Auto Insurance Claims Analysis

Arian Farahani

2024-08-01

Objective

The goal of this analysis is to identify and analyze the key factors that influence the amount of insurance claims.

Questions to Answer:

1. What are the most significant predictors of claim amounts?
2. How do different policy types and customer demographics affect claim amounts?

Load Data

```r
# Import the data set
autoClaims <- read_csv("AutoClaims.csv")
```

```
## Rows: 6773 Columns: 6
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (3): STATE, CLASS, GENDER
## dbl (3): Index, AGE, PAID
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Exploratory Data Analysis (EDA)

Descriptive Statistics

```r
# Summary statistics for all variables
summary(autoClaims)
```

```
##      Index         STATE              CLASS              GENDER
##  Min.   :   1   Length:6773        Length:6773        Length:6773
##  1st Qu.:1694   Class :character   Class :character   Class :character
##  Median :3387   Mode  :character   Mode  :character   Mode  :character
##  Mean   :3387
##  3rd Qu.:5080
##  Max.   :6773
##       AGE             PAID
##  Min.   :50.00   Min.   :    9.5
##  1st Qu.:54.00   1st Qu.:  523.7
##  Median :62.00   Median : 1001.7
##  Mean   :63.81   Mean   : 1853.0
```

```
##   3rd Qu.:72.00   3rd Qu.: 2137.4
##   Max.   :97.00   Max.   :60000.0
```

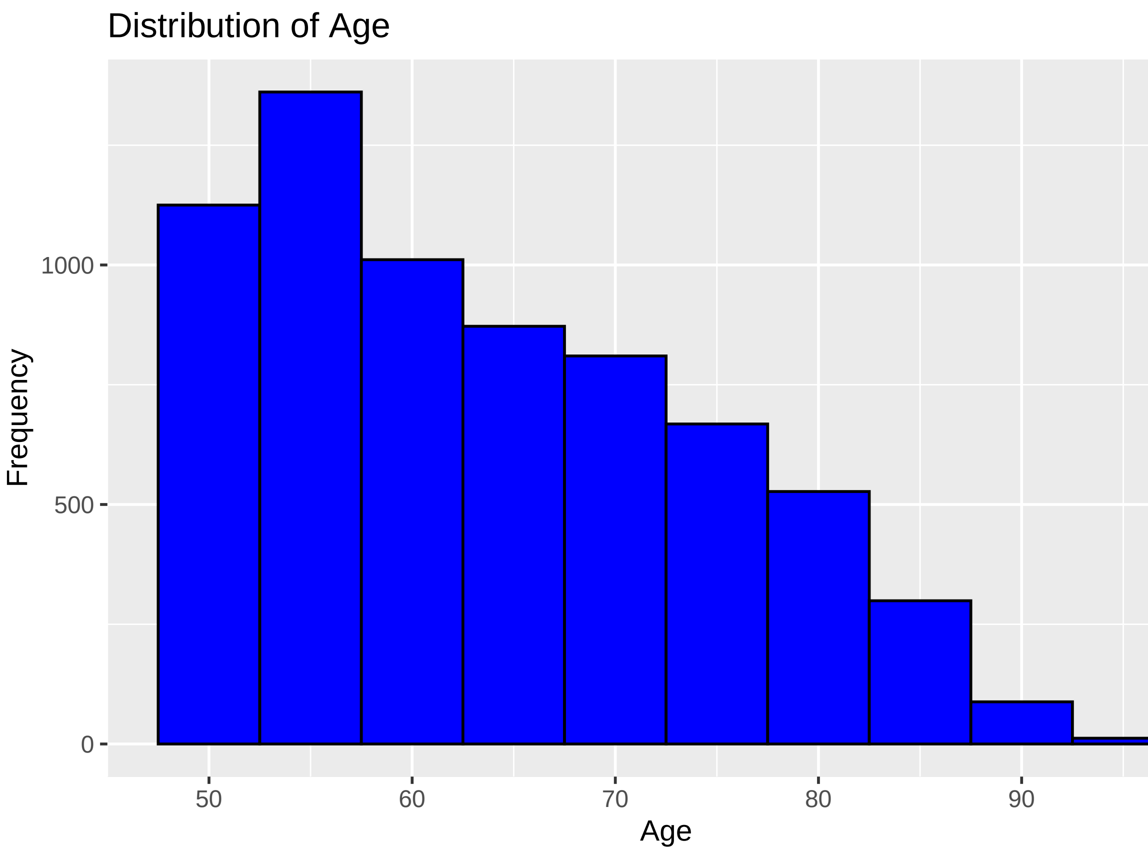
Handle Missing Values

```
any_na <- any(is.na(autoClaims))  # Check for missing values
if(any_na) {
  clean_autoClaims <- na.omit(autoClaims)  # Remove missing values if present
  print("Data contained missing values and they were removed.")
} else {
  # Use original data set if no missing values
  print("Data is clean with no missing values.")
}
```

