Auto-insurance Claim Prediction

XI y Hes

1. Overview

This dataset has 9,134 entries of customers from an anonymous auto insurance company with information on their demography (location type, education, employment status, gender, income, marital status) and their auto insurance plan (claim amount, monthly premium, months since last claim, months since policy inception, number of complaints, number of policies, policy type, claim reasons, vehicle class, vehicle size).

The goal for this analysis is to find how this information can be used to predict the insurance claim, which can be helpful and applicable for insurance companies to identify riskier customers and thus to customize suitable auto insurance plans.

Set up

2. Preliminary Analysis

Sneak peak into the data

```
dim(autoinsurance)
```

```
## [1] 9134   26
```

```
colnames(autoinsurance)
```

```
##  [1] "CUSTOMER"                      "COUNTRY"
##  [3] "STATE_CODE"                    "STATE"
##  [5] "CLAIM_AMOUNT"                  "RESPONSE"
##  [7] "COVERAGE"                      "EDUCATION"
##  [9] "EFFECTIVE_TO_DATE"             "EMPLOYMENT"
## [11] "GENDER"                        "INCOME"
## [13] "LOCATION_CODE"                 "MARITAL_STATUS"
## [15] "MONTHLY_PREMIUM"               "MONTHS_SINCE_LAST_CLAIM"
## [17] "MONTHS_SINCE_POLICY_INCEPTION" "NUMBER_COMPLAINTS"
## [19] "NUMBER_POLICIES"               "POLICY_TYPE"
## [21] "POLICY"                        "CLAIM_REASON"
## [23] "SALES_CHANNEL"                 "TOTAL_CLAIM"
## [25] "VEHICLE_CLASS"                 "VEHICLE_SIZE"
```

```
#There are 26 variables.

sum(is.na(autoinsurance))
```

```
## [1] 0
```

```
# This dataset is all filled!
```

The dataset contains 9,134 cases with 26 variables as listed below. There is no missing value in the dataset.

```
#Reduce columns
autoinsurance <- autoinsurance[c(4,7,8,10,11,12,13,14,15,16,17,18,19,20,22,24,25,26)]
```

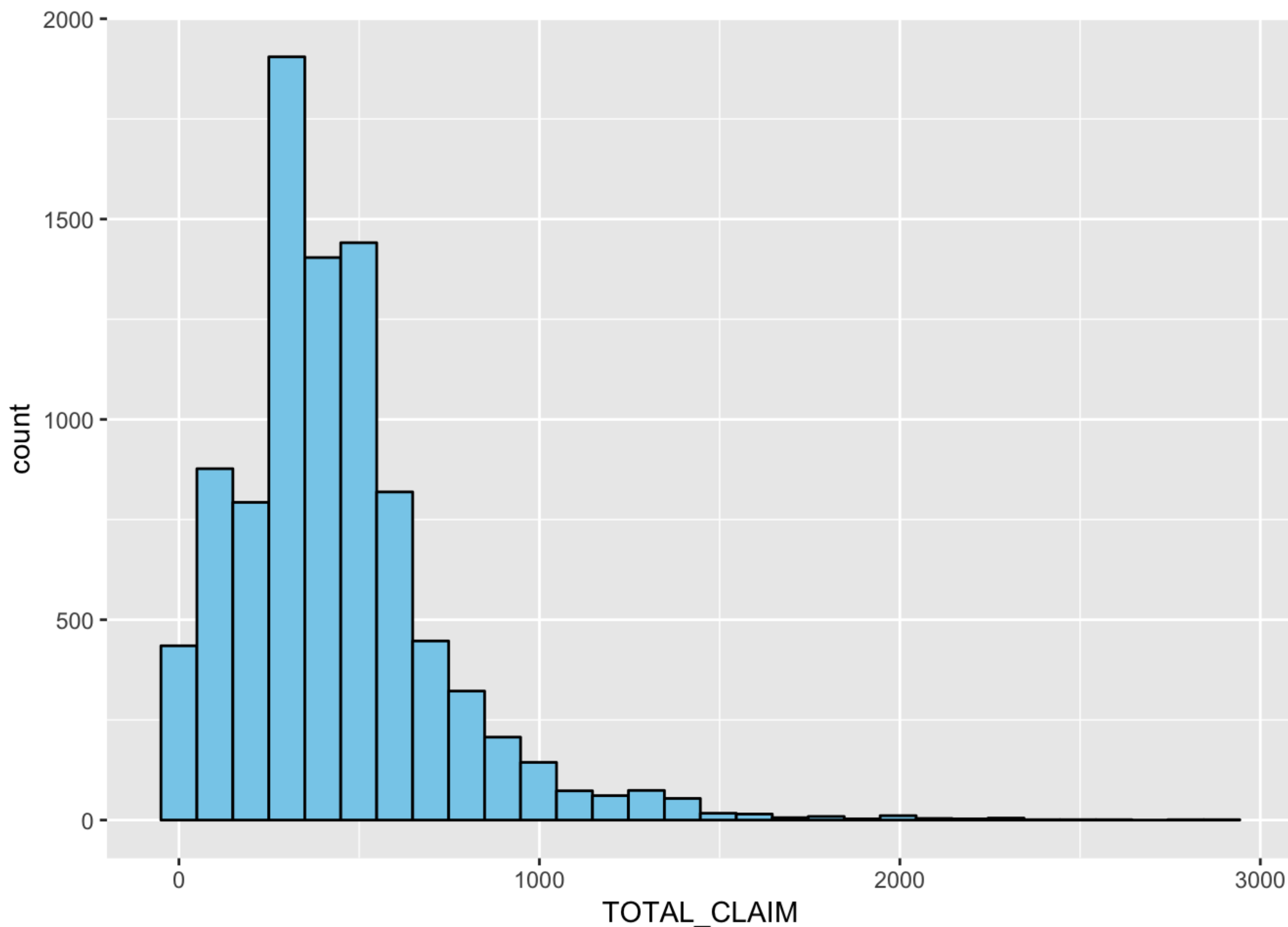
We are interested in the “Total Claim variable”

```
summary(autoinsurance$TOTAL_CLAIM)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##    0.099  272.258  383.945  434.089  547.515 2893.240
```

```
ggplot(autoinsurance, aes(x=TOTAL_CLAIM)) + geom_histogram(color="black", fill="sky blu
e")
```

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

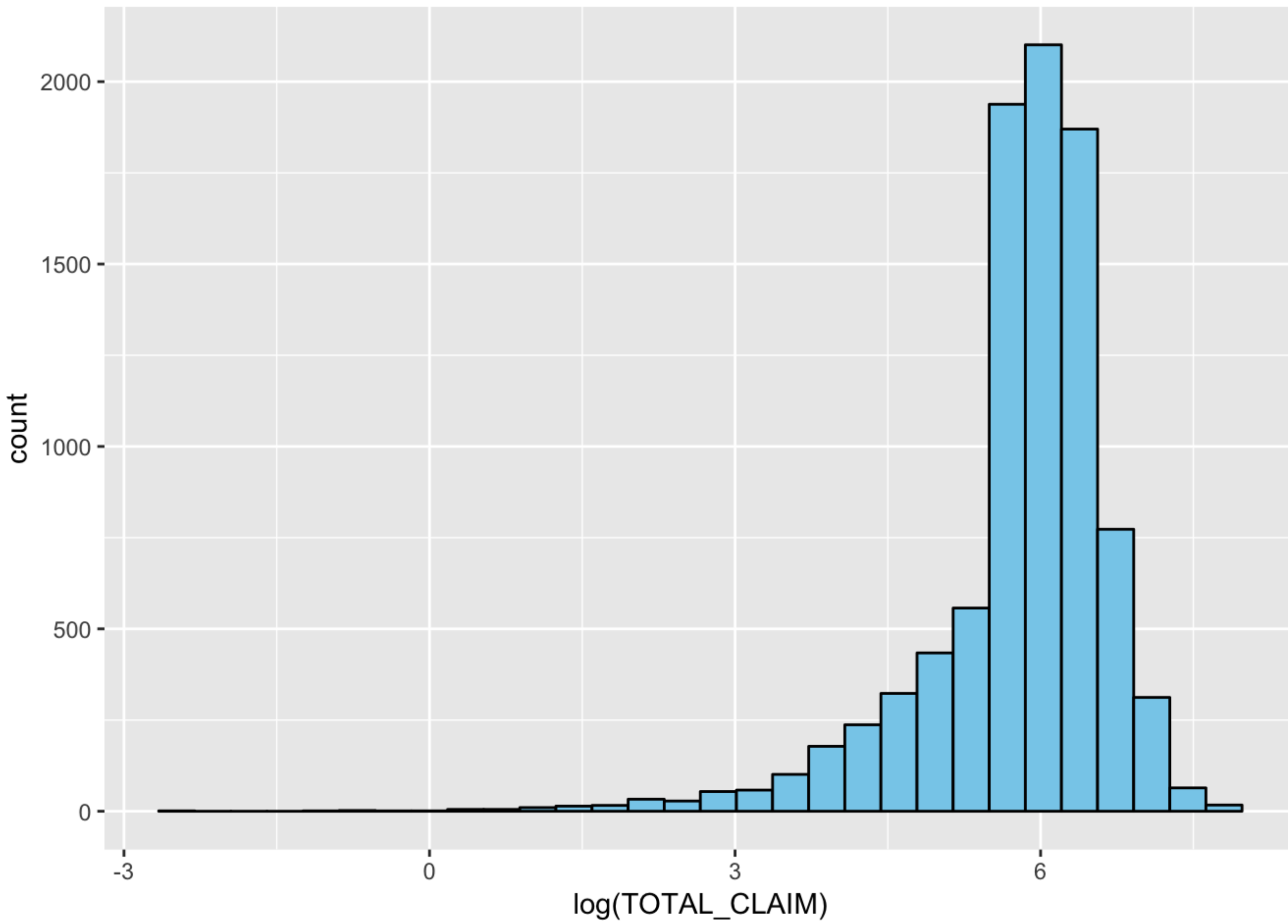



```
library(e1071)

skewness(autoinsurance$TOTAL_CLAIM)
```

```
## [1] 1.714403
```

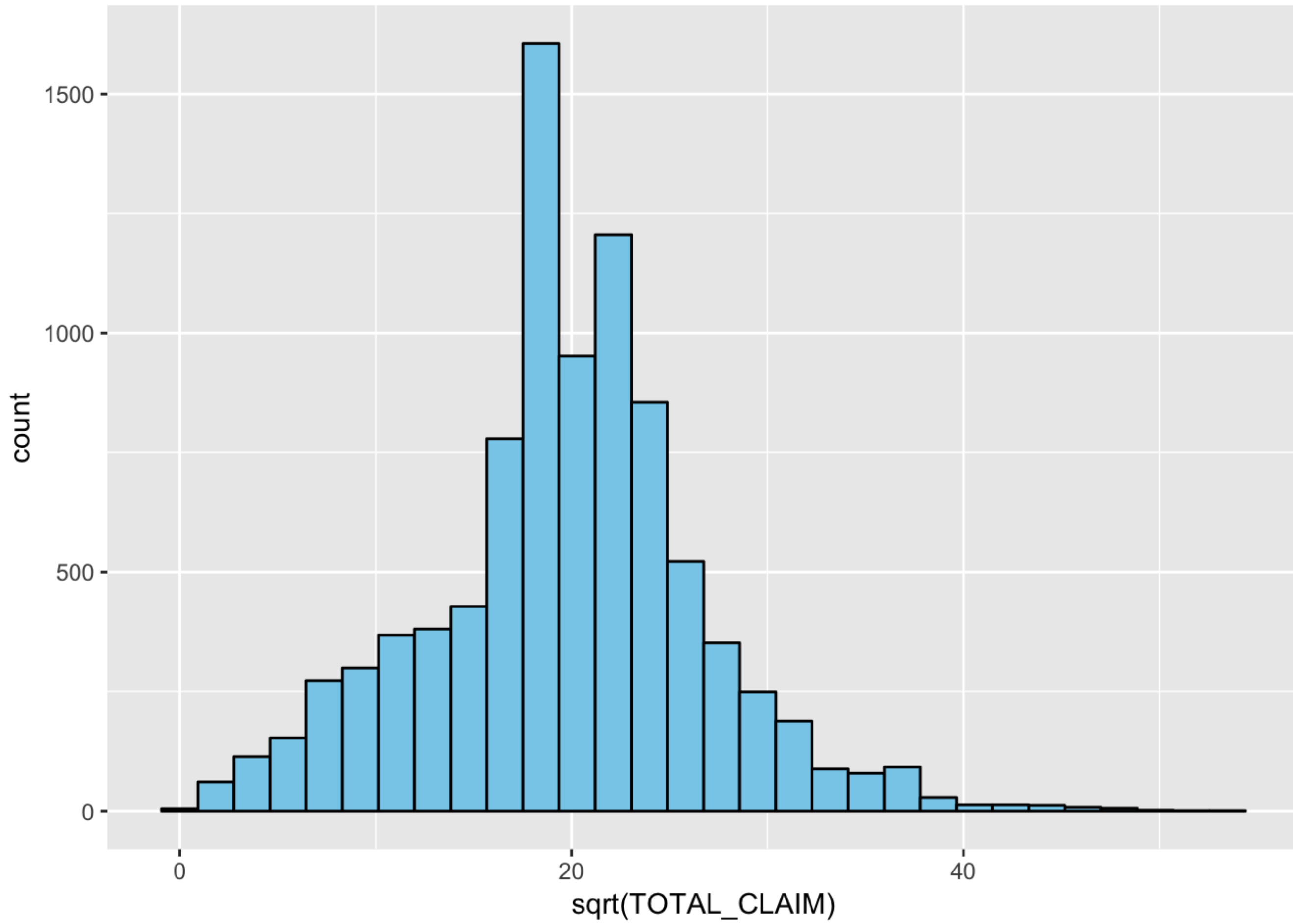
```
# It's very skewed. So I will transform the data.
ggplot(autoinsurance, aes(x=log(TOTAL_CLAIM))) + geom_histogram(color="black", fill="sky
 blue")
```

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




```
ggplot(autoinsurance, aes(x=sqrt(TOTAL_CLAIM))) + geom_histogram(color="black", fill="sk
y blue")
```

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




```
# Sqrt data looks better. Let's create a new column for this.
autoinsurance <- autoinsurance %>%
  mutate(SQRT_TOTAL_CLAIM = sqrt(TOTAL_CLAIM))

skewness(autoinsurance$SQRT_TOTAL_CLAIM)
```

```
## [1] 0.1371862
```

```
summary(autoinsurance$SQRT_TOTAL_CLAIM)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.3146 16.5003 19.5945 19.6496 23.3990 53.7888
```

Taking square root has considerably reduced the skewness. Sqrt(TOTAL_CLAIM) will be the new predicted value.

```r
# Factorize the categorical variables
autoinsurance$STATE <- factor(autoinsurance$STATE)
autoinsurance$COVERAGE <- factor(autoinsurance$COVERAGE)
autoinsurance$EDUCATION <- factor(autoinsurance$EDUCATION)
autoinsurance$EMPLOYMENT <- factor(autoinsurance$EMPLOYMENT)
autoinsurance$GENDER <- factor(autoinsurance$GENDER)
autoinsurance$LOCATION_CODE <- factor(autoinsurance$LOCATION_CODE)
autoinsurance$MARITAL_STATUS <- factor(autoinsurance$MARITAL_STATUS)
autoinsurance$POLICY_TYPE <- factor(autoinsurance$POLICY_TYPE)
autoinsurance$CLAIM_REASON <- factor(autoinsurance$CLAIM_REASON)
autoinsurance$VEHICLE_CLASS <- factor(autoinsurance$VEHICLE_CLASS)
autoinsurance$VEHICLE_SIZE <- factor(autoinsurance$VEHICLE_SIZE)
```

```r
# Create a subset of categorical variables
num_var  <- autoinsurance[, c(6,9,10,11,12,13,16,19)]
cat_var  <- autoinsurance[, -c(6,9,10,11,12,13,16,19)]
```

```r
for (i in 1:11) { # Loop over loop.vector

  # Get uniques
  print(unique(cat_var[,i]))
}
```

```
## # A tibble: 5 x 1
##   STATE
##   <fct>
## 1 Kansas
## 2 Nebraska
## 3 Oklahoma
## 4 Missouri
## 5 Iowa
## # A tibble: 3 x 1
##   COVERAGE
##   <fct>
## 1 Basic
## 2 Extended
## 3 Premium
## # A tibble: 5 x 1
##   EDUCATION
##   <fct>
## 1 Bachelor
## 2 College
## 3 Master
## 4 High School or Below
## 5 Doctor
## # A tibble: 5 x 1
##   EMPLOYMENT
##   <fct>
## 1 Employed
## 2 Unemployed
## 3 Medical Leave
## 4 Disabled
## 5 Retired
## # A tibble: 2 x 1
##   GENDER
##   <fct>
## 1 F
## 2 M
## # A tibble: 3 x 1
##   LOCATION_CODE
##   <fct>
## 1 Suburban
## 2 Rural
## 3 Urban
## # A tibble: 3 x 1
##   MARITAL_STATUS
##   <fct>
## 1 Married
## 2 Single
## 3 Divorced
## # A tibble: 3 x 1
##   POLICY_TYPE
##   <fct>
## 1 Corporate Auto
## 2 Personal Auto
## 3 Special Auto
```

