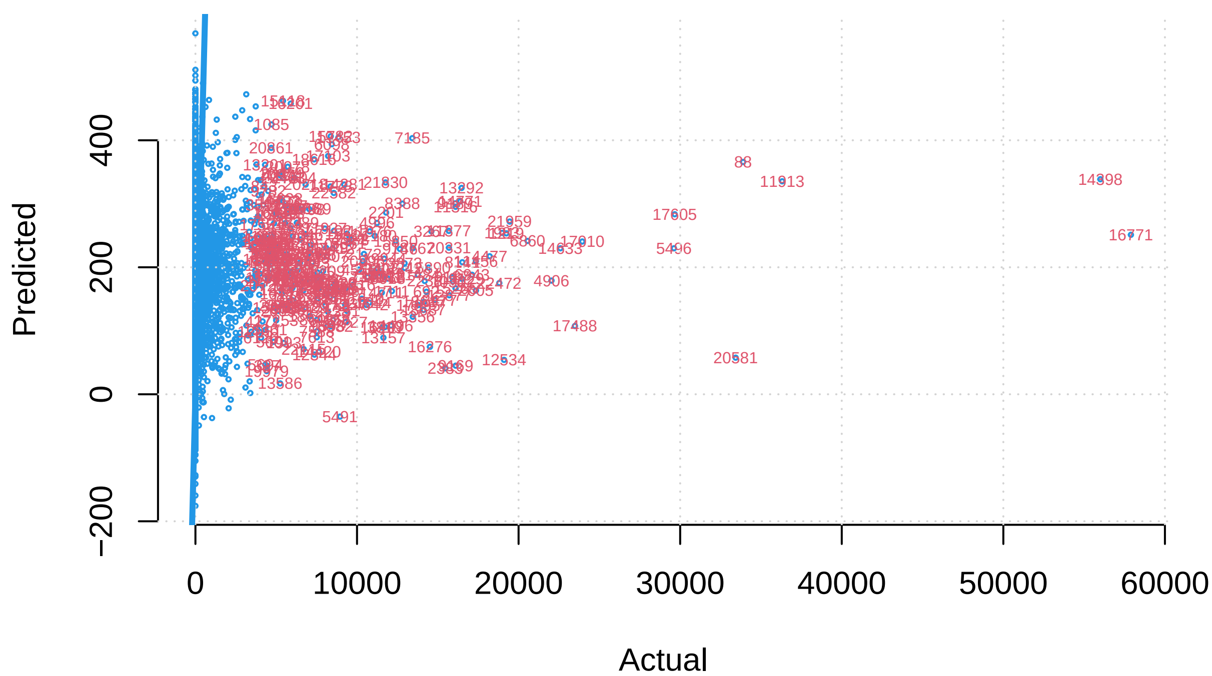



```
## [1] 16771 14398
```

```
SumModelGini <- function(actuals, predictions) {
  df = data.frame(actuals = actuals, predictions = predictions)
  df <- df[order(df$predictions, decreasing = TRUE),]
  df$random = (1:nrow(df))/nrow(df)
  totalPos <- sum(df$actuals)
  df$cumPosFound <- cumsum(df$actuals) # this will store the cumulative number of positive examples fou
  df$Lorentz <- df$cumPosFound / totalPos # this will store the cumulative proportion of positive examp
  df$Gini <- df$Lorentz - df$random # will store Lorentz minus random
  return(sum(df$Gini))
}

NormalizedGini <- function(actuals, predictions) {
  SumModelGini(actuals, predictions) / SumModelGini(actuals, actuals)
}

InsNova.data$id <- NULL
InsNova.data$clm <- NULL
InsNova.data$numclaims <- NULL

InsNova.data[, column_names] <- lapply(InsNova.data[, column_names], as.factor)

mlr.train.claimcst0 <- predict(mlr.full.mod, newdata = InsNova.data, type = "response")

NormalizedGini(mlr.train.claimcst0, train.data$claimcst0 )
```

```
## [1] 0.04291969
```

```r
plot(train.data$claimcst0, predict(mlr.full.mod,newdata = train.data),
     col=4, cex=0.3, xlab="Actual", ylab="Predicted", axes=FALSE)
extpts <- which(abs(residuals(mlr.full.mod)) > 3*sd(residuals(mlr.full.mod)))
text(train.data$claimcst0[extpts],
     predict(mlr.full.mod,newdata = train.data)[extpts],
     rownames(train.data)[extpts], cex=0.5, col=2)
axis(1); axis(2); grid(); abline(0,1, col=4, lwd=3)
```




```r
#Variable inflation factor

car::vif(mlr.full.mod)
```

```
##                           GVIF Df GVIF^(1/(2*Df))
## veh_value             2.382901  1        1.543665
## exposure              1.796002  1        1.340150
## veh_body              5.839956 12        1.076301
## veh_age               1.862105  1        1.364590
## gender                1.098828  1        1.048250
## area                  1.121578  5        1.011540
## agecat                1.077085  5        1.007453
## engine_type           1.188567  3        1.029210
## max_power             2.867577  1        1.693392
## driving_history_score 1.002010  1        1.001005
## veh_color             1.014390  8        1.000893
## marital_status        1.002976  1        1.001487
```

```
## e_bill                 1.031841  1        1.015796
## time_of_week_driven    1.002285  1        1.001142
## time_driven            1.006252  3        1.001039
## trm_len                1.288464  1        1.135105
## credit_score           1.009476  1        1.004727
## high_education_ind     1.494499  1        1.222497
```

```r
cond_num <- round(max(car::vif(mlr.full.mod))  / min(car::vif(mlr.full.mod))   , 0)

cond_num
```

```
## [1] 12
```