```
## [1] "Data is clean with no missing values."
```

Distribution Plots

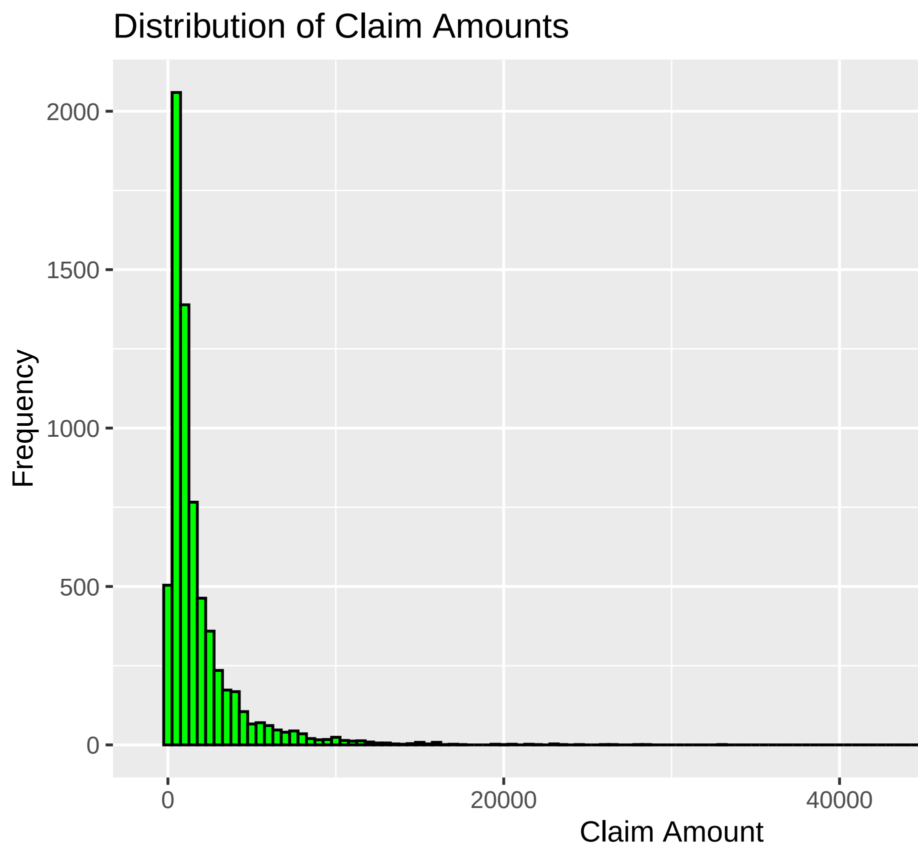
```
ggplot(autoClaims, aes(x = AGE)) +
  geom_histogram(binwidth = 5, fill = "blue", color = "black") +
  labs(title = "Distribution of Age", x = "Age", y = "Frequency")
```



Histogram for AGE

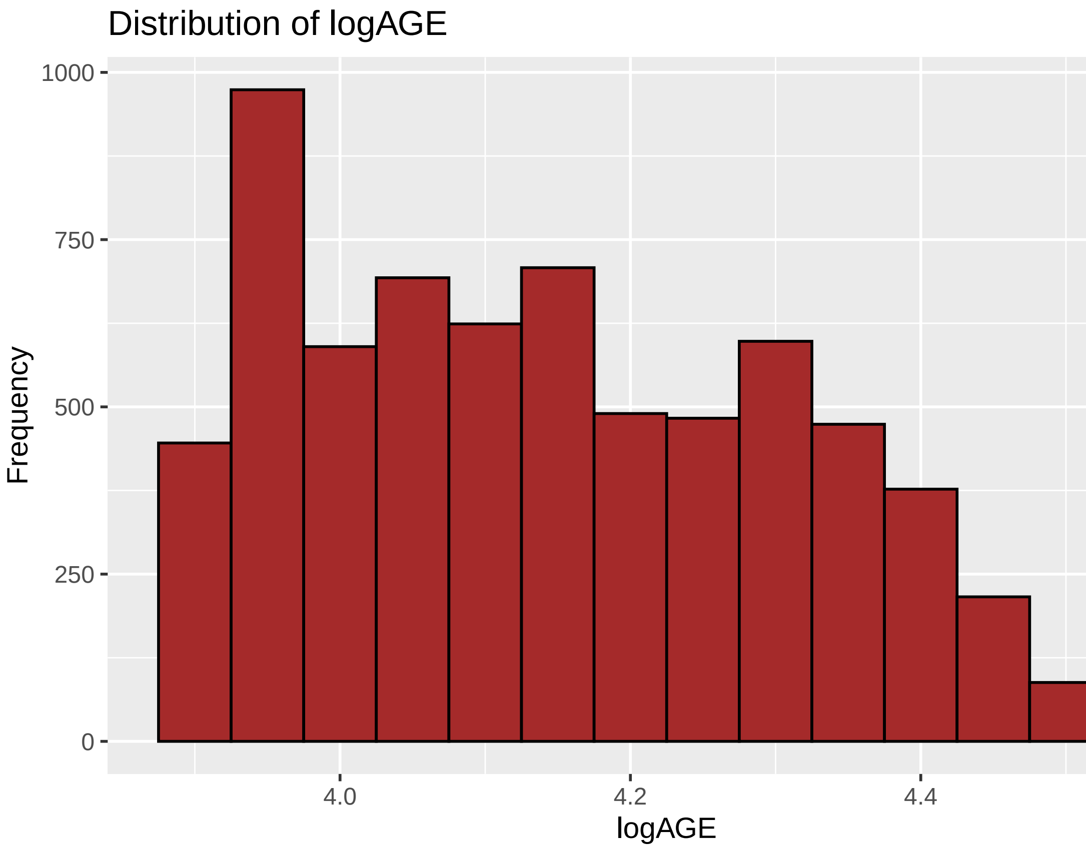
```
ggplot(autoClaims, aes(x = PAID)) +
  geom_histogram(binwidth = 500, fill = "green", color = "black") +
```