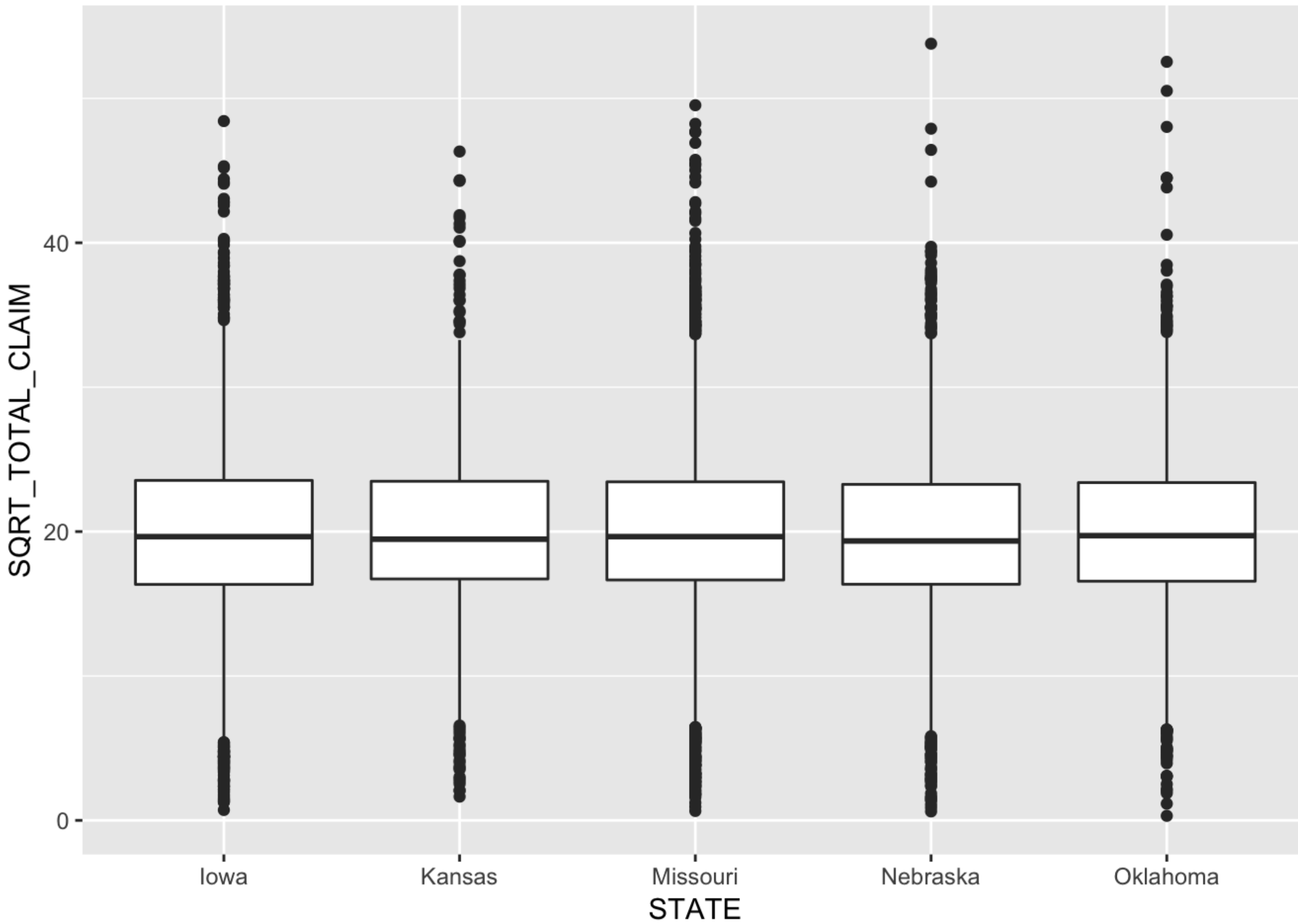
```
## # A tibble: 4 x 1
##   CLAIM_REASON
##   <fct>
## 1 Collision
## 2 Scratch/Dent
## 3 Hail
## 4 Other
## # A tibble: 6 x 1
##   VEHICLE_CLASS
##   <fct>
## 1 Two-Door Car
## 2 Four-Door Car
## 3 SUV
## 4 Luxury SUV
## 5 Sports Car
## 6 Luxury Car
## # A tibble: 3 x 1
##   VEHICLE_SIZE
##   <fct>
## 1 Medsize
## 2 Small
## 3 Large
```

```
# Random sample indexes
set.seed(123)
train_index <- sample(1:nrow(autoinsurance), 0.75 * nrow(autoinsurance))
test_index <- setdiff(1:nrow(autoinsurance), train_index)

# Split train test data
train <- autoinsurance[train_index,]
test <- autoinsurance[test_index,]

num_var_train  <- train[, c(6,9,10,11,12,13,16,19)] #numerical variables
cat_var_train  <- train[, -c(6,9,10,11,12,13,16,19)] #categorical variables
```

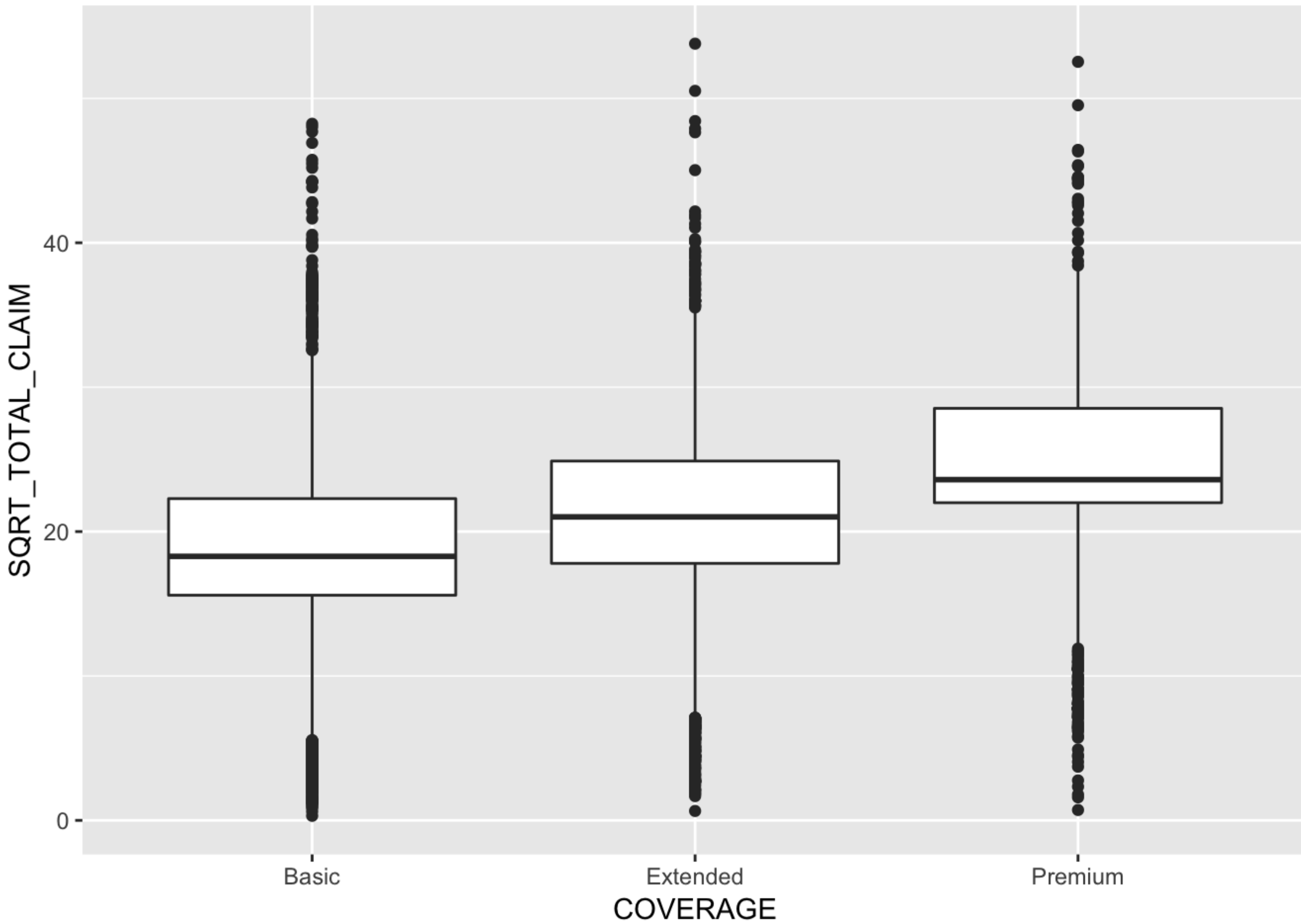
```
# Categorical Variables: STATE, COVERAGE, EDUCATION, EMPLOYMENT, GENDER, CLAIM_REASON, L
OCATION_CODE, MARITAL_STATUS, POLICY_TYPE, CLAIM_REASON, VEHICLE_CLASS, VEHICLE_SIZE
```

3. Fitting Multiple Linear Regression Model

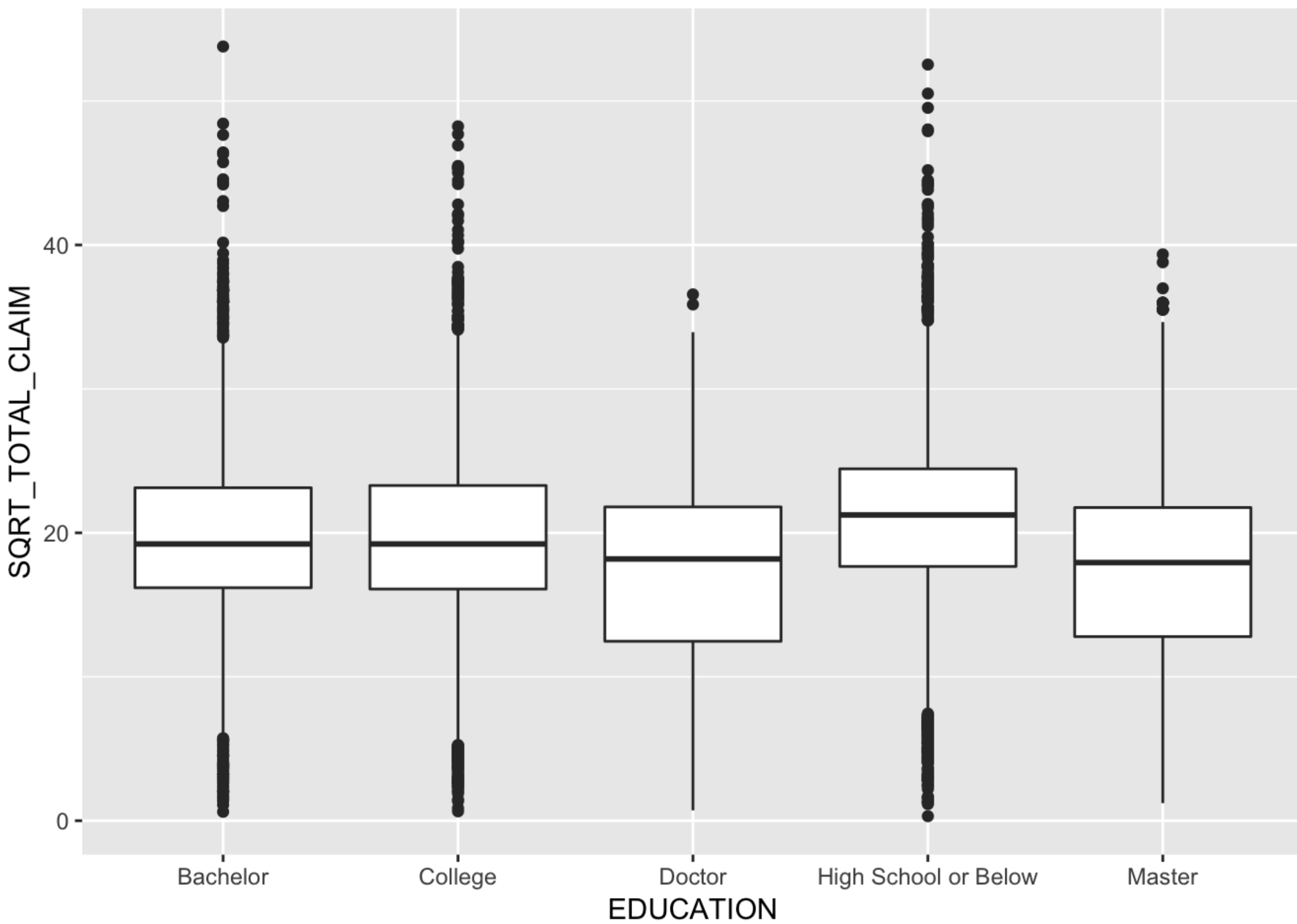
```
ggplot(autoinsurance, aes(x = STATE, y = SQRT_TOTAL_CLAIM)) + geom_boxplot()
```



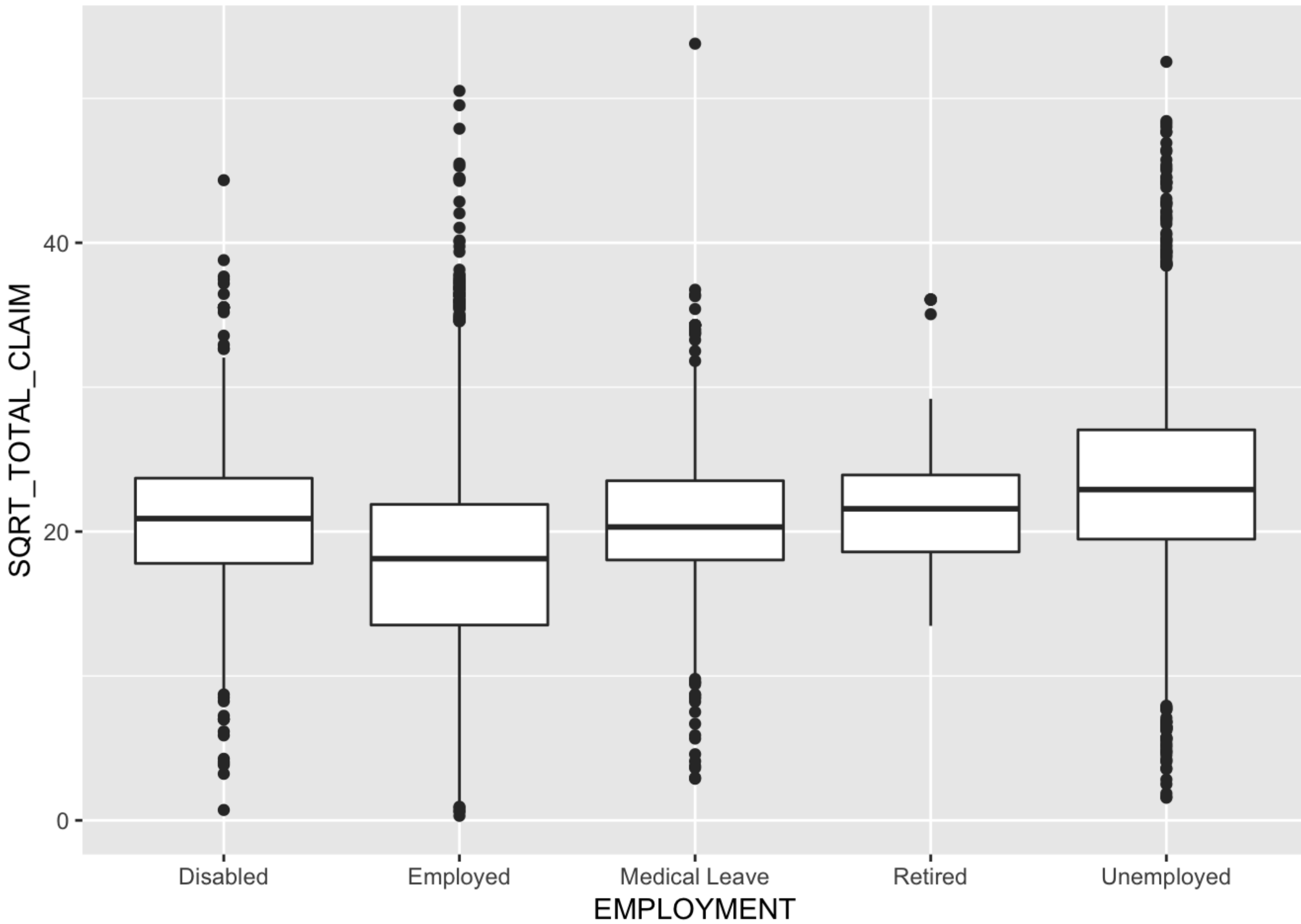

```
ggplot(autoinsurance, aes(x = COVERAGE, y = SQRT_TOTAL_CLAIM)) + geom_boxplot()
```



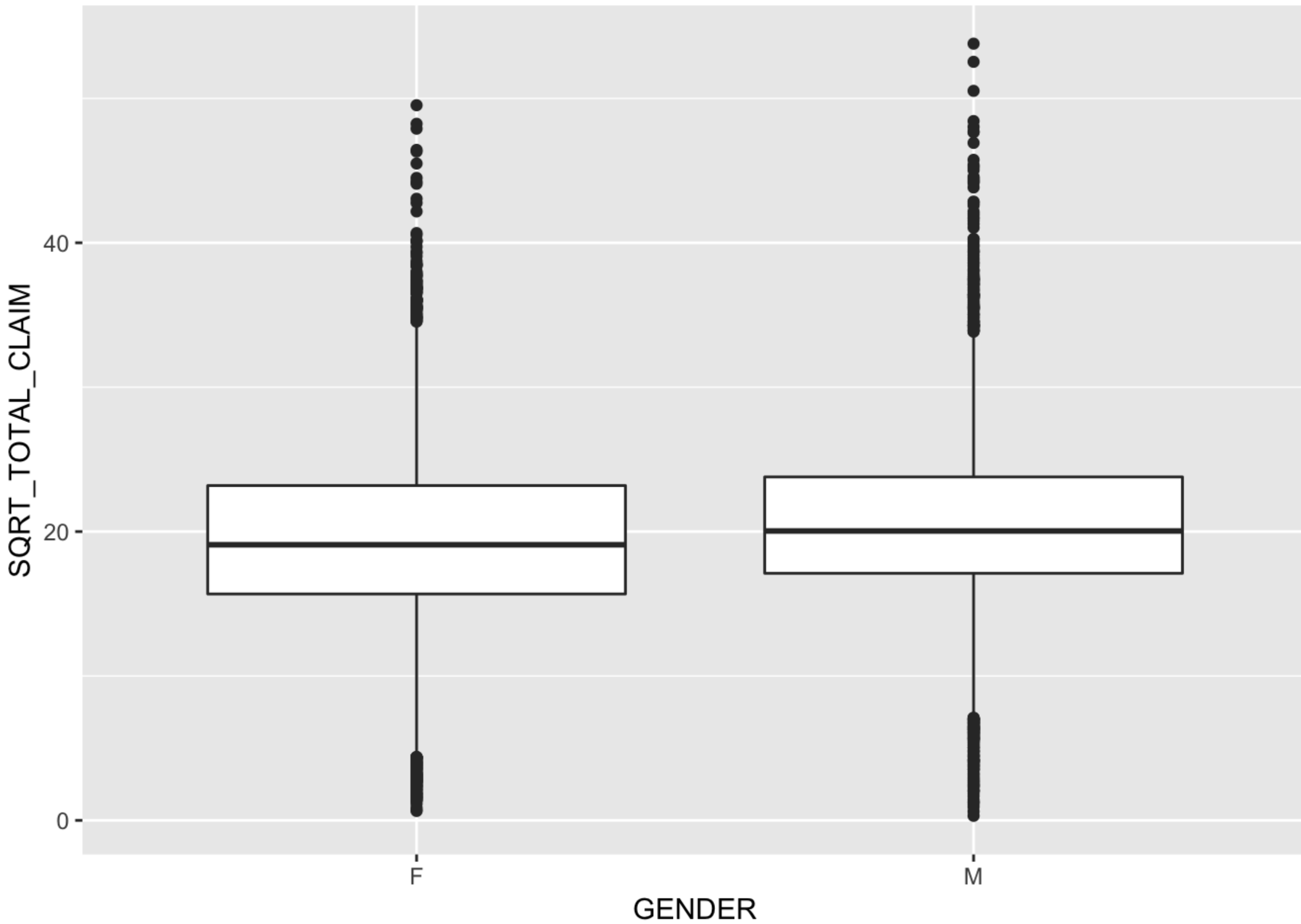

```
ggplot(autoinsurance, aes(x = EDUCATION, y = SQRT_TOTAL_CLAIM)) + geom_boxplot()
```