```r
  labs(title = "Distribution of Claim Amounts", x = "Claim Amount", y = "Frequency")
```



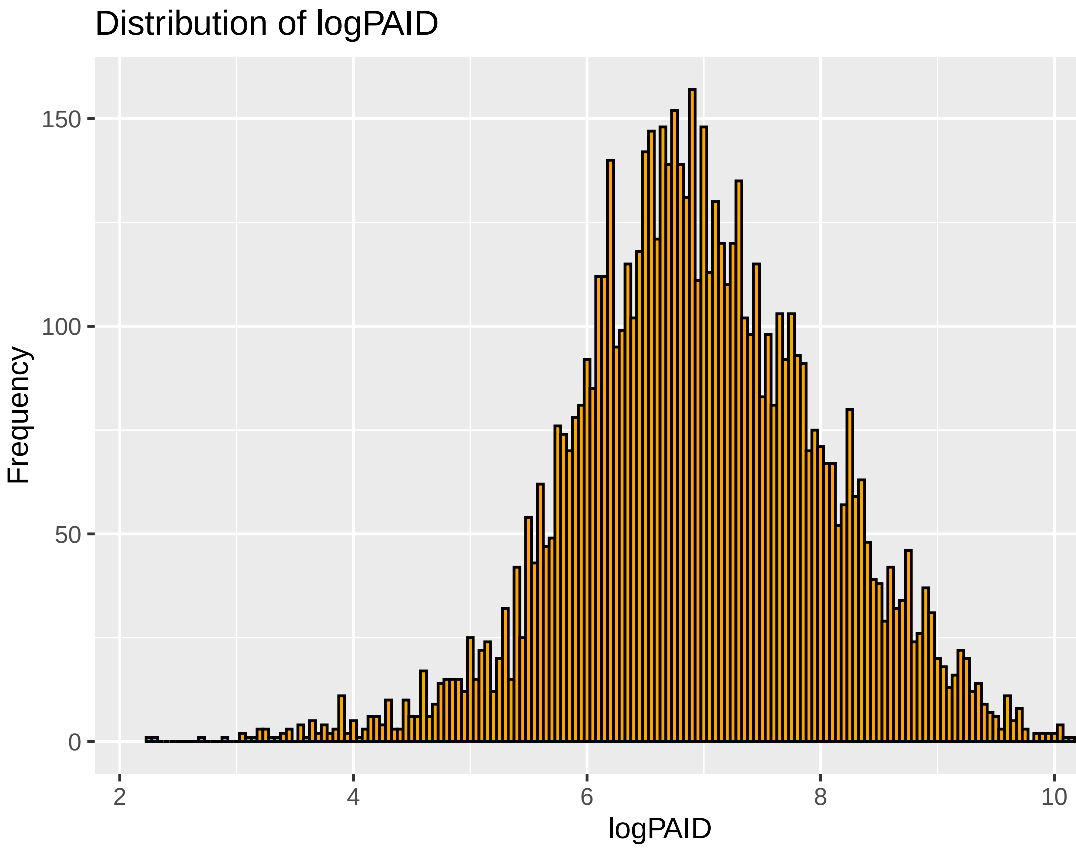
Histogram for PAID (Claim Amount)

```r
# distribution of log AGE
ggplot(autoClaims, aes(x = log(AGE))) +
  geom_histogram(binwidth = .05,
                 fill = "brown",
                 color = "black") +
  labs(title = "Distribution of logAGE",
       x = "logAGE",
       y = "Frequency")
```



Histogram for log(AGE)

```
# distribution of log PAID
ggplot(autoClaims, aes(x = log(PAID))) +
  geom_histogram(binwidth = .05,
                 fill = "orange",
                 color = "black") +
  labs(title = "Distribution of logPAID",
       x = "logPAID",
       y = "Frequency")
```

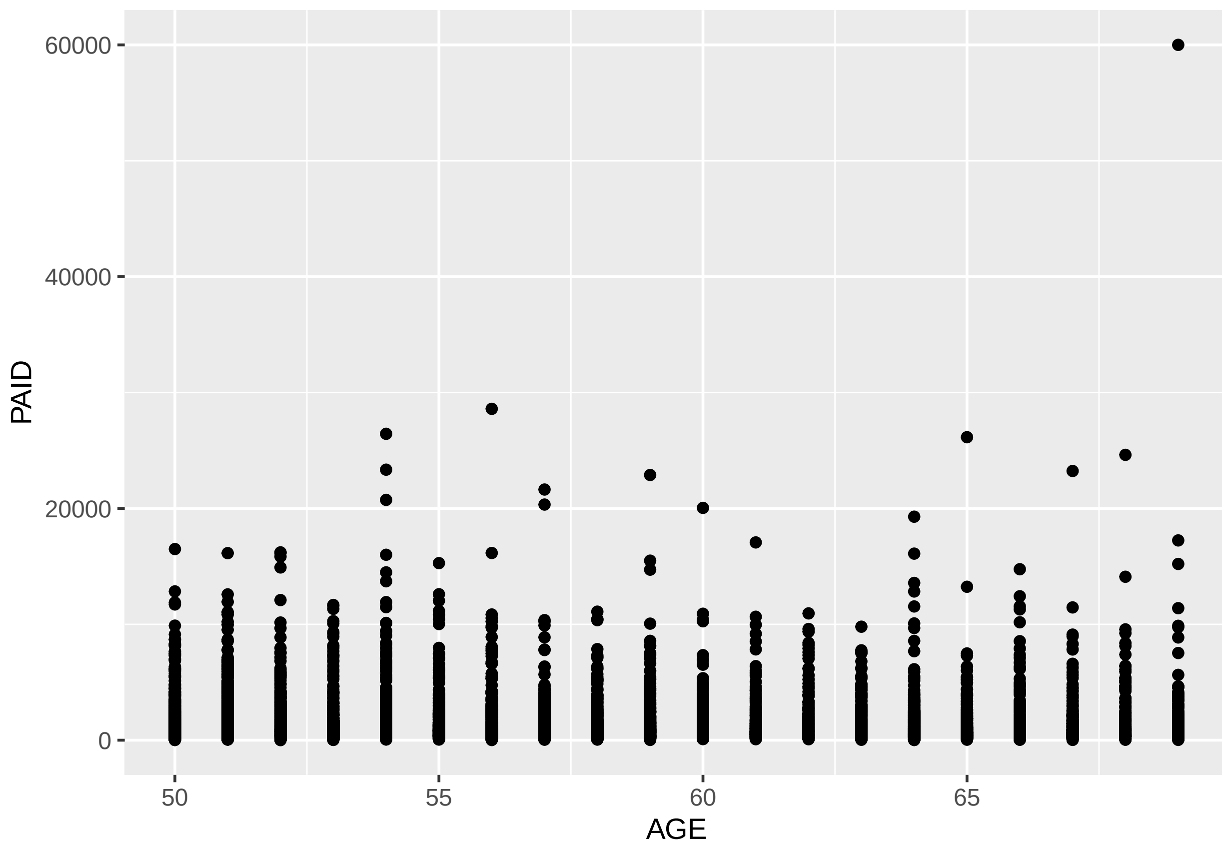


Histogram for log(PAID)

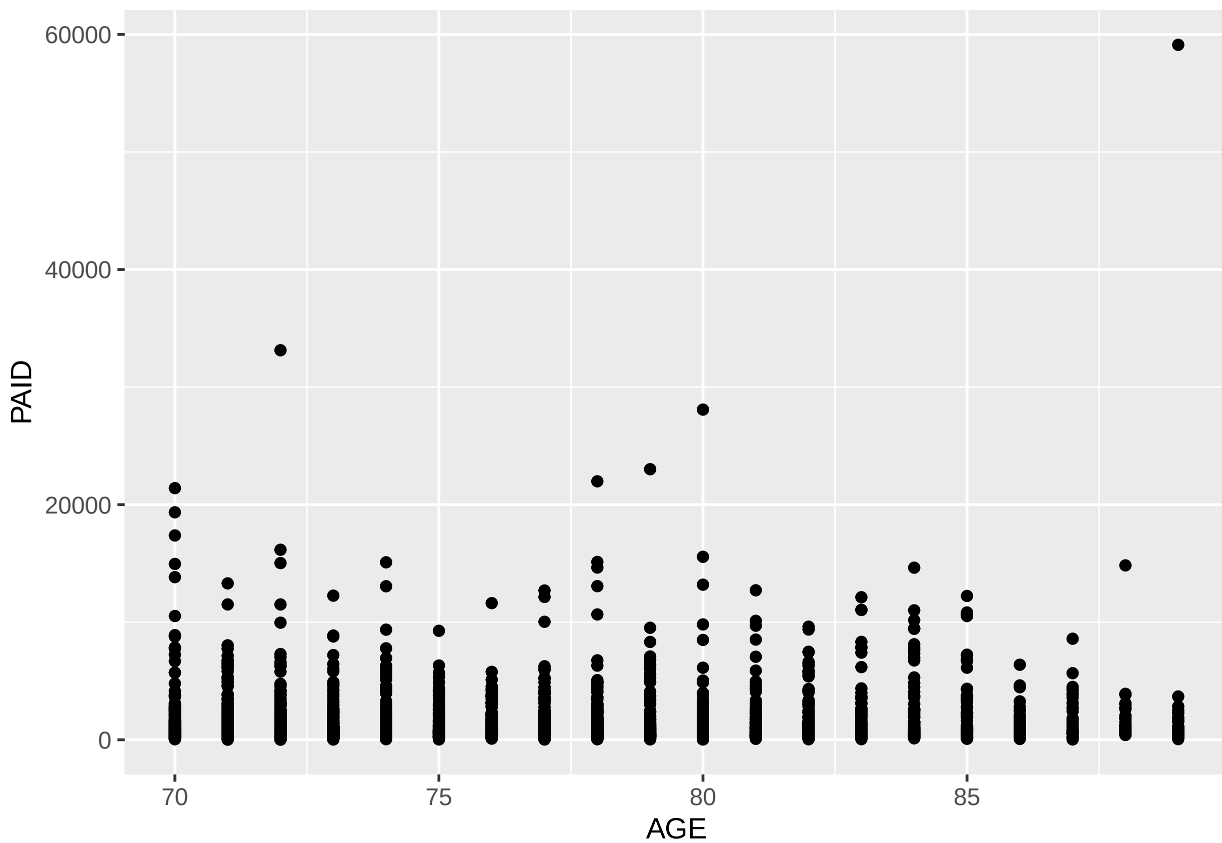
Feature Engineering

Subset Data by Age Groups

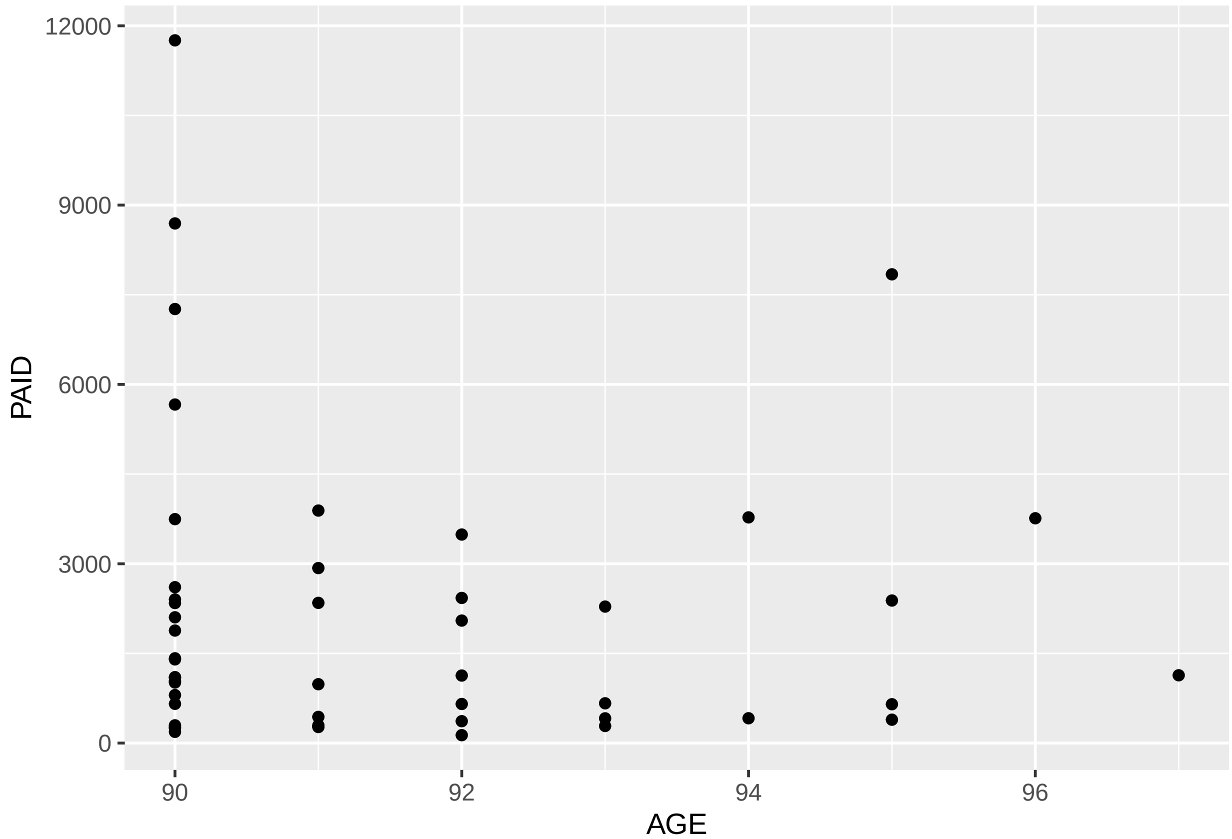
```
# Subset data into specific age ranges
fiftiesToSixties <- autoClaims %>%
  filter(AGE >= 50 & AGE < 70)