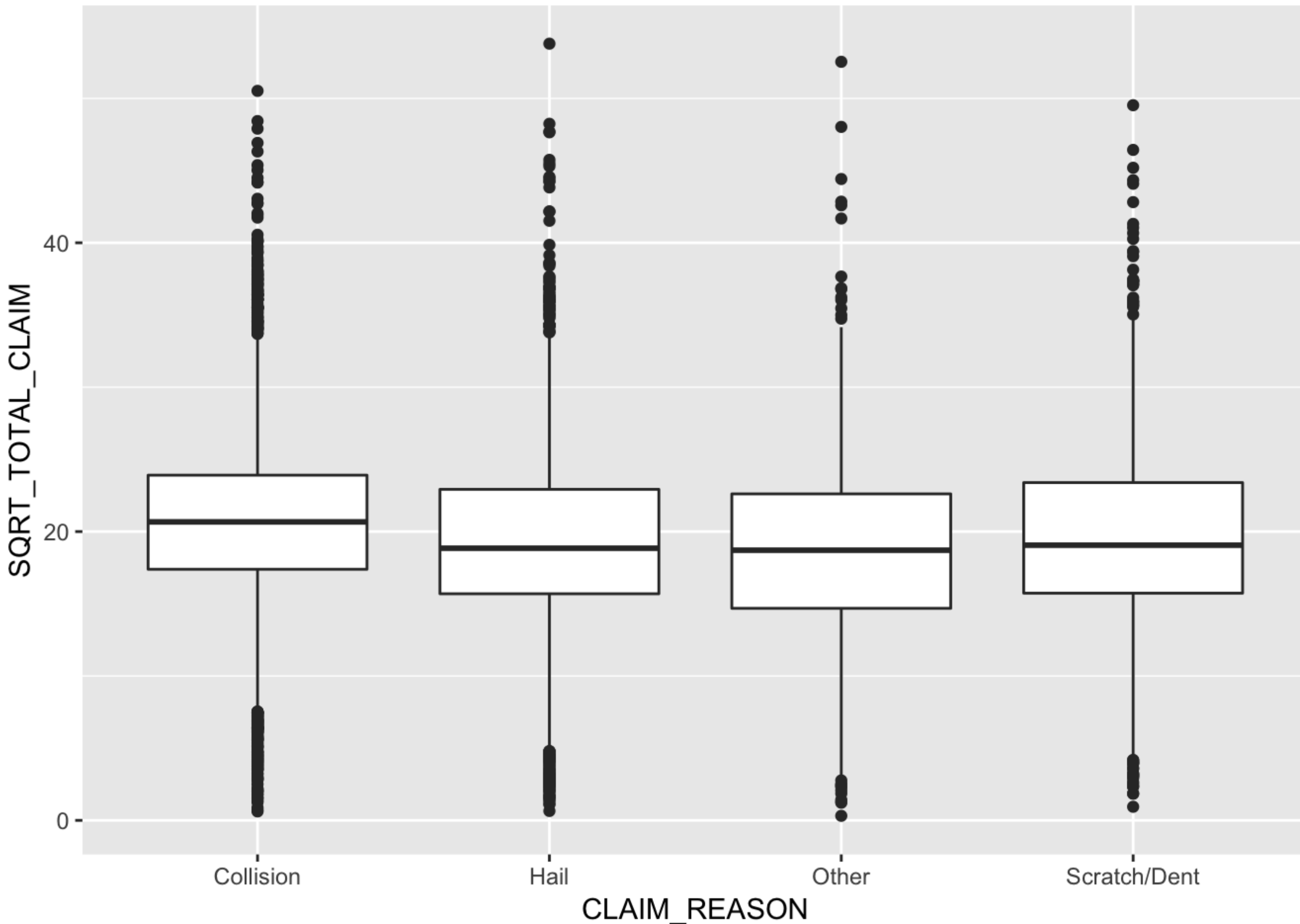

```
ggplot(autoinsurance, aes(x = EMPLOYMENT, y = SQRT_TOTAL_CLAIM)) + geom_boxplot()
```



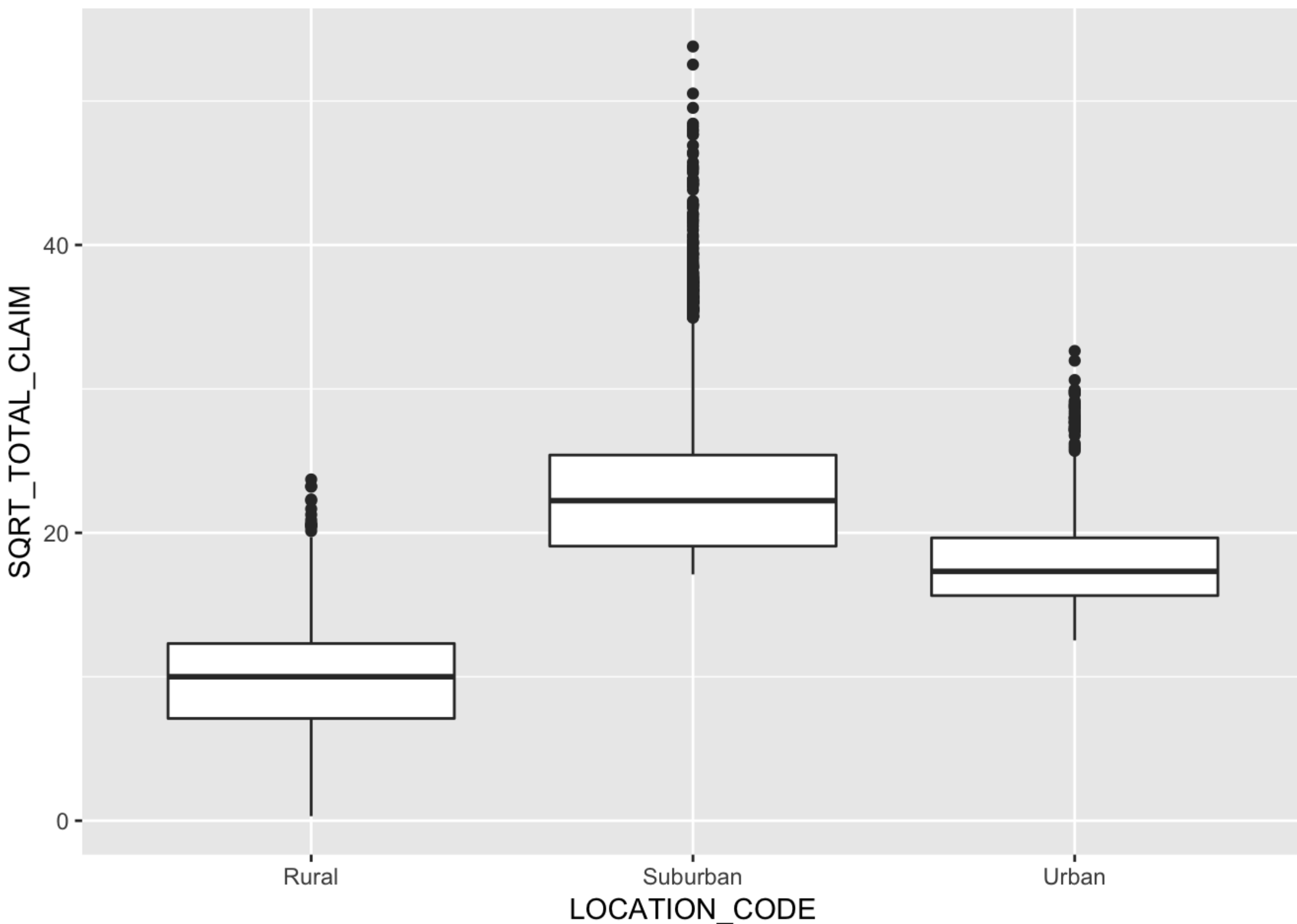

```
ggplot(autoinsurance, aes(x = GENDER, y = SQRT_TOTAL_CLAIM)) + geom_boxplot()
```



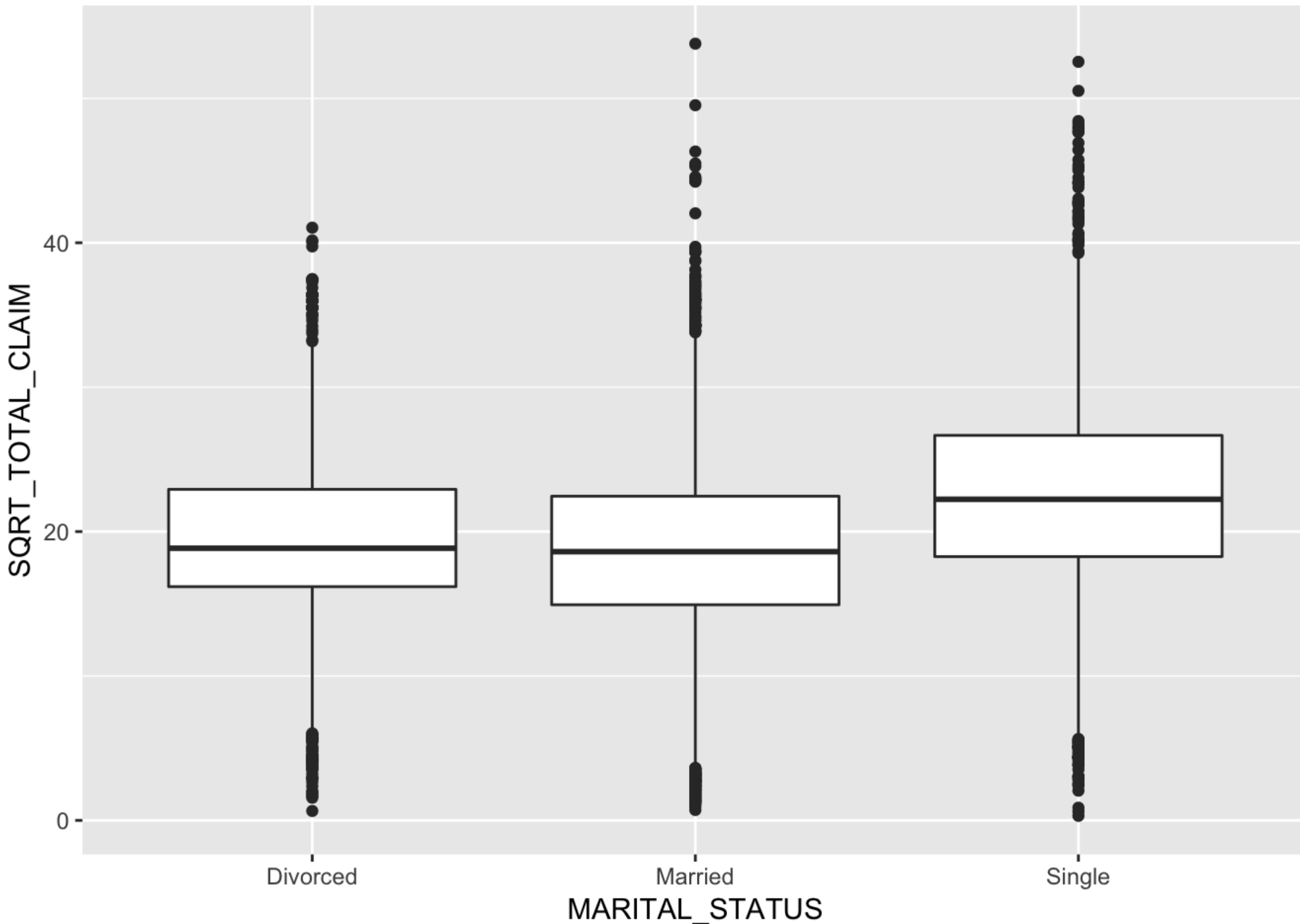

```
ggplot(autoinsurance, aes(x = CLAIM_REASON, y = SQRT_TOTAL_CLAIM)) + geom_boxplot()
```



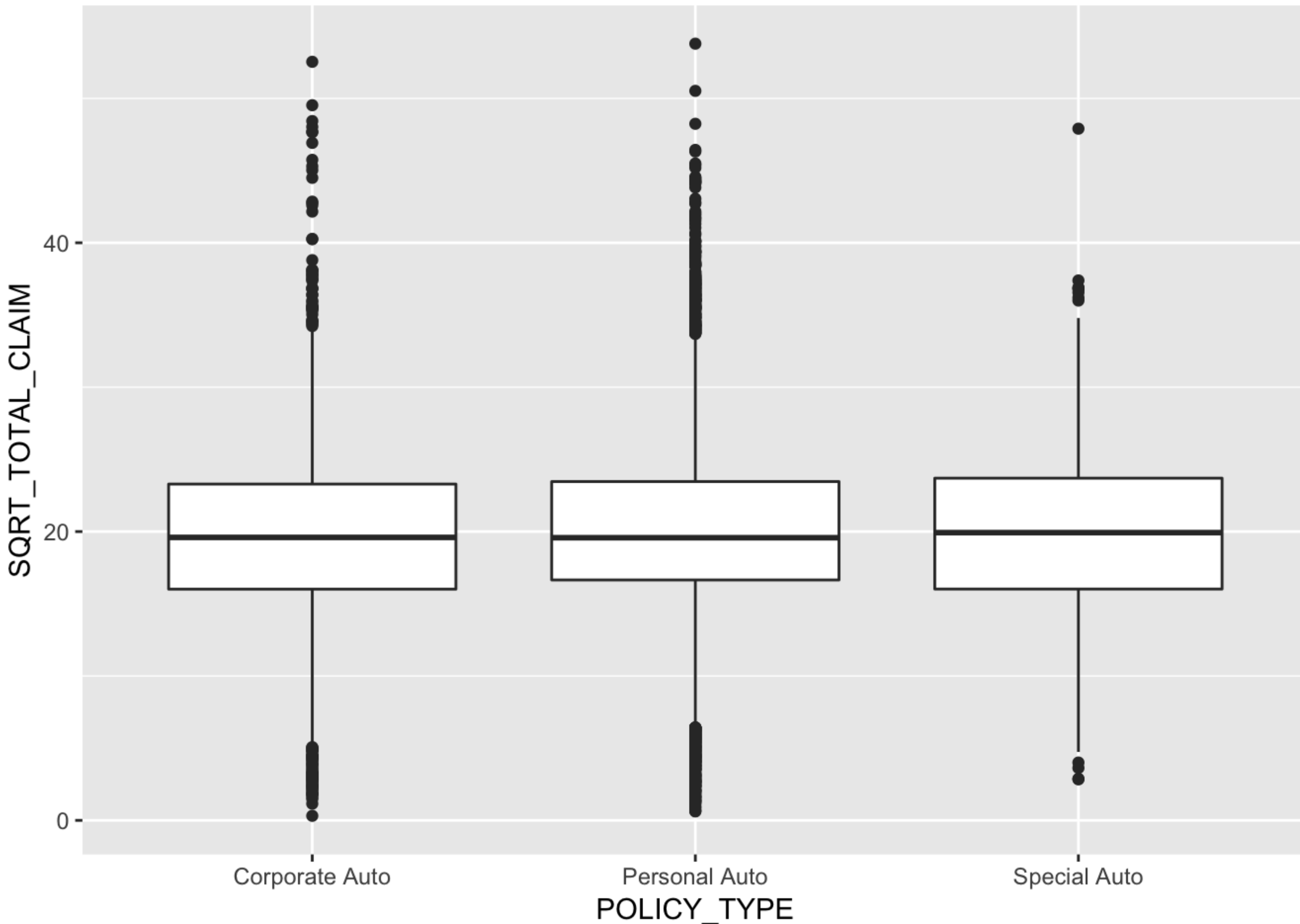

```
ggplot(autoinsurance, aes(x = LOCATION_CODE, y = SQRT_TOTAL_CLAIM)) + geom_boxplot()
```



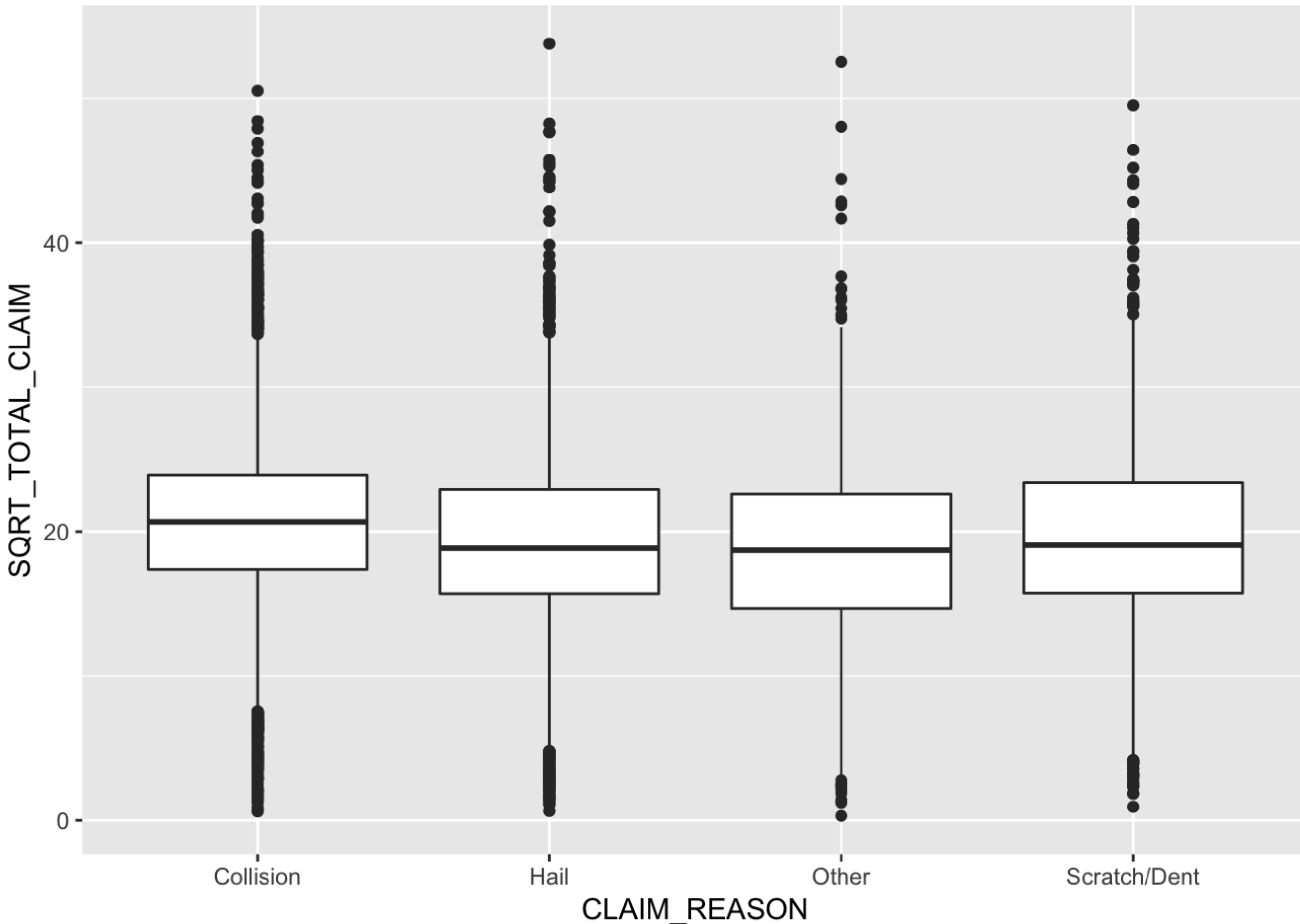

```
ggplot(autoinsurance, aes(x = MARITAL_STATUS, y = SQRT_TOTAL_CLAIM)) + geom_boxplot()
```



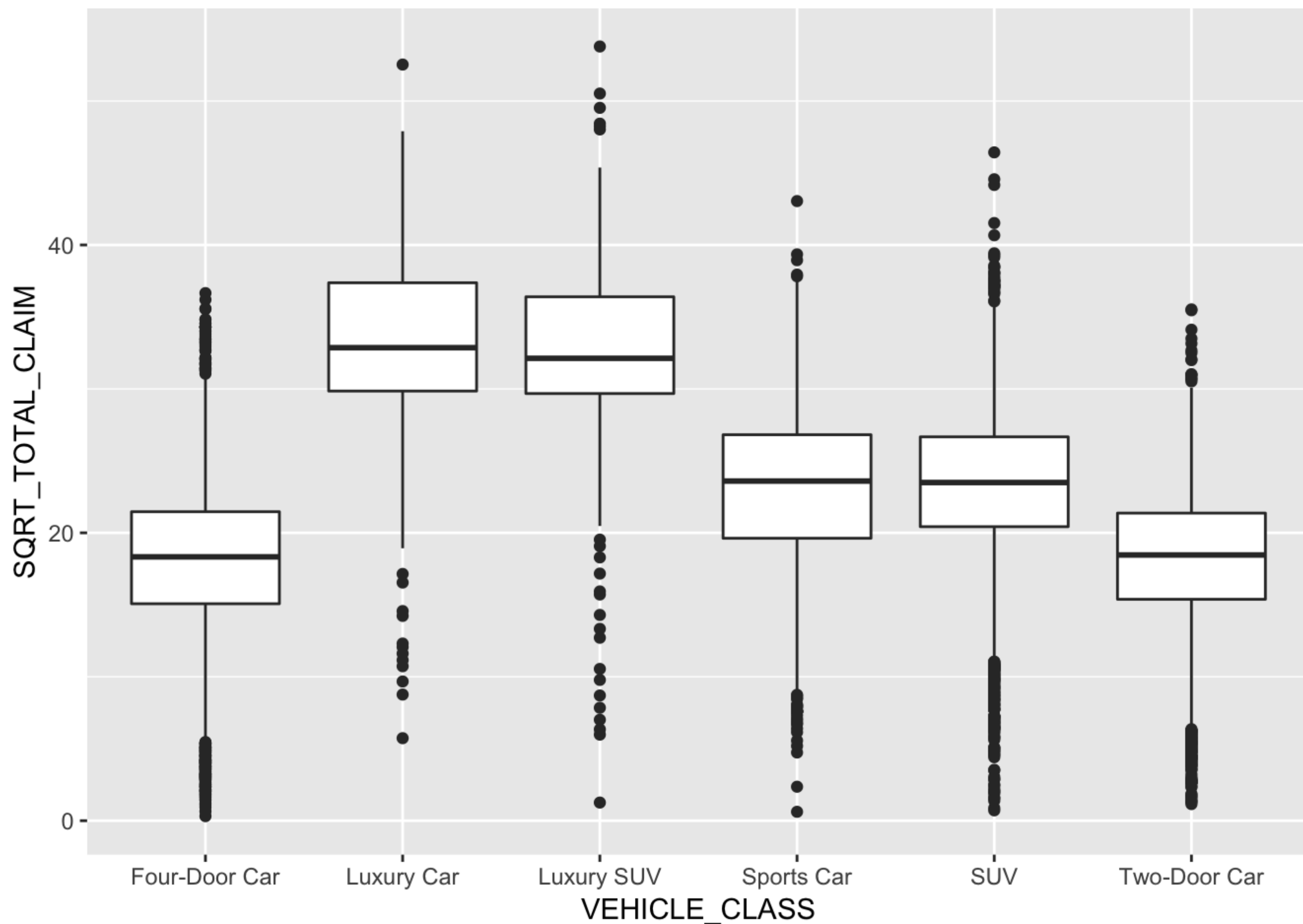

```
ggplot(autoinsurance, aes(x = POLICY_TYPE, y = SQRT_TOTAL_CLAIM)) + geom_boxplot()
```



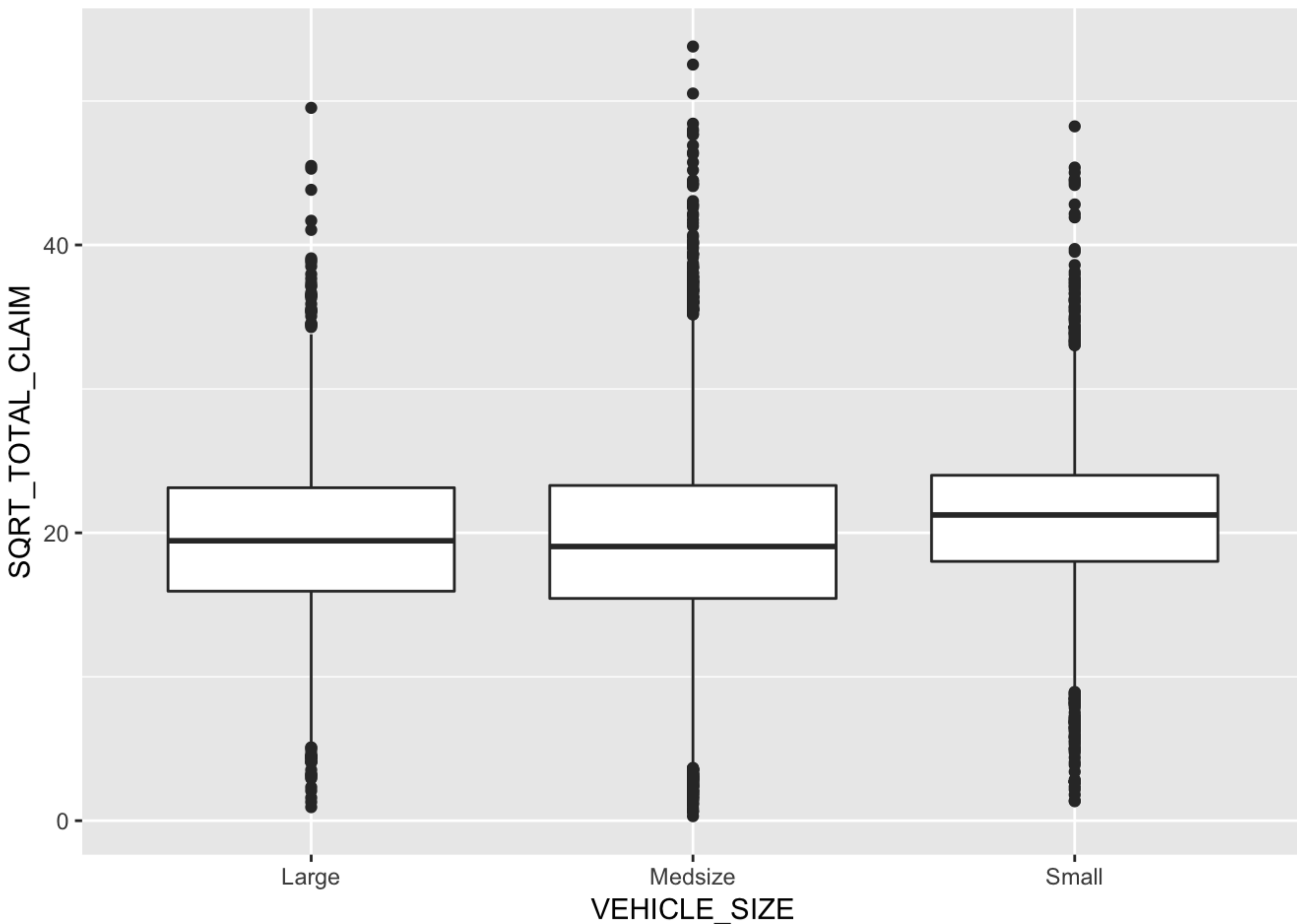

```
ggplot(autoinsurance, aes(x = CLAIM_REASON, y = SQRT_TOTAL_CLAIM)) + geom_boxplot()
```




```
ggplot(autoinsurance, aes(x = VEHICLE_CLASS, y = SQRT_TOTAL_CLAIM)) + geom_boxplot()
```




```
ggplot(autoinsurance, aes(x = VEHICLE_SIZE, y = SQRT_TOTAL_CLAIM)) + geom_boxplot()
```



Looing at the boxplots, the mean between the groups mostly differ in LOCATION_CODE, EMPLOYMENT, VEHICLE_CLASS.

```
# ANOVA analysis
anova(aov(SQRT_TOTAL_CLAIM ~ MARITAL_STATUS, data=train))
```

```
## Analysis of Variance Table
##
## Response: SQRT_TOTAL_CLAIM
##                  Df Sum Sq Mean Sq F value    Pr(>F)
## MARITAL_STATUS    2  19069  9534.3  215.47 < 2.2e-16 ***
## Residuals      6847 302974    44.2
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(aov(SQRT_TOTAL_CLAIM ~ EDUCATION, data=train))
```

```
## Analysis of Variance Table
##
## Response: SQRT_TOTAL_CLAIM
##             Df Sum Sq Mean Sq F value    Pr(>F)
## EDUCATION    4   6693 1673.30  36.321 < 2.2e-16 ***
## Residuals 6845 315350   46.07
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(aov(SQRT_TOTAL_CLAIM ~ STATE, data=train))
```

```
## Analysis of Variance Table
##
## Response: SQRT_TOTAL_CLAIM
##             Df Sum Sq Mean Sq F value Pr(>F)
## STATE        4     87  21.715  0.4617 0.7639
## Residuals 6845 321956  47.035
```

```
anova(aov(SQRT_TOTAL_CLAIM ~ COVERAGE, data=train))
```

```
## Analysis of Variance Table
##
## Response: SQRT_TOTAL_CLAIM
##             Df Sum Sq Mean Sq F value    Pr(>F)
## COVERAGE     2  21341 10670.5  242.97 < 2.2e-16 ***
## Residuals 6847 300702    43.9
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(aov(SQRT_TOTAL_CLAIM ~ EMPLOYMENT, data=train))
```

```
## Analysis of Variance Table
##
## Response: SQRT_TOTAL_CLAIM
##              Df Sum Sq Mean Sq F value    Pr(>F)
## EMPLOYMENT    4  44764 11191.0  276.26 < 2.2e-16 ***
## Residuals  6845 277279    40.5
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(aov(SQRT_TOTAL_CLAIM ~ CLAIM_REASON, data=train))
```

```
## Analysis of Variance Table
##
## Response: SQRT_TOTAL_CLAIM
##                  Df Sum Sq Mean Sq F value    Pr(>F)
## CLAIM_REASON    3   4149 1382.91  29.782 < 2.2e-16 ***
## Residuals    6846 317894   46.44
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(aov(SQRT_TOTAL_CLAIM ~ LOCATION_CODE, data=train))
```

```
## Analysis of Variance Table
##
## Response: SQRT_TOTAL_CLAIM
##                   Df Sum Sq Mean Sq F value    Pr(>F)
## LOCATION_CODE    2 183682   91841  4544.9 < 2.2e-16 ***
## Residuals     6847 138361      20
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(aov(SQRT_TOTAL_CLAIM ~ VEHICLE_CLASS, data=train))
```

```
## Analysis of Variance Table
##
## Response: SQRT_TOTAL_CLAIM
##                   Df Sum Sq Mean Sq F value    Pr(>F)
## VEHICLE_CLASS    5  72086 14417.3  394.76 < 2.2e-16 ***
## Residuals     6844 249957    36.5
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(aov(SQRT_TOTAL_CLAIM ~ POLICY_TYPE, data=train))
```

```
## Analysis of Variance Table
##
## Response: SQRT_TOTAL_CLAIM
##               Df Sum Sq Mean Sq F value Pr(>F)
## POLICY_TYPE    2     68  34.118  0.7255 0.4841
## Residuals   6847 321975  47.024
```

```
anova(aov(SQRT_TOTAL_CLAIM ~ VEHICLE_SIZE, data=train))
```

```
## Analysis of Variance Table
##
## Response: SQRT_TOTAL_CLAIM
##                Df Sum Sq Mean Sq F value    Pr(>F)
## VEHICLE_SIZE    2   4046 2022.93  43.557 < 2.2e-16 ***
## Residuals    6847 317997   46.44
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The ANOVA result shows that there is not a significant difference in the group means between STATE and POLICY_TYPES.

One of the assumption of multiple regression is that the predictor variables are numeric or are categorical with maximal two categories. However in our dataset we have the variable region containing four categories. Normally we should use dummy variables. However this is something the lm function in R does automatically.