seventiesToEighties <- autoClaims %>%
  filter(AGE >= 70 & AGE < 90)

nineties <- autoClaims %>%
  filter(AGE >= 90)

# Graph each age group to claims
ggplot(fiftiesToSixties, aes(AGE, PAID)) +
  geom_point()  # Plot points for fifties to sixties
```




```
ggplot(seventiesToEighties, aes(AGE, PAID)) +
  geom_point()  # Plot points for seventies to eighties
```

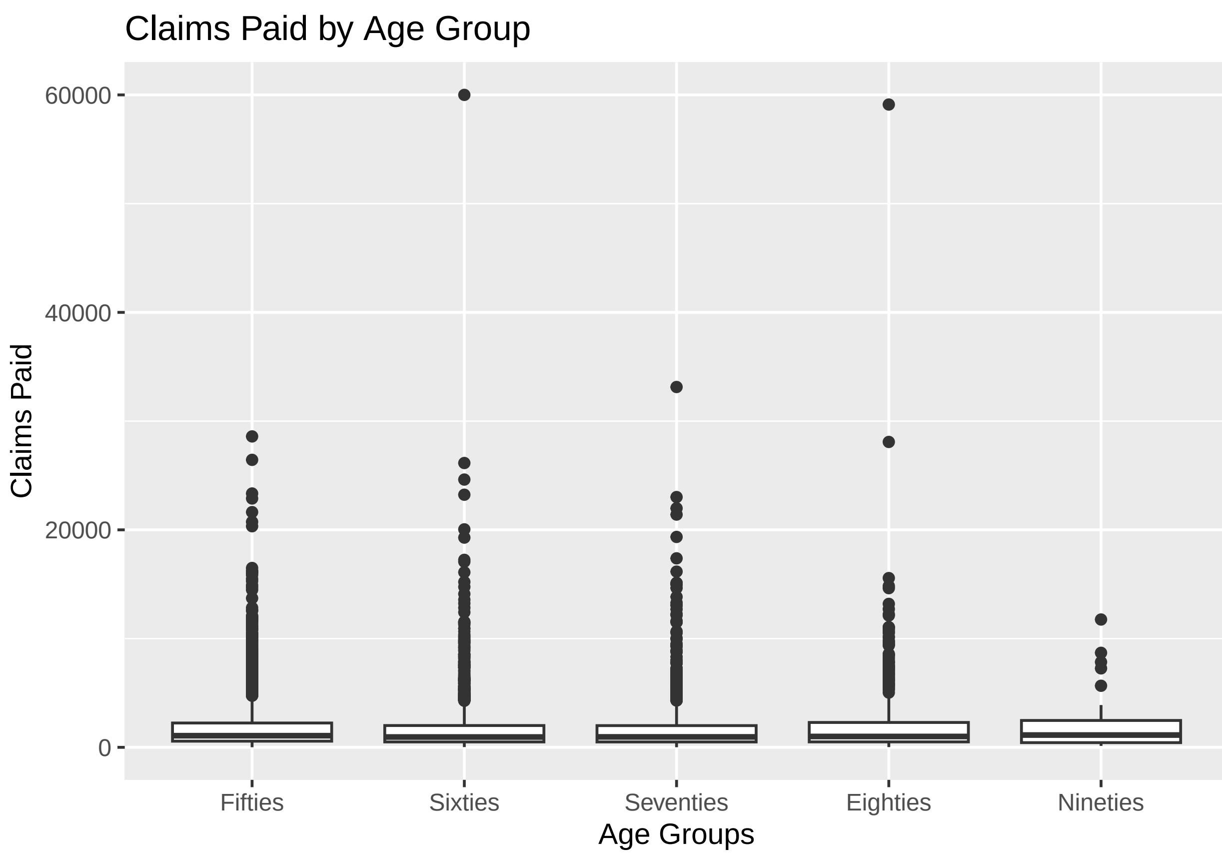



```
ggplot(nineties, aes(AGE, PAID)) +
  geom_point()   # Plot points for nineties
```



Create Age Groups

```
autoClaims <- autoClaims %>%
  mutate(age_group = case_when(
    AGE >= 50 & AGE < 60 ~ "Fifties",
    AGE >= 60 & AGE < 70 ~ "Sixties",
    AGE >= 70 & AGE < 80 ~ "Seventies",
    AGE >= 80 & AGE < 90 ~ "Eighties",
    AGE >= 90 ~ "Nineties"
  ))  # Create age group categories

# Ensure age groups are factors with correct order
myLevels <- c("Fifties", "Sixties", "Seventies", "Eighties", "Nineties")
autoClaims$age_group <- ordered(autoClaims$age_group, levels = myLevels)

# Visualize claim amounts by age group
ggplot(autoClaims, aes(age_group, PAID)) +
  geom_boxplot() +
  labs(title = "Claims Paid by Age Group", x = "Age Groups", y = "Claims Paid")
```



Summary Statistics by Age Group

```r
# Summary statistics for each age group
summary(autoClaims[autoClaims$age_group == "Fifties", "PAID"])
```

```
##       PAID
##  Min.   :    9.5
##  1st Qu.:  567.5
##  Median : 1071.4
##  Mean   : 1890.8
##  3rd Qu.: 2239.4
##  Max.   :28593.5
```

```r
summary(autoClaims[autoClaims$age_group == "Sixties", "PAID"])
```

```
##       PAID
##  Min.   :   25.0
##  1st Qu.:  498.3
##  Median :  950.0
##  Mean   : 1776.3
##  3rd Qu.: 2006.8
##  Max.   :60000.0
```

```r
summary(autoClaims[autoClaims$age_group == "Seventies", "PAID"])
```

```
##       PAID
##  Min.   :   10.0
##  1st Qu.:  496.2
##  Median :  959.1
```

```
##  Mean   : 1769.4
##  3rd Qu.: 1997.1
##  Max.   :33137.5
```

```r
summary(autoClaims[autoClaims$age_group == "Eighties", "PAID"])
```

```
##       PAID
##  Min.   :   25.0
##  1st Qu.:  500.4
##  Median : 1000.0
##  Mean   : 2049.0
##  3rd Qu.: 2288.0
##  Max.   :59113.8
```

```r
summary(autoClaims[autoClaims$age_group == "Nineties", "PAID"])
```

```
##       PAID
##  Min.   :  132.5
##  1st Qu.:  432.4
##  Median : 1132.2
##  Mean   : 2153.9
##  3rd Qu.: 2473.0
##  Max.   :11756.3
```

Dummification of Categorical Variables