```
# Fitting the first model with all categorical variables
fit1 <- lm(SQRT_TOTAL_CLAIM ~ STATE + COVERAGE + EDUCATION + EMPLOYMENT + GENDER + CLAIM
_REASON + LOCATION_CODE + MARITAL_STATUS + POLICY_TYPE + VEHICLE_CLASS + VEHICLE_SIZE, d
ata=train)

summary(fit1)
```

```
##
## Call:
## lm(formula = SQRT_TOTAL_CLAIM ~ STATE + COVERAGE + EDUCATION +
##     EMPLOYMENT + GENDER + CLAIM_REASON + LOCATION_CODE + MARITAL_STATUS +
##     POLICY_TYPE + VEHICLE_CLASS + VEHICLE_SIZE, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.7546  -1.7961  -0.3764   1.6571  19.3149
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   7.09055    0.25291  28.036  < 2e-16 ***
## STATEKansas                   0.14850    0.13313   1.115 0.264676
## STATEMissouri                 0.02313    0.08781   0.263 0.792285
## STATENebraska                 0.01648    0.10409   0.158 0.874206
## STATEOklahoma                 0.06481    0.13035   0.497 0.619094
## COVERAGEExtended              2.16117    0.07740  27.923  < 2e-16 ***
## COVERAGEPremium               5.03780    0.12591  40.010  < 2e-16 ***
## EDUCATIONCollege             -0.19144    0.08986  -2.130 0.033167 *
## EDUCATIONDoctor              -0.40427    0.19174  -2.108 0.035034 *
## EDUCATIONHigh School or Below  0.11881    0.09148   1.299 0.194045
## EDUCATIONMaster              -0.17928    0.13992  -1.281 0.200125
## EMPLOYMENTEmployed           -0.22930    0.17570  -1.305 0.191901
## EMPLOYMENTMedical Leave       0.40817    0.23212   1.758 0.078717 .
## EMPLOYMENTRetired            -0.33809    0.25612  -1.320 0.186863
## EMPLOYMENTUnemployed          1.35710    0.18466   7.349 2.23e-13 ***
## GENDERM                       0.26058    0.07006   3.719 0.000201 ***
## CLAIM_REASONHail              0.18100    0.08434   2.146 0.031902 *
## CLAIM_REASONOther             0.06615    0.12040   0.549 0.582758
## CLAIM_REASONScratch/Dent      0.19353    0.10484   1.846 0.064935 .
## LOCATION_CODESuburban        11.93919    0.09959 119.884  < 2e-16 ***
## LOCATION_CODEUrban            8.04409    0.11541  69.698  < 2e-16 ***
## MARITAL_STATUSMarried        -0.16099    0.10228  -1.574 0.115518
## MARITAL_STATUSSingle          1.04972    0.11732   8.948  < 2e-16 ***
## POLICY_TYPEPersonal Auto      0.10478    0.08493   1.234 0.217373
## POLICY_TYPESpecial Auto       0.19152    0.18255   1.049 0.294161
## VEHICLE_CLASSLuxury Car      12.67504    0.27186  46.624  < 2e-16 ***
## VEHICLE_CLASSLuxury SUV      12.39251    0.25536  48.530  < 2e-16 ***
## VEHICLE_CLASSSports Car       4.75509    0.16250  29.263  < 2e-16 ***
## VEHICLE_CLASSSUV              4.72582    0.09292  50.858  < 2e-16 ***
## VEHICLE_CLASSTwo-Door Car     0.07520    0.09044   0.831 0.405759
## VEHICLE_SIZEMedsize           0.08047    0.11617   0.693 0.488524
## VEHICLE_SIZESmall             0.37994    0.13491   2.816 0.004874 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.871 on 6818 degrees of freedom
## Multiple R-squared:  0.8255, Adjusted R-squared:  0.8248
## F-statistic:  1041 on 31 and 6818 DF,  p-value: < 2.2e-16
```

```
# Remove STATE, POLICY_TYPE, the ones with no significance
fit2 <- lm(SQRT_TOTAL_CLAIM ~ COVERAGE + EDUCATION + EMPLOYMENT + GENDER + CLAIM_REASON
 + LOCATION_CODE + MARITAL_STATUS + VEHICLE_CLASS + VEHICLE_SIZE,data=train)

summary(fit2)
```

```
##
## Call:
## lm(formula = SQRT_TOTAL_CLAIM ~ COVERAGE + EDUCATION + EMPLOYMENT +
##     GENDER + CLAIM_REASON + LOCATION_CODE + MARITAL_STATUS +
##     VEHICLE_CLASS + VEHICLE_SIZE, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.7634  -1.7845  -0.3783   1.6541  19.3220
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   7.20187    0.23862  30.182  < 2e-16 ***
## COVERAGEExtended              2.16159    0.07737  27.939  < 2e-16 ***
## COVERAGEPremium               5.03657    0.12586  40.018  < 2e-16 ***
## EDUCATIONCollege             -0.19079    0.08984  -2.124 0.033729 *
## EDUCATIONDoctor              -0.40056    0.19161  -2.090 0.036612 *
## EDUCATIONHigh School or Below  0.12087    0.09144   1.322 0.186273
## EDUCATIONMaster              -0.17178    0.13978  -1.229 0.219145
## EMPLOYMENTEmployed           -0.23007    0.17565  -1.310 0.190302
## EMPLOYMENTMedical Leave       0.40974    0.23205   1.766 0.077482 .
## EMPLOYMENTRetired            -0.33599    0.25605  -1.312 0.189486
## EMPLOYMENTUnemployed          1.35319    0.18460   7.330 2.56e-13 ***
## GENDERM                       0.25920    0.07004   3.701 0.000216 ***
## CLAIM_REASONHail              0.18308    0.08429   2.172 0.029892 *
## CLAIM_REASONOther             0.06452    0.12036   0.536 0.591962
## CLAIM_REASONScratch/Dent      0.19289    0.10481   1.840 0.065747 .
## LOCATION_CODESuburban        11.94274    0.09954 119.983  < 2e-16 ***
## LOCATION_CODEUrban            8.04813    0.11533  69.782  < 2e-16 ***
## MARITAL_STATUSMarried        -0.15811    0.10222  -1.547 0.121955
## MARITAL_STATUSSingle          1.05577    0.11723   9.006  < 2e-16 ***
## VEHICLE_CLASSLuxury Car      12.67059    0.27176  46.624  < 2e-16 ***
## VEHICLE_CLASSLuxury SUV      12.38579    0.25523  48.527  < 2e-16 ***
## VEHICLE_CLASSSports Car       4.75253    0.16239  29.267  < 2e-16 ***
## VEHICLE_CLASSSUV              4.72225    0.09286  50.856  < 2e-16 ***
## VEHICLE_CLASSTwo-Door Car     0.07371    0.09038   0.816 0.414781
## VEHICLE_SIZEMedsize           0.08091    0.11614   0.697 0.486052
## VEHICLE_SIZESmall             0.37970    0.13486   2.815 0.004884 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.87 on 6824 degrees of freedom
## Multiple R-squared:  0.8255, Adjusted R-squared:  0.8248
## F-statistic:  1291 on 25 and 6824 DF,  p-value: < 2.2e-16
```

```
anova(fit1,fit2)
```

```
## Analysis of Variance Table
##
## Model 1: SQRT_TOTAL_CLAIM ~ STATE + COVERAGE + EDUCATION + EMPLOYMENT +
##     GENDER + CLAIM_REASON + LOCATION_CODE + MARITAL_STATUS +
##     POLICY_TYPE + VEHICLE_CLASS + VEHICLE_SIZE
## Model 2: SQRT_TOTAL_CLAIM ~ COVERAGE + EDUCATION + EMPLOYMENT + GENDER +
##     CLAIM_REASON + LOCATION_CODE + MARITAL_STATUS + VEHICLE_CLASS +
##     VEHICLE_SIZE
##   Res.Df   RSS Df Sum of Sq      F Pr(>F)
## 1   6818 56182
## 2   6824 56210 -6   -27.413 0.5544 0.7668
```

With a p value > 0.05, we can see that there is not much difference between model 2 and model 1. Thus we keep model 2 for less variables. Without STATE and POLICY_TYPE, both model explains 82.55% the variability in SQRT_TOTAL_CLAIM.

Now, let's explore the numerical variables

```
# Numerical variables:
cormatrix <- round(cor(num_var_train), 3)
cormatrix
```

```
##                                  INCOME MONTHLY_PREMIUM
## INCOME                            1.000          -0.020
## MONTHLY_PREMIUM                  -0.020           1.000
## MONTHS_SINCE_LAST_CLAIM          -0.031           0.001
## MONTHS_SINCE_POLICY_INCEPTION     0.002           0.028
## NUMBER_COMPLAINTS                 0.014          -0.008
## NUMBER_POLICIES                  -0.013          -0.008
## TOTAL_CLAIM                      -0.357           0.632
## SQRT_TOTAL_CLAIM                 -0.378           0.539
##                               MONTHS_SINCE_LAST_CLAIM
## INCOME                                         -0.031
## MONTHLY_PREMIUM                                 0.001
## MONTHS_SINCE_LAST_CLAIM                         1.000
## MONTHS_SINCE_POLICY_INCEPTION                  -0.049
## NUMBER_COMPLAINTS                               0.007
## NUMBER_POLICIES                                 0.012
## TOTAL_CLAIM                                     0.003
## SQRT_TOTAL_CLAIM                               -0.008
##                               MONTHS_SINCE_POLICY_INCEPTION
## INCOME                                                0.002
## MONTHLY_PREMIUM                                       0.028
## MONTHS_SINCE_LAST_CLAIM                              -0.049
## MONTHS_SINCE_POLICY_INCEPTION                         1.000
## NUMBER_COMPLAINTS                                     0.002
## NUMBER_POLICIES                                      -0.010
## TOTAL_CLAIM                                           0.010
## SQRT_TOTAL_CLAIM                                      0.007
##                               NUMBER_COMPLAINTS NUMBER_POLICIES
## INCOME                                    0.014          -0.013
## MONTHLY_PREMIUM                          -0.008          -0.008
## MONTHS_SINCE_LAST_CLAIM                   0.007           0.012
## MONTHS_SINCE_POLICY_INCEPTION             0.002          -0.010
## NUMBER_COMPLAINTS                         1.000           0.001
## NUMBER_POLICIES                           0.001           1.000
## TOTAL_CLAIM                              -0.012           0.009
## SQRT_TOTAL_CLAIM                         -0.009           0.009
##                               TOTAL_CLAIM SQRT_TOTAL_CLAIM
## INCOME                             -0.357           -0.378
## MONTHLY_PREMIUM                     0.632            0.539
## MONTHS_SINCE_LAST_CLAIM             0.003           -0.008
## MONTHS_SINCE_POLICY_INCEPTION       0.010            0.007
## NUMBER_COMPLAINTS                  -0.012           -0.009
## NUMBER_POLICIES                     0.009            0.009
## TOTAL_CLAIM                         1.000            0.961
## SQRT_TOTAL_CLAIM                    0.961            1.000
```

There is only noticeable correlation with INCOME and MONTHLY PREMIUM.

```
# Trying the models with numerical variables
summary(lm(SQRT_TOTAL_CLAIM ~ INCOME, data = train))
```

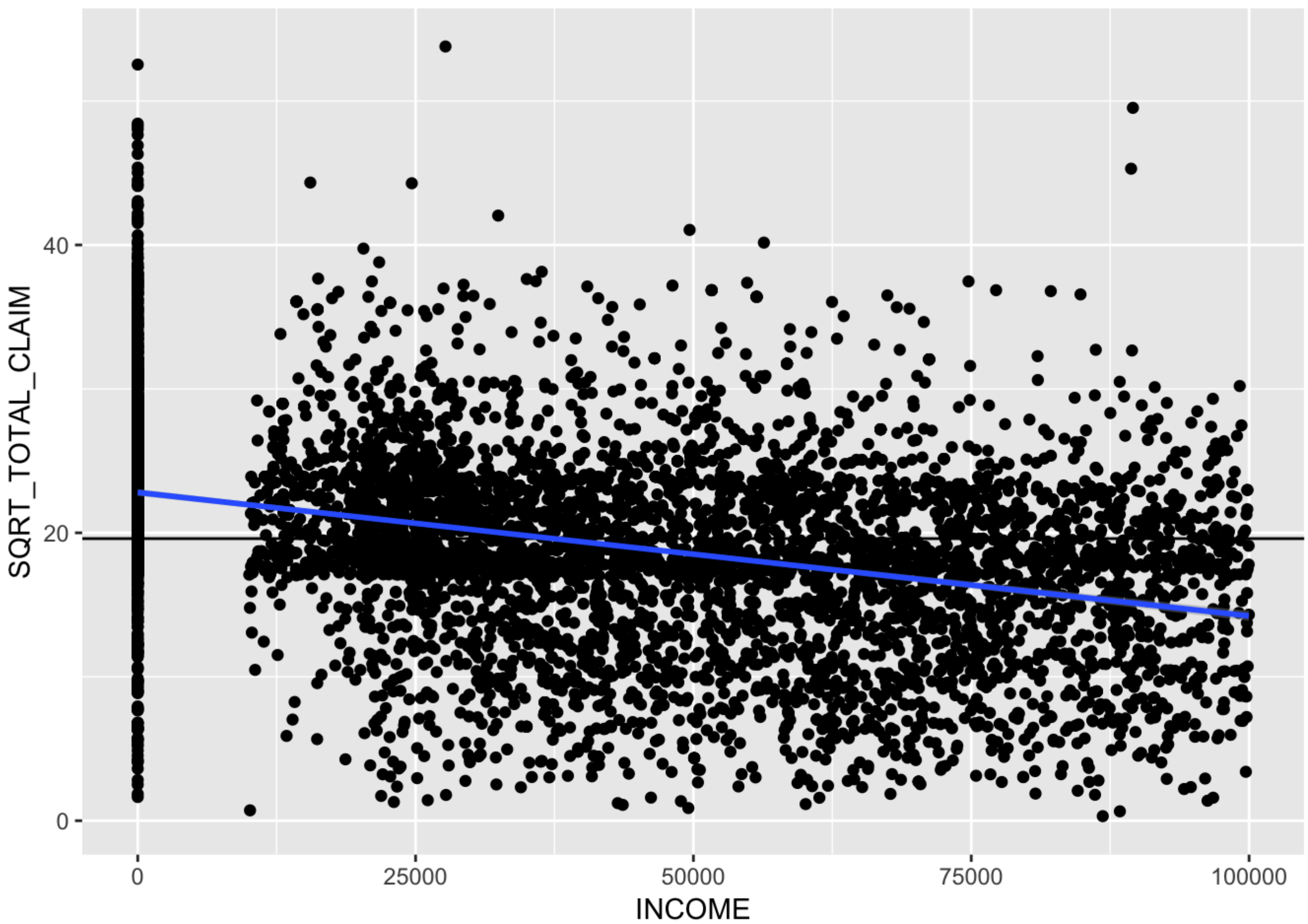
```
##
## Call:
## lm(formula = SQRT_TOTAL_CLAIM ~ INCOME, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -21.220  -3.628  -0.038   3.720  34.401
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.281e+01  1.220e-01  187.00   <2e-16 ***
## INCOME      -8.578e-05  2.535e-06  -33.83   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.348 on 6848 degrees of freedom
## Multiple R-squared:  0.1432, Adjusted R-squared:  0.1431
## F-statistic:  1145 on 1 and 6848 DF,  p-value: < 2.2e-16
```

```
summary(lm(SQRT_TOTAL_CLAIM ~ MONTHLY_PREMIUM, data = train))
```

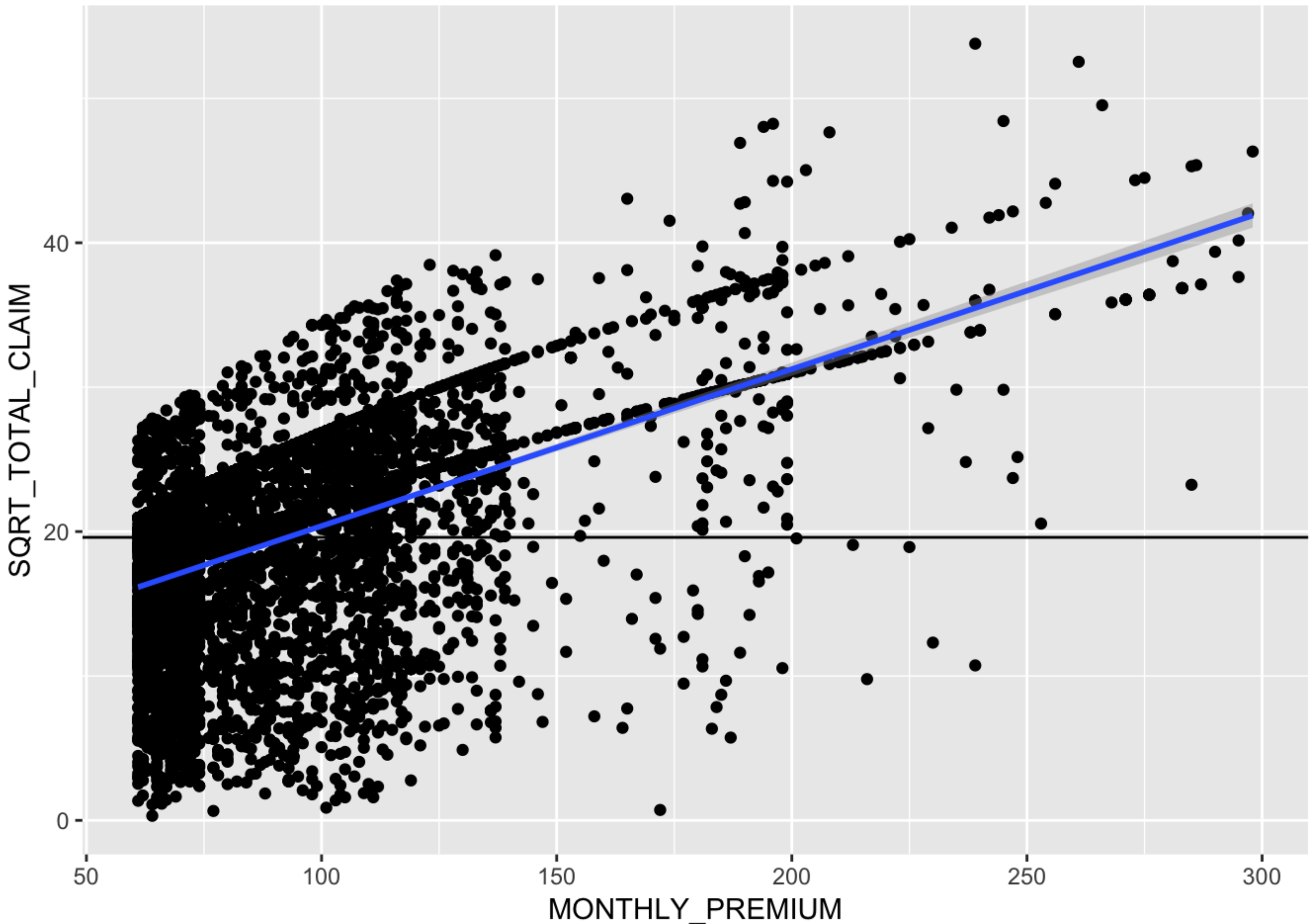
```
##
## Call:
## lm(formula = SQRT_TOTAL_CLAIM ~ MONTHLY_PREMIUM, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -27.487  -2.286   1.222   2.857  18.306
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     9.529281   0.202448   47.07   <2e-16 ***
## MONTHLY_PREMIUM 0.108590   0.002049   52.98   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.775 on 6848 degrees of freedom
## Multiple R-squared:  0.2908, Adjusted R-squared:  0.2906
## F-statistic:  2807 on 1 and 6848 DF,  p-value: < 2.2e-16
```

Each numerical variable alone explains a considerable percentage of the variability in SQRT_TOTAL_CLAIM.

```
ggplot(train, aes(x = INCOME, y = SQRT_TOTAL_CLAIM)) +
 geom_point() +
 geom_hline(yintercept = mean(train$SQRT_TOTAL_CLAIM)) +
 geom_smooth(method='lm')
```




```
ggplot(train, aes(x = MONTHLY_PREMIUM, y = SQRT_TOTAL_CLAIM)) +
 geom_point() +
 geom_hline(yintercept = mean(train$SQRT_TOTAL_CLAIM)) +
 geom_smooth(method='lm')
```



The only problem is that we have a lot of people with 0 INCOME. These are also people who are Unemployed. This might be a colinearity problem for these two variables.

```
# Incorporate numerical variables into the model
fit3 = lm(SQRT_TOTAL_CLAIM ~ COVERAGE + EDUCATION + EMPLOYMENT + GENDER + CLAIM_REASON +
 LOCATION_CODE + MARITAL_STATUS + VEHICLE_CLASS + VEHICLE_SIZE + INCOME + MONTHLY_PREMIU
M, data=train)
summary(fit3)
```

```
##
## Call:
## lm(formula = SQRT_TOTAL_CLAIM ~ COVERAGE + EDUCATION + EMPLOYMENT +
##     GENDER + CLAIM_REASON + LOCATION_CODE + MARITAL_STATUS +
##     VEHICLE_CLASS + VEHICLE_SIZE + INCOME + MONTHLY_PREMIUM,
##     data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -17.1001  -1.6959  -0.4657   1.6712  17.5809
##
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    1.155e+00  4.105e-01   2.814 0.004913 **
## COVERAGEExtended               2.348e-01  1.279e-01   1.836 0.066418 .
## COVERAGEPremium                4.991e-01  2.733e-01   1.826 0.067927 .
## EDUCATIONCollege              -1.737e-01  8.760e-02  -1.983 0.047355 *
## EDUCATIONDoctor               -4.596e-01  1.869e-01  -2.460 0.013929 *
## EDUCATIONHigh School or Below  1.671e-01  8.923e-02   1.872 0.061188 .
## EDUCATIONMaster               -2.126e-01  1.363e-01  -1.560 0.118892
## EMPLOYMENTEmployed            -5.675e-02  1.828e-01  -0.310 0.756220
## EMPLOYMENTMedical Leave        3.950e-01  2.263e-01   1.746 0.080875 .
## EMPLOYMENTRetired             -3.290e-01  2.497e-01  -1.318 0.187678
## EMPLOYMENTUnemployed           1.212e+00  1.843e-01   6.575 5.21e-11 ***
## GENDERM                        3.107e-01  6.835e-02   4.546 5.56e-06 ***
## CLAIM_REASONHail               1.983e-01  8.243e-02   2.406 0.016150 *
## CLAIM_REASONOther              5.432e-02  1.174e-01   0.463 0.643521
## CLAIM_REASONScratch/Dent       2.448e-01  1.022e-01   2.395 0.016652 *
## LOCATION_CODESuburban          1.184e+01  9.871e-02 119.928  < 2e-16 ***
## LOCATION_CODEUrban             8.043e+00  1.125e-01  71.520  < 2e-16 ***
## MARITAL_STATUSMarried         -1.093e-01  9.972e-02  -1.096 0.272909
## MARITAL_STATUSSingle           1.101e+00  1.143e-01   9.632  < 2e-16 ***
## VEHICLE_CLASSLuxury Car        3.194e-01  7.157e-01   0.446 0.655436
## VEHICLE_CLASSLuxury SUV       -3.803e-02  7.127e-01  -0.053 0.957454
## VEHICLE_CLASSSports Car        5.686e-01  2.748e-01   2.069 0.038574 *
## VEHICLE_CLASSSUV               6.663e-01  2.362e-01   2.821 0.004808 **
## VEHICLE_CLASSTwo-Door Car      5.357e-02  8.813e-02   0.608 0.543310
## VEHICLE_SIZEMedsize            3.163e-02  1.133e-01   0.279 0.780055
## VEHICLE_SIZESmall              3.417e-01  1.316e-01   2.597 0.009431 **
## INCOME                        -6.677e-06  1.981e-06  -3.370 0.000756 ***
## MONTHLY_PREMIUM                9.363e-02  5.035e-03  18.597  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.798 on 6822 degrees of freedom
## Multiple R-squared:  0.8341, Adjusted R-squared:  0.8335
## F-statistic:  1271 on 27 and 6822 DF,  p-value: < 2.2e-16
```

Adding the two numerical variables increases 1 percent in the proportion of variability in Y explained by the model. the small p-value shows that both these variables are significant in the model.

```
anova(fit2, fit3)
```

```
## Analysis of Variance Table
##
## Model 1: SQRT_TOTAL_CLAIM ~ COVERAGE + EDUCATION + EMPLOYMENT + GENDER +
##     CLAIM_REASON + LOCATION_CODE + MARITAL_STATUS + VEHICLE_CLASS +
##     VEHICLE_SIZE
## Model 2: SQRT_TOTAL_CLAIM ~ COVERAGE + EDUCATION + EMPLOYMENT + GENDER +
##     CLAIM_REASON + LOCATION_CODE + MARITAL_STATUS + VEHICLE_CLASS +
##     VEHICLE_SIZE + INCOME + MONTHLY_PREMIUM
##   Res.Df   RSS Df Sum of Sq      F    Pr(>F)
## 1   6824 56210
## 2   6822 53420  2    2790.2 178.16 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The anova analysis shows that model fit3 performs much better than model fit2. Now we explore the assumptions:

Independence assumption with durbin watson test:

```
library(car)
```

```
## Loading required package: carData
```

```
##
## Attaching package: 'car'
```

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

```
## The following object is masked from 'package:purrr':
##
##     some
```

```
dwt(fit3)
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1    -0.001097147      1.999783   0.998
##  Alternative hypothesis: rho != 0
```

It’s very close to 2 and large p value => our independence assumption is met.

```
#Checking multicolinearity
vif(fit3)
```

```
##                      GVIF Df GVIF^(1/(2*Df))
## COVERAGE         6.044328  2        1.567967
## EDUCATION        1.067521  4        1.008201
## EMPLOYMENT       3.467672  4        1.168166
## GENDER           1.021167  1        1.010528
## CLAIM_REASON     1.111120  3        1.017717
## LOCATION_CODE    1.404628  2        1.088655
## MARITAL_STATUS   1.257113  2        1.058872
## VEHICLE_CLASS   20.645922  5        1.353578
## VEHICLE_SIZE     1.054642  2        1.013389
## INCOME           3.142434  1        1.772691
## MONTHLY_PREMIUM 25.704005  1        5.069912
```

```
1/vif(fit3)
```

```
##                       GVIF        Df GVIF^(1/(2*Df))
## COVERAGE        0.16544437 0.5000000       0.6377684
## EDUCATION       0.93674943 0.2500000       0.9918658
## EMPLOYMENT      0.28837794 0.2500000       0.8560424
## GENDER          0.97927137 1.0000000       0.9895814
## CLAIM_REASON    0.89999284 0.3333333       0.9825919
## LOCATION_CODE   0.71193251 0.5000000       0.9185646
## MARITAL_STATUS  0.79547367 0.5000000       0.9444010
## VEHICLE_CLASS   0.04843571 0.2000000       0.7387825
## VEHICLE_SIZE    0.94818896 0.5000000       0.9867877
## INCOME          0.31822466 1.0000000       0.5641141
## MONTHLY_PREMIUM 0.03890444 1.0000000       0.1972421
```

```
mean(vif(fit3))
```

```
## [1] 3.334855
```

A VIF larger than 10 indicates multicolinearity. There seems to be multicolinearity between Vehicle Class and Monthly Premium. This makes sense. Keeping MONTHLY_PREMIUM gives a higher R-squared that keeping Vehicle Class. We keep MONTHLY_PREMIUM

```
fit4 = lm(SQRT_TOTAL_CLAIM ~  COVERAGE + EDUCATION + EMPLOYMENT + GENDER + CLAIM_REASON
+ LOCATION_CODE + MARITAL_STATUS + MONTHLY_PREMIUM + VEHICLE_SIZE + INCOME, data=train)
summary(fit4)
```

```
##
## Call:
## lm(formula = SQRT_TOTAL_CLAIM ~ COVERAGE + EDUCATION + EMPLOYMENT +
##     GENDER + CLAIM_REASON + LOCATION_CODE + MARITAL_STATUS +
##     MONTHLY_PREMIUM + VEHICLE_SIZE + INCOME, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.8586  -1.6951  -0.4345   1.7038  16.6683
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   8.989e-01  2.527e-01   3.557 0.000377 ***
## COVERAGEExtended              1.192e-01  7.920e-02   1.505 0.132243
## COVERAGEPremium               2.242e-01  1.362e-01   1.647 0.099617 .
## EDUCATIONCollege             -1.590e-01  8.769e-02  -1.814 0.069775 .
## EDUCATIONDoctor              -4.535e-01  1.873e-01  -2.421 0.015497 *
## EDUCATIONHigh School or Below  1.794e-01  8.931e-02   2.009 0.044601 *
## EDUCATIONMaster              -2.034e-01  1.366e-01  -1.489 0.136513
## EMPLOYMENTEmployed           -7.790e-02  1.832e-01  -0.425 0.670755
## EMPLOYMENTMedical Leave       3.645e-01  2.267e-01   1.608 0.107959
## EMPLOYMENTRetired            -3.310e-01  2.502e-01  -1.323 0.185915
## EMPLOYMENTUnemployed          1.210e+00  1.847e-01   6.550 6.15e-11 ***
## GENDERM                       3.157e-01  6.845e-02   4.612 4.06e-06 ***
## CLAIM_REASONHail              1.755e-01  8.251e-02   2.127 0.033464 *
## CLAIM_REASONOther            -1.195e-03  1.173e-01  -0.010 0.991870
## CLAIM_REASONScratch/Dent      2.261e-01  1.024e-01   2.207 0.027358 *
## LOCATION_CODESuburban         1.183e+01  9.889e-02 119.604  < 2e-16 ***
## LOCATION_CODEUrban            8.047e+00  1.127e-01  71.379  < 2e-16 ***
## MARITAL_STATUSMarried        -8.541e-02  9.986e-02  -0.855 0.392387
## MARITAL_STATUSSingle          1.106e+00  1.145e-01   9.654  < 2e-16 ***
## MONTHLY_PREMIUM               9.892e-02  1.135e-03  87.189  < 2e-16 ***
## VEHICLE_SIZEMedsize           4.682e-02  1.135e-01   0.413 0.679876
## VEHICLE_SIZESmall             3.421e-01  1.319e-01   2.593 0.009533 **
## INCOME                       -6.705e-06  1.986e-06  -3.376 0.000740 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.806 on 6827 degrees of freedom
## Multiple R-squared:  0.8331, Adjusted R-squared:  0.8325
## F-statistic:  1549 on 22 and 6827 DF,  p-value: < 2.2e-16
```

```
vif(fit4)
```

```
##                      GVIF Df GVIF^(1/(2*Df))
## COVERAGE         1.295593  2        1.066884
## EDUCATION        1.060239  4        1.007339
## EMPLOYMENT       3.456135  4        1.167680
## GENDER           1.018812  1        1.009362
## CLAIM_REASON     1.100356  3        1.016067
## LOCATION_CODE    1.401707  2        1.088089
## MARITAL_STATUS   1.251469  2        1.057682
## MONTHLY_PREMIUM 1.298255  1        1.139410
## VEHICLE_SIZE     1.051681  2        1.012677
## INCOME           3.140399  1        1.772117
```

There is still multicolinearity between INCOME and EMPLOYMENT. This also makes sense. Removing INCOME gives a better model thatn removing EMPLOYMENT.. (82.55 > 82.3 Rsqure). Thus we keep EMPLOYMENT in the model.

```
fit5 = lm(SQRT_TOTAL_CLAIM ~  COVERAGE + EDUCATION + GENDER + CLAIM_REASON + LOCATION_CO
DE + MARITAL_STATUS + MONTHLY_PREMIUM + VEHICLE_SIZE + EMPLOYMENT, data=train)
summary(fit5)
```

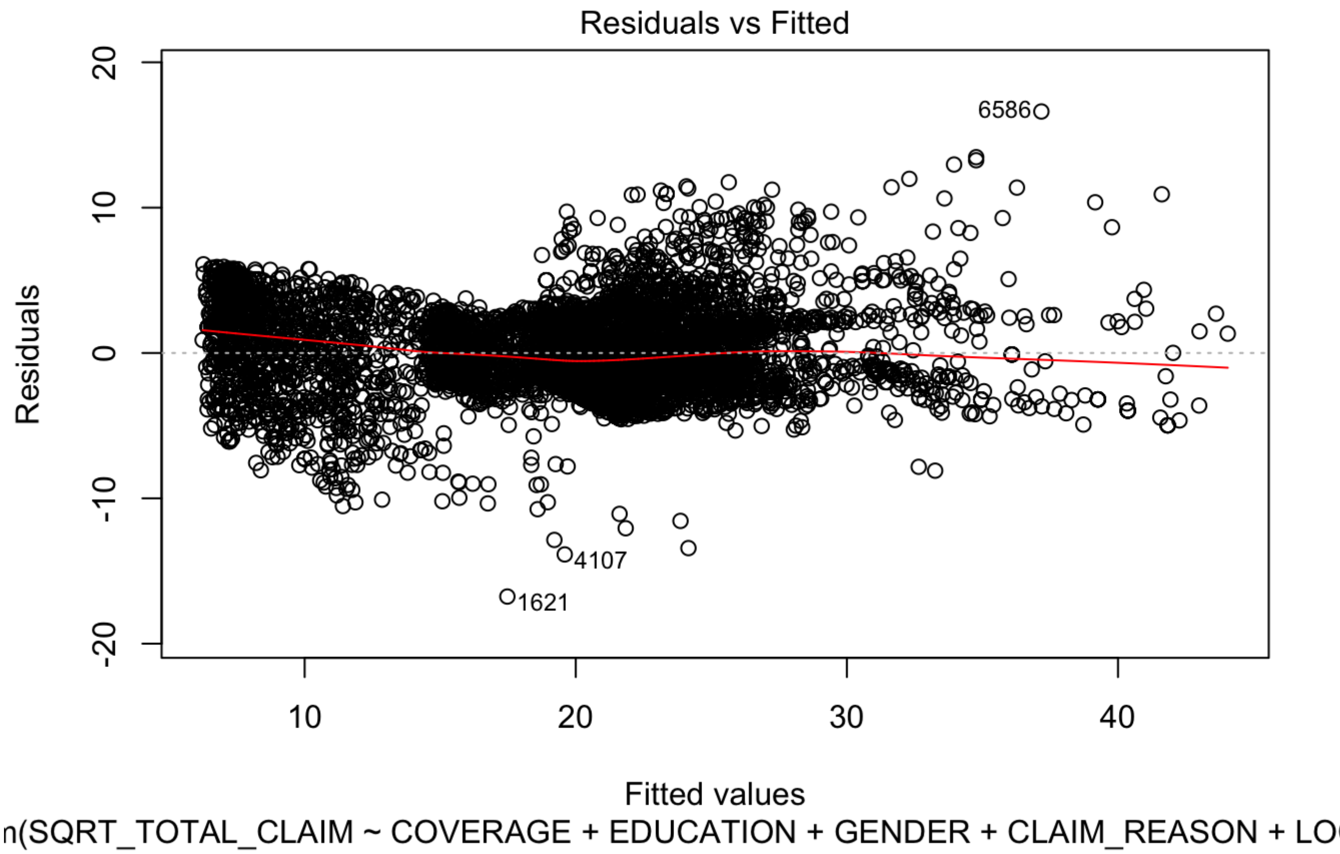
```
##
## Call:
## lm(formula = SQRT_TOTAL_CLAIM ~ COVERAGE + EDUCATION + GENDER +
##     CLAIM_REASON + LOCATION_CODE + MARITAL_STATUS + MONTHLY_PREMIUM +
##     VEHICLE_SIZE + EMPLOYMENT, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.7572  -1.6863  -0.4356   1.7202  16.6151
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    0.722357   0.247413   2.920  0.00352 **
## COVERAGEExtended               0.126893   0.079227   1.602  0.10928
## COVERAGEPremium                0.220067   0.136252   1.615  0.10633
## EDUCATIONCollege              -0.158772   0.087761  -1.809  0.07047 .
## EDUCATIONDoctor               -0.443695   0.187421  -2.367  0.01794 *
## EDUCATIONHigh School or Below  0.169954   0.089331   1.903  0.05715 .
## EDUCATIONMaster               -0.197864   0.136689  -1.448  0.14779
## GENDERM                        0.311564   0.068496   4.549 5.49e-06 ***
## CLAIM_REASONHail               0.154071   0.082327   1.871  0.06133 .
## CLAIM_REASONOther             -0.008390   0.117364  -0.071  0.94301
## CLAIM_REASONScratch/Dent       0.221997   0.102507   2.166  0.03037 *
## LOCATION_CODESuburban         11.888366   0.097350 122.120  < 2e-16 ***
## LOCATION_CODEUrban             8.051076   0.112811  71.368  < 2e-16 ***
## MARITAL_STATUSMarried         -0.080365   0.099924  -0.804  0.42127
## MARITAL_STATUSSingle           1.105942   0.114623   9.648  < 2e-16 ***
## MONTHLY_PREMIUM                0.098912   0.001135  87.113  < 2e-16 ***
## VEHICLE_SIZEMedsize            0.045499   0.113546   0.401  0.68865
## VEHICLE_SIZESmall              0.326371   0.131934   2.474  0.01339 *
## EMPLOYMENTEmployed            -0.294125   0.171823  -1.712  0.08698 .
## EMPLOYMENTMedical Leave        0.365407   0.226904   1.610  0.10736
## EMPLOYMENTRetired             -0.341569   0.250377  -1.364  0.17254
## EMPLOYMENTUnemployed           1.343772   0.180580   7.441 1.12e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.808 on 6828 degrees of freedom
## Multiple R-squared:  0.8328, Adjusted R-squared:  0.8323
## F-statistic:  1620 on 21 and 6828 DF,  p-value: < 2.2e-16
```