```r
# Convert categorical variables into dummy variables and remove original columns
autoClaims_dummies <- autoClaims %>%
  mutate(across(c(CLASS, STATE, GENDER, age_group), as.factor)) %>%
  fastDummies::dummy_cols(select_columns = c("CLASS", "STATE", "GENDER", "age_group"),
                          remove_first_dummy = TRUE) %>%
  dplyr::select(-Index,-STATE, -CLASS, -GENDER, -age_group)  # Remove the original categorical columns

# View the modified data set with dummy variables
head(autoClaims_dummies)
```

```
## # A tibble: 6 x 36
##     AGE  PAID CLASS_C11 CLASS_C1A CLASS_C1B CLASS_C1C CLASS_C2 CLASS_C6 CLASS_C7
##   <dbl> <dbl>     <int>     <int>     <int>     <int>    <int>    <int>    <int>
## 1    97 1134.         0         0         0         0        0        1        0
## 2    96 3761.         0         0         0         0        0        1        0
## 3    95 7842.         1         0         0         0        0        0        0
## 4    95 2385.         0         0         0         0        0        0        0
## 5    95  650          0         0         0         0        0        0        0
## 6    95  391.         0         0         0         0        0        0        0
## # i 27 more variables: CLASS_C71 <int>, CLASS_C72 <int>, CLASS_C7A <int>,
## #   CLASS_C7B <int>, CLASS_C7C <int>, CLASS_F1 <int>, CLASS_F11 <int>,
## #   CLASS_F6 <int>, CLASS_F7 <int>, CLASS_F71 <int>, `STATE_STATE 02` <int>,
## #   `STATE_STATE 03` <int>, `STATE_STATE 04` <int>, `STATE_STATE 06` <int>,
## #   `STATE_STATE 07` <int>, `STATE_STATE 10` <int>, `STATE_STATE 11` <int>,
## #   `STATE_STATE 12` <int>, `STATE_STATE 13` <int>, `STATE_STATE 14` <int>,
## #   `STATE_STATE 15` <int>, `STATE_STATE 17` <int>, GENDER_M <int>, ...
```

Interaction and Polynomial Features

```r
# Generate interaction terms and polynomial features
autoClaims_dummies <- autoClaims_dummies %>%
  mutate(
    AGE_CLASS = AGE * as.numeric(CLASS_C11),  # Example interaction with one CLASS variable
    AGE_GENDER = AGE * as.numeric(GENDER_M),
    AGE_STATE = AGE * as.numeric(`STATE_STATE 02`),  # Example with one STATE variable
    AGE_AGE = poly(AGE, 2, raw = TRUE)
  )
```

Advanced Modeling

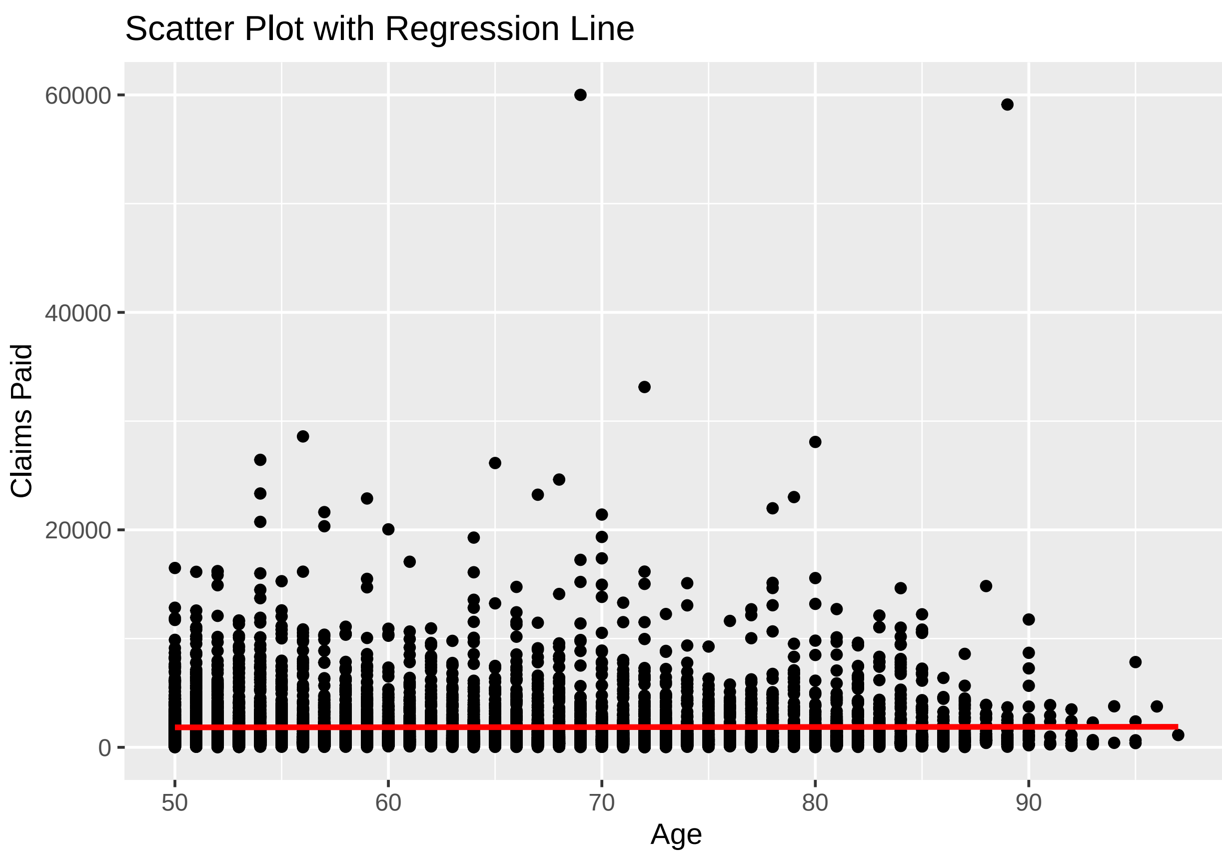
Simple Linear Regression