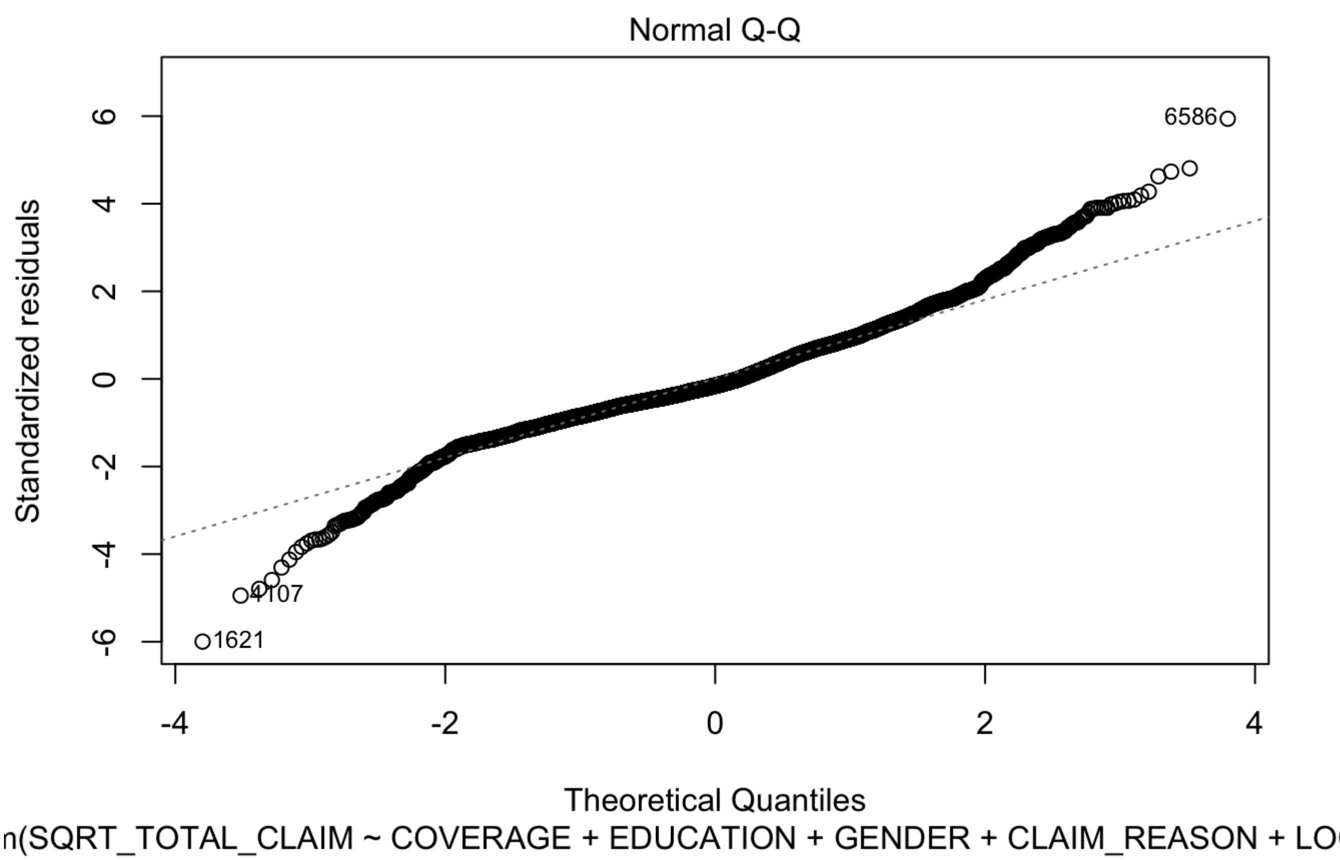
```
vif(fit5)
```

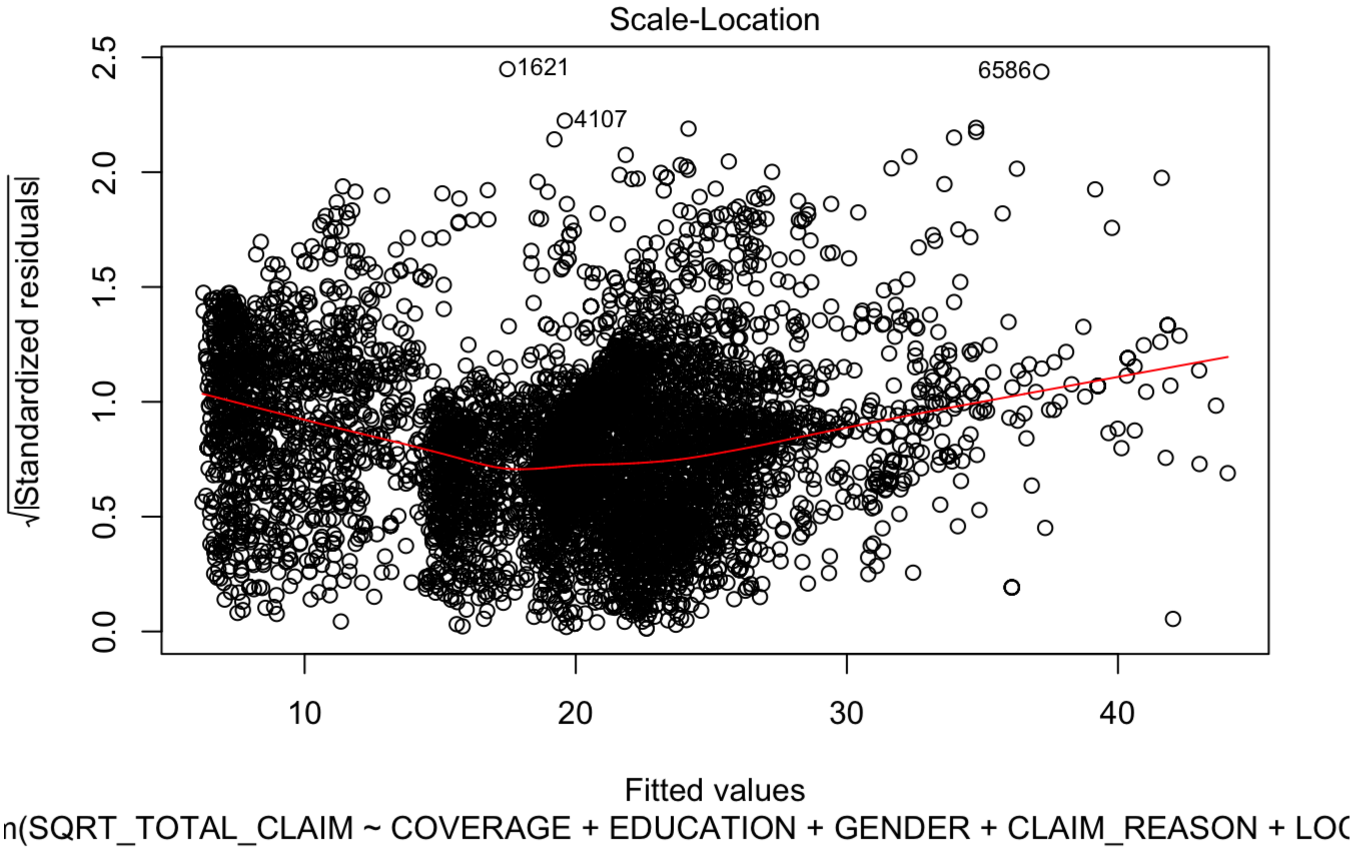
```
##                      GVIF Df GVIF^(1/(2*Df))
## COVERAGE        1.294082  2        1.066573
## EDUCATION       1.058040  4        1.007077
## GENDER          1.018481  1        1.009198
## CLAIM_REASON    1.093464  3        1.015003
## LOCATION_CODE   1.341603  2        1.076232
## MARITAL_STATUS  1.250977  2        1.057578
## MONTHLY_PREMIUM 1.298243  1        1.139405
## VEHICLE_SIZE    1.049136  2        1.012064
## EMPLOYMENT      1.505025  4        1.052429
```

Multicolinearity is all solved!

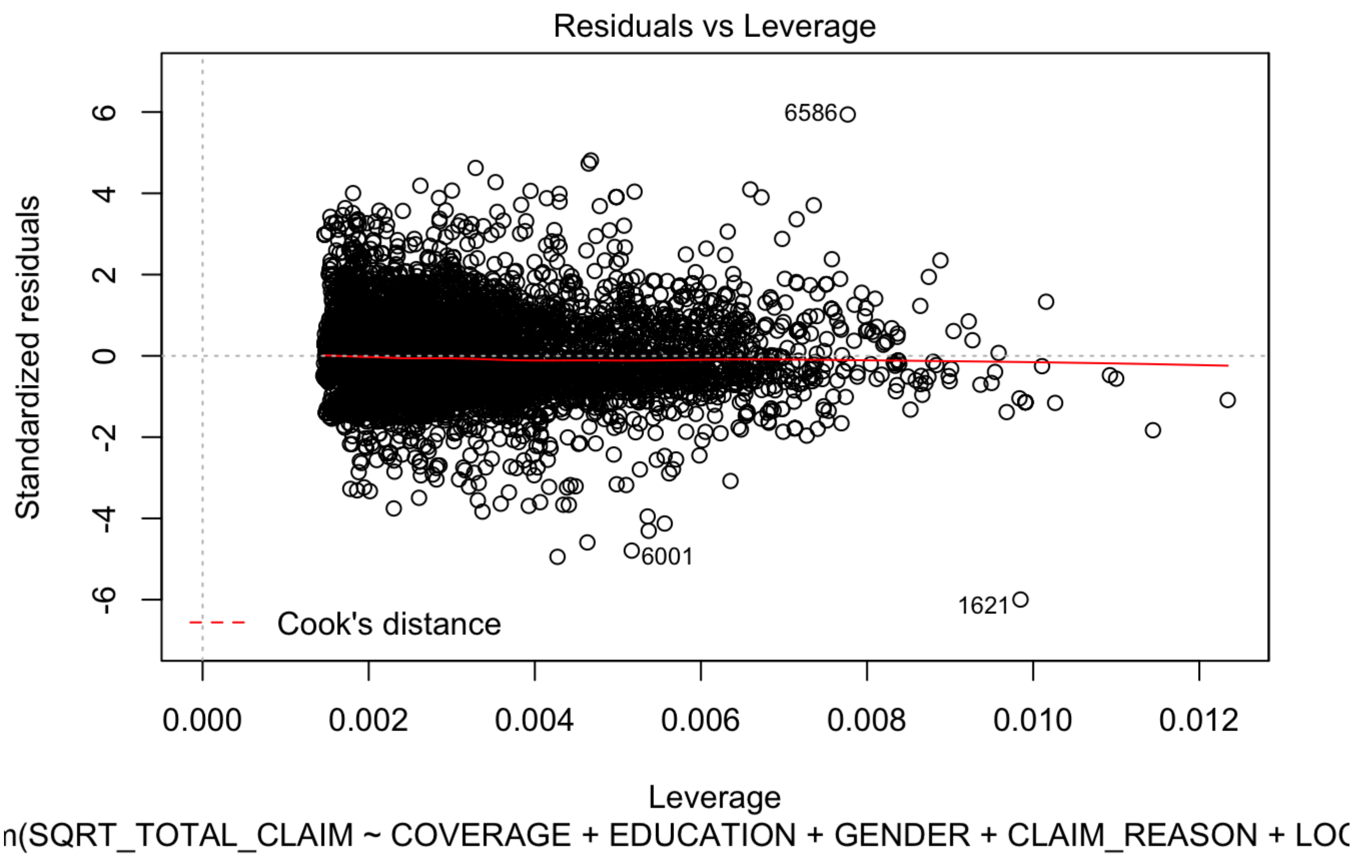
```
plot(fit5)
```


Residuals vs Fitted
Residuals
Fitted values
n(SQRT_TOTAL_CLAIM ~ COVERAGE + EDUCATION + GENDER + CLAIM_REASON + LOC
6586
4107
1621
20
10
0
-10
-20
10
20
30
40


Normal Q-Q
Standardized residuals
Theoretical Quantiles
n(SQRT_TOTAL_CLAIM ~ COVERAGE + EDUCATION + GENDER + CLAIM_REASON + LOC
6586
4107
1621
6
4
2
0
-2
-4
-6
-4
-2
0
2
4



Scale-Location
√|Standardized residuals|
1621
4107
6586
Fitted values
n(SQRT_TOTAL_CLAIM ~ COVERAGE + EDUCATION + GENDER + CLAIM_REASON + LOC



Residuals vs Leverage
Standardized residuals
6586
6001
1621
Cook's distance
Leverage
n(SQRT_TOTAL_CLAIM ~ COVERAGE + EDUCATION + GENDER + CLAIM_REASON + LOC

Residual plots for the assumptions should be acceptable. However, the relationship is not completely linear and there might be a better statistical model to fit the data.

```
predlm <- predict(fit5, test)
summary(predlm)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   6.137  16.229  20.453  19.739  23.528  41.834
```

```
summary(test$SQRT_TOTAL_CLAIM)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.6181 16.5846 19.8393 19.8028 23.3923 50.5207
```

```
library(ModelMetrics)
```

```
##
## Attaching package: 'ModelMetrics'
```

```
## The following objects are masked from 'package:caret':
##
##     confusionMatrix, precision, recall, sensitivity, specificity
```

```
## The following object is masked from 'package:base':
##
##     kappa
```

```
RMSE(test$SQRT_TOTAL_CLAIM, predlm)
```

```
## [1] 2.979767
```

4. Decision Tree - Conditional Inference Trees

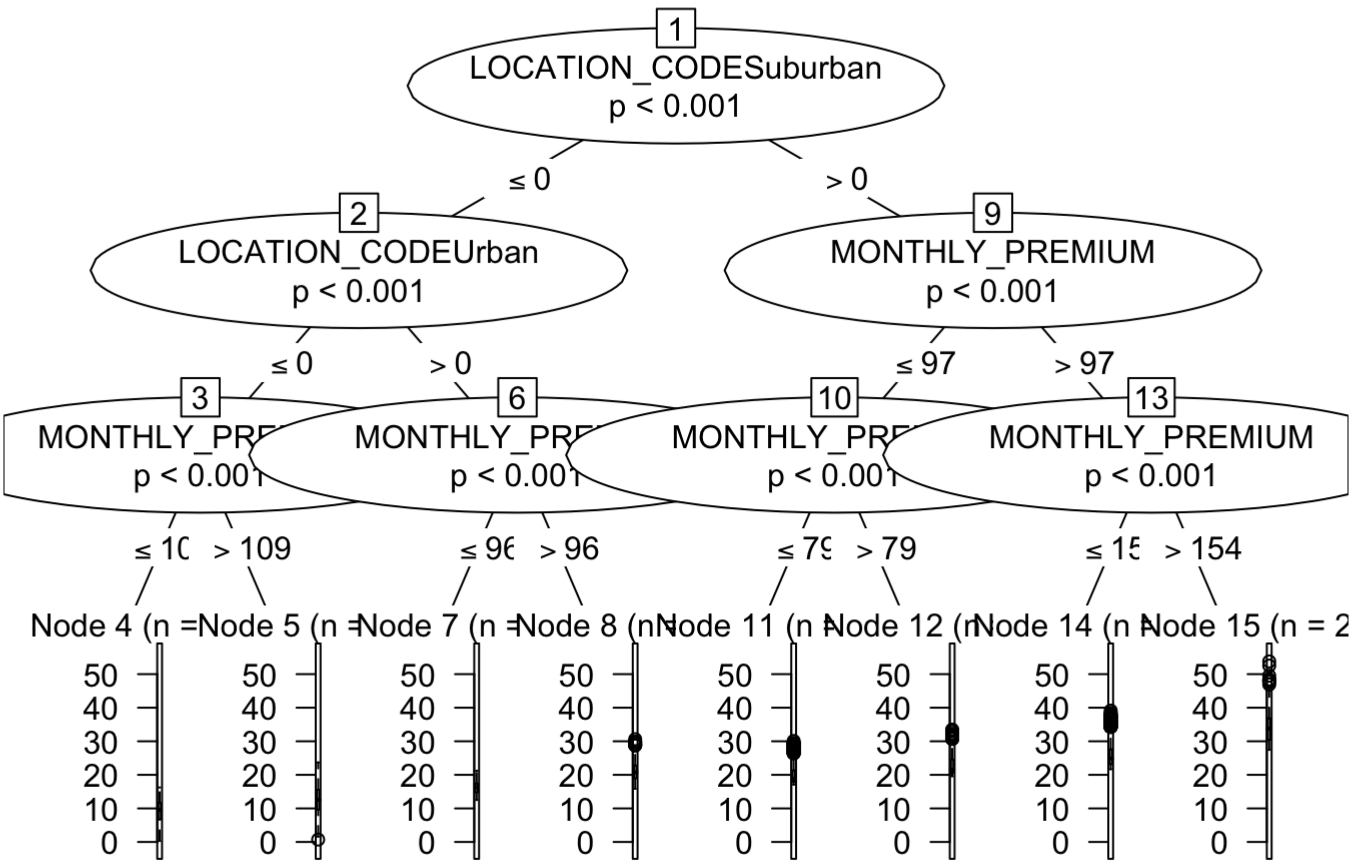
Conditional Inference Trees avoids the variable selection bias of normal decision trees (and related methods). They tend to select variables that have many possible splits or many missing values. Unlike the others, Conditional Inference Trees uses a significance test procedure in order to select variables instead of selecting the variable that maximizes an information measure (e.g. Gini coefficient).

The significance test, or better: the multiple significance tests computed at each start of the algorithm (select covariate - choose split - recurse) are permutation tests, that is, the “the distribution of the test statistic under the null hypothesis is obtained by calculating all possible values of the test statistic under rearrangements of the labels on the observed data points.” (from the wikipedia article).

(Source: Stack exchange https://stats.stackexchange.com/questions/12140/conditional-inference-trees-vs-traditional-decision-trees (https://stats.stackexchange.com/questions/12140/conditional-inference-trees-vs-traditional-decision-trees))

Since we are interested in a lot of categorical predictors, let's try conditional inference tree:

```
fit.tree <- train(
  SQRT_TOTAL_CLAIM ~ COVERAGE + EDUCATION + GENDER + CLAIM_REASON + LOCATION_CODE + MARI
TAL_STATUS + MONTHLY_PREMIUM + VEHICLE_SIZE + EMPLOYMENT, data = train, method = "ctree
2")
plot(fit.tree$finalModel)
```




```
pred.tree <- predict(fit.tree, test)
RMSE(test$SQRT_TOTAL_CLAIM, pred.tree)
```

```
## [1] 3.230054
```

The RMSE is higher compared to our fit5 multiple regression model. MONTHLY_PREMIUM is the variable that has the most possible split hence it appears in most of the nodes. Let's upgrade the tree to Random Forest.

5. Random Forest

Random forests are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set.

```r
library(randomForest)
```

```
## randomForest 4.6-14
```

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

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

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

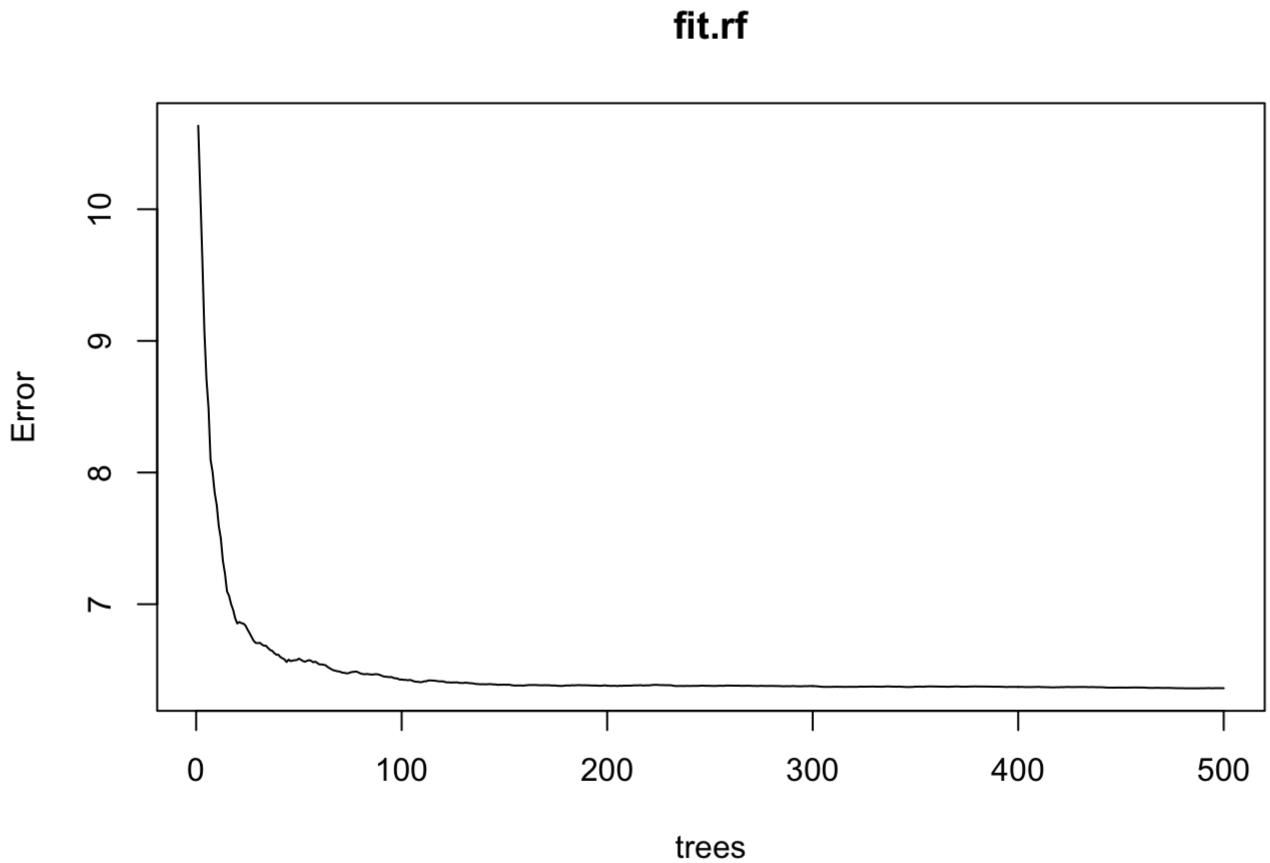
```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```r
train$SQRT_TOTAL_CLAIM <- as.numeric(train$SQRT_TOTAL_CLAIM)
train$INCOME <- as.numeric(train$INCOME)
train$MONTHLY_PREMIUM <- as.numeric(train$MONTHLY_PREMIUM)
fit.rf = randomForest(SQRT_TOTAL_CLAIM ~ COVERAGE + EDUCATION + GENDER + CLAIM_REASON +
LOCATION_CODE + MARITAL_STATUS + MONTHLY_PREMIUM + VEHICLE_SIZE + EMPLOYMENT, data=trai
n)
```

```r
fit.rf
```

```
##
## Call:
##  randomForest(formula = SQRT_TOTAL_CLAIM ~ COVERAGE + EDUCATION +      GENDER + CLAIM
_REASON + LOCATION_CODE + MARITAL_STATUS +      MONTHLY_PREMIUM + VEHICLE_SIZE + EMPLOYM
ENT, data = train)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 3
##
##           Mean of squared residuals: 6.361841
##                     % Var explained: 86.47
```

```r
plot(fit.rf)
```



mtry:

Number of variables randomly sampled as candidates at each split. ntree: Number of trees to grow.

The plot illustatres error rate as we average across more trees and shows that the error rate stabalizes with around 200 trees, and slowly decrease afterwards. Rsqured = 86.43 is better than the multiple regression model.

```
pred.rf <- predict(fit.rf, test)
RMSE(test$SQRT_TOTAL_CLAIM, pred.rf)
```

```
## [1] 2.684745
```

RMSE = 2.677 is also smaller than RMSE = 2.979 in our multiple regression model. This Random Forest model seems to be a better model to fit. Now let’s try tuning the parameters to see if we can achieve an even better Random Forest model

```
# number of trees with lowest MSE
which.min(fit.rf$mse)
```

```
## [1] 487
```

```
# RMSE of this optimal random forest
sqrt(fit.rf$mse[which.min(fit.rf$mse)])
```

```
## [1] 2.522043
```

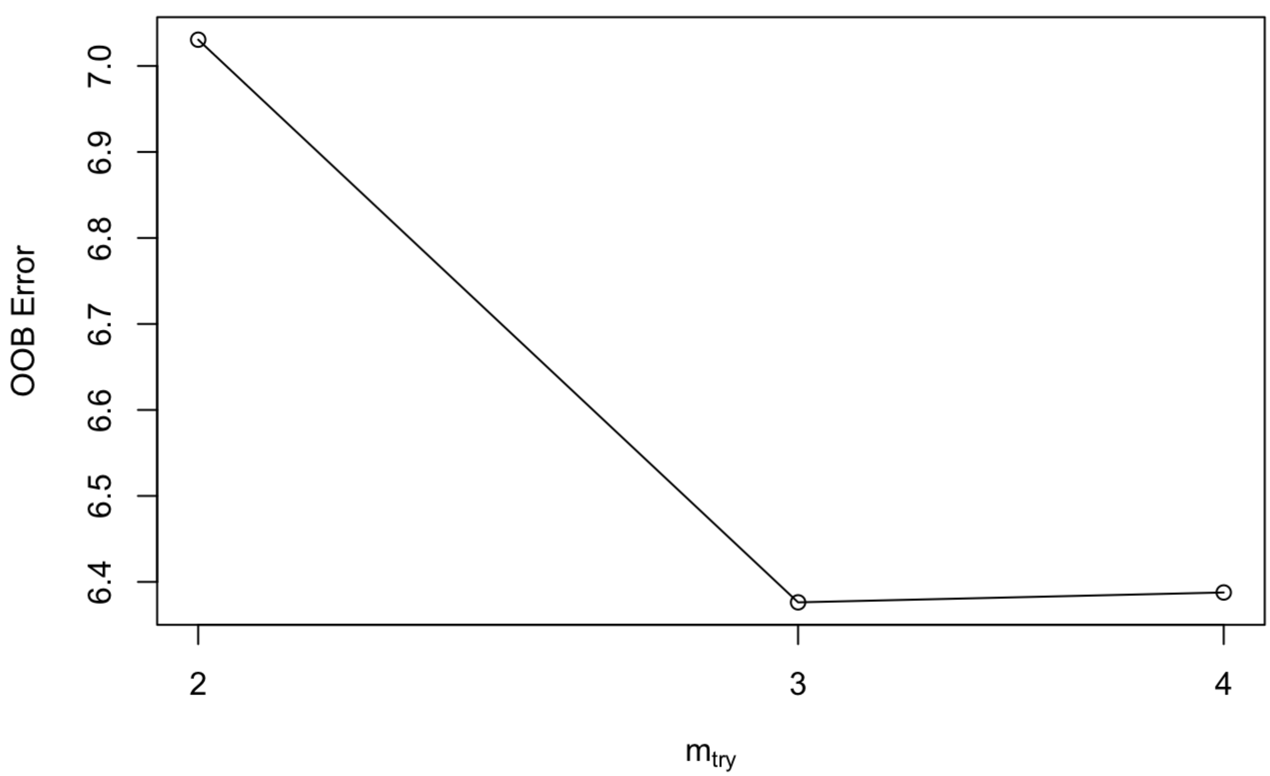
=> best ntree is 462, with RMSE = 462.

```
finalfeatures <- train[c(2,3,4,5,7,8,9,15,18)]
```

Let's use tuneRf for quick and easy tuning assesment. tuneRF will start at a value of mtry that is suppled and increase by a certain step factor until the OOB error stops improving be a specified amount.The below starts with mtry = 3, just as our default model started, and increases by a factor of 1.5 until the OOB error stops improving by 1%.

```
m2 <- tuneRF(
  x          = finalfeatures,
  y          = train$SQRT_TOTAL_CLAIM,
  ntreeTry   = 500,
  mtryStart  = 3,
  stepFactor = 1.5,
  improve    = 0.01,
  trace      = FALSE
)
```

```
## -0.10262 0.01
## -0.001801799 0.01
```

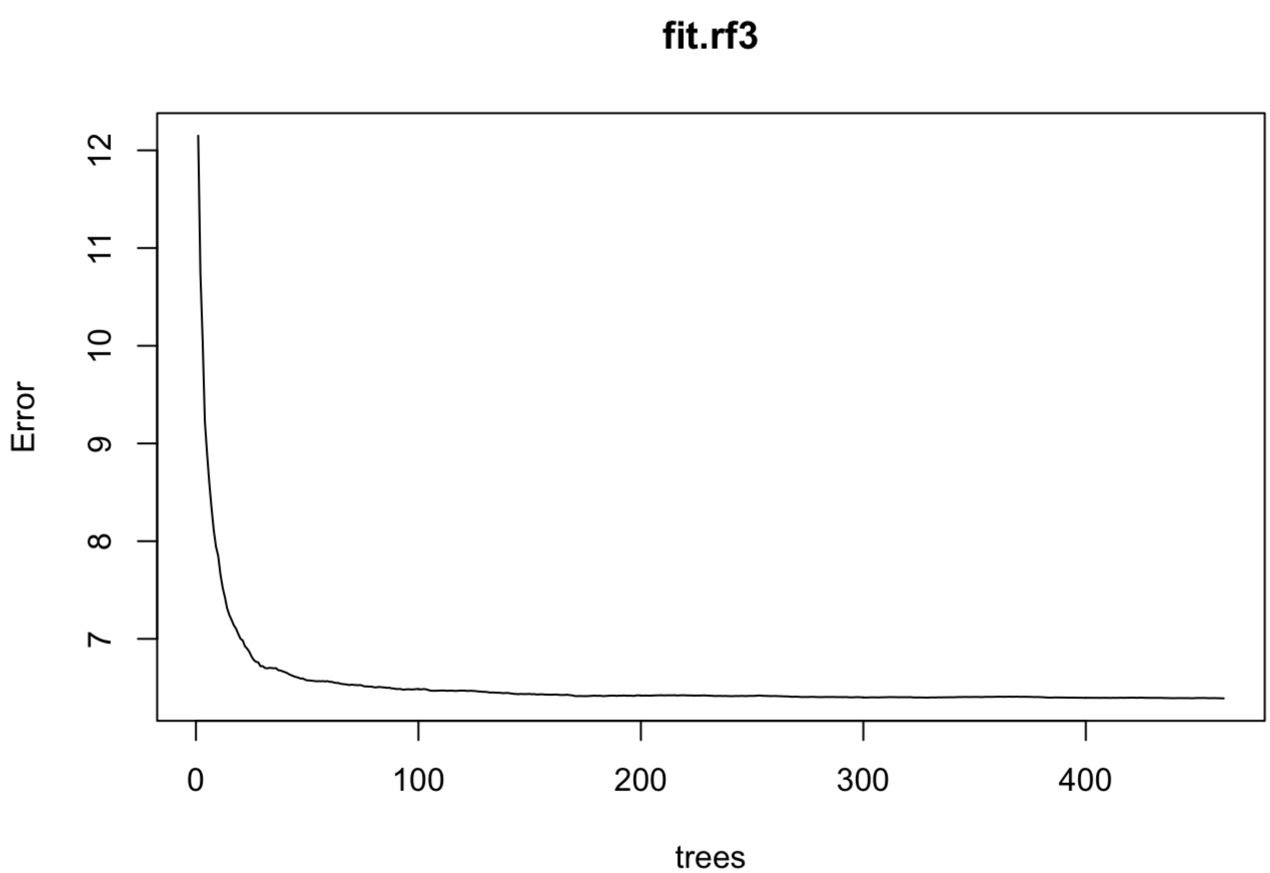


=> best mtry is 3, just as our default model.

```
fit.rf3 = randomForest(SQRT_TOTAL_CLAIM ~ COVERAGE + EDUCATION + GENDER + CLAIM_REASON +
LOCATION_CODE + MARITAL_STATUS + MONTHLY_PREMIUM + VEHICLE_SIZE + EMPLOYMENT, data=trai
n, ntree=462)
fit.rf3
```

```
##
## Call:
##  randomForest(formula = SQRT_TOTAL_CLAIM ~ COVERAGE + EDUCATION +      GENDER + CLAIM
_REASON + LOCATION_CODE + MARITAL_STATUS +      MONTHLY_PREMIUM + VEHICLE_SIZE + EMPLOYM
ENT, data = train,      ntree = 462)
##                Type of random forest: regression
##                      Number of trees: 462
## No. of variables tried at each split: 3
##
##           Mean of squared residuals: 6.391302
##                     % Var explained: 86.41
```

```
plot(fit.rf3)
```

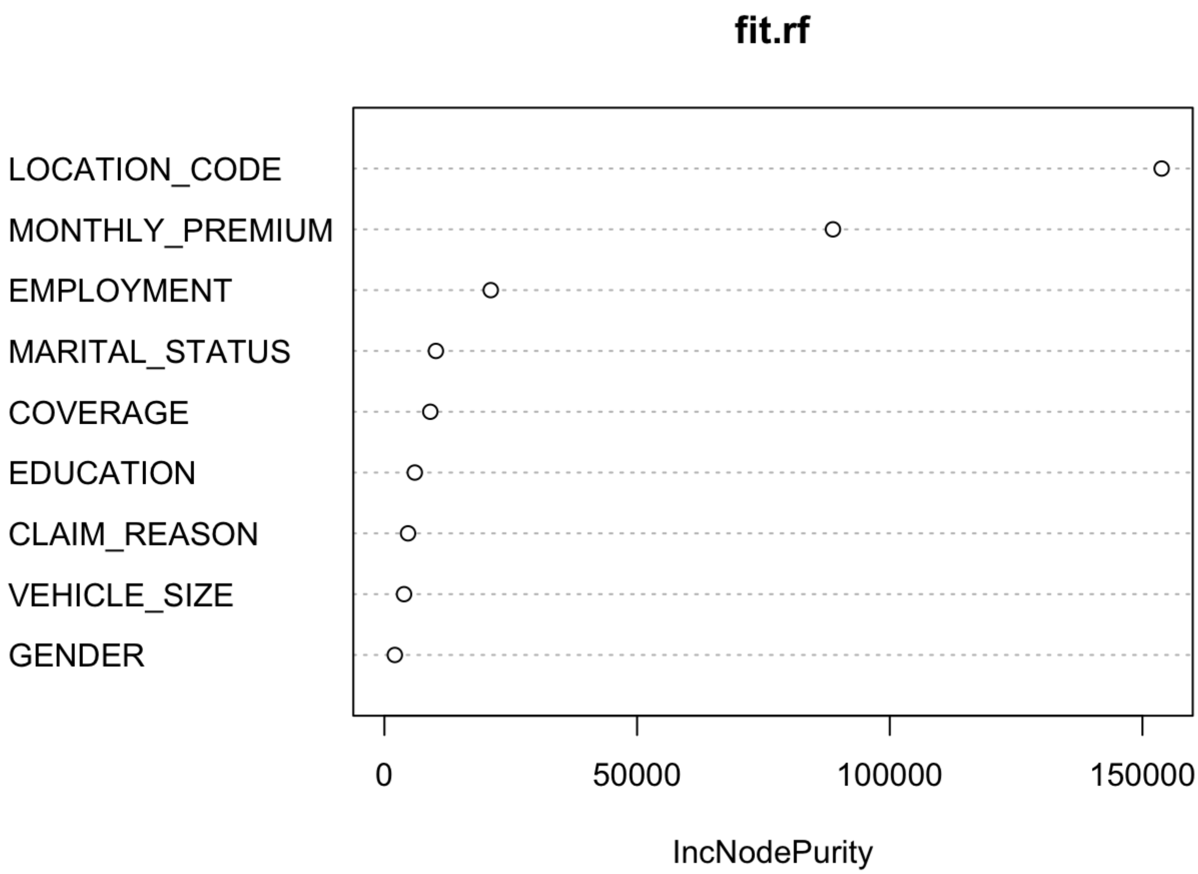



```
pred.rf3 <- predict(fit.rf3, test)
RMSE(test$SQRT_TOTAL_CLAIM, pred.rf3)
```

```
## [1] 2.683257
```

% var explained has slightly decreased and RMSE has slightly increase. Let's stick to the original model fit.rf.

```
varImpPlot(fit.rf)
```



Variable importance plot. It's interesting that Location Code is the most important variable, followed by Monthly Premium and Employment. Marital Status, Coverage, Education, Claim Reason and Vehicle Size all add a smaller amount of importance to the model. Gender doesn't seem to be that predictive.

5. Conclusion

A multiple regression has been fitted, explaining but since the relationship between the is not completely linear, a better type of model might be better. Conditional Inference Trees and Random Forest are briefly explored. We conclude that out Random Forest model provides the best fit and prediction.