```r
# Linear regression model with AGE as the predictor
claimsModel <- lm(PAID ~ AGE, data = autoClaims)
summary(claimsModel)
```

```
##
## Call:
## lm(formula = PAID ~ AGE, data = autoClaims)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -1851  -1329   -848    280  58142
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1786.738    195.031   9.161   <2e-16 ***
## AGE            1.039      3.015   0.345     0.73
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2647 on 6771 degrees of freedom
## Multiple R-squared:  1.754e-05,  Adjusted R-squared:  -0.0001301
## F-statistic: 0.1188 on 1 and 6771 DF,  p-value: 0.7304
```

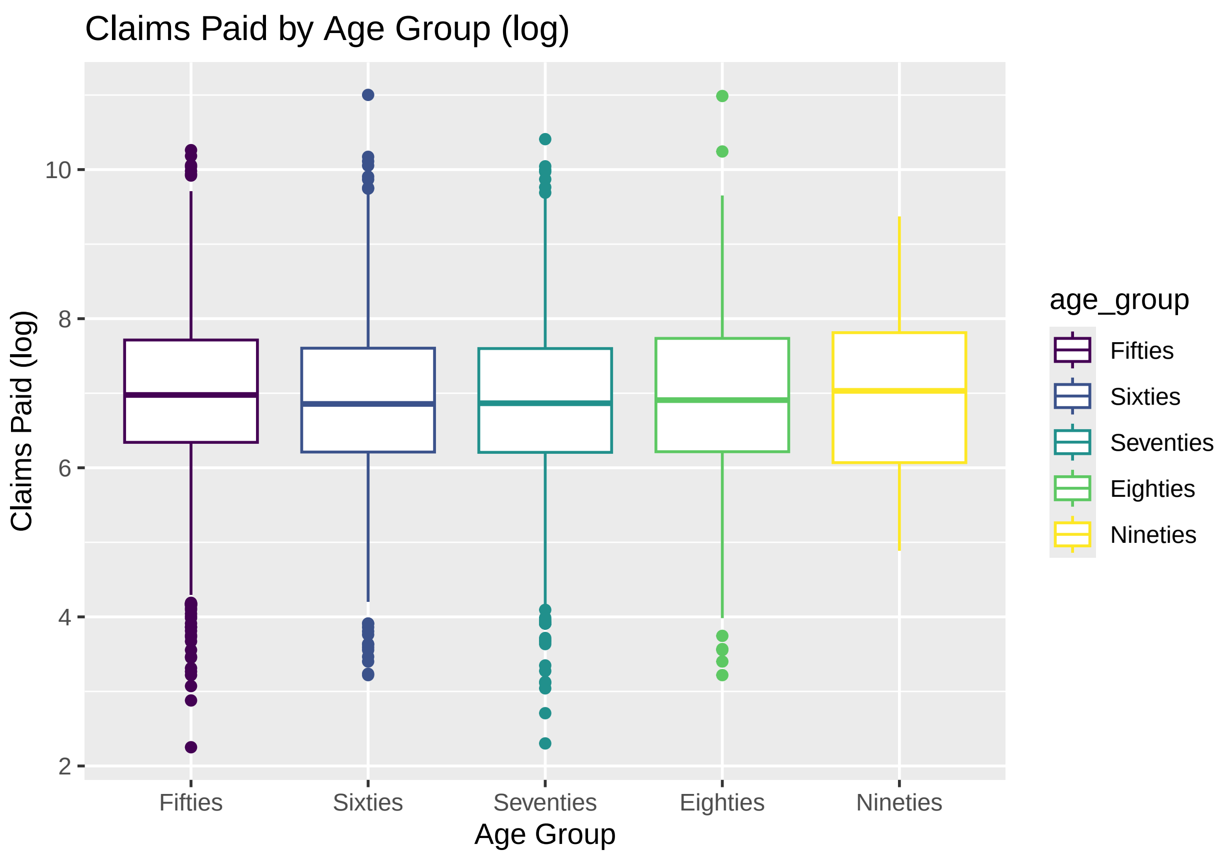
```r
# Scatter plot with regression line
ggplot(autoClaims, aes(x = AGE, y = PAID)) +
  geom_point() +
  geom_smooth(method = "lm", color = "red") +
  labs(title = "Scatter Plot with Regression Line",
       x = "Age",
       y = "Claims Paid")
```

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



Log Transformation of PAID

```
# Log transformation of PAID
autoClaims <- autoClaims %>%
  mutate(logPaid = log(PAID))

# Boxplot of logClaims by age group
ggplot(autoClaims, aes(x = age_group, y = logPaid, color = age_group)) +
  geom_boxplot() +
  labs(title = "Claims Paid by Age Group (log)", x = "Age Group", y = "Claims Paid (log)")
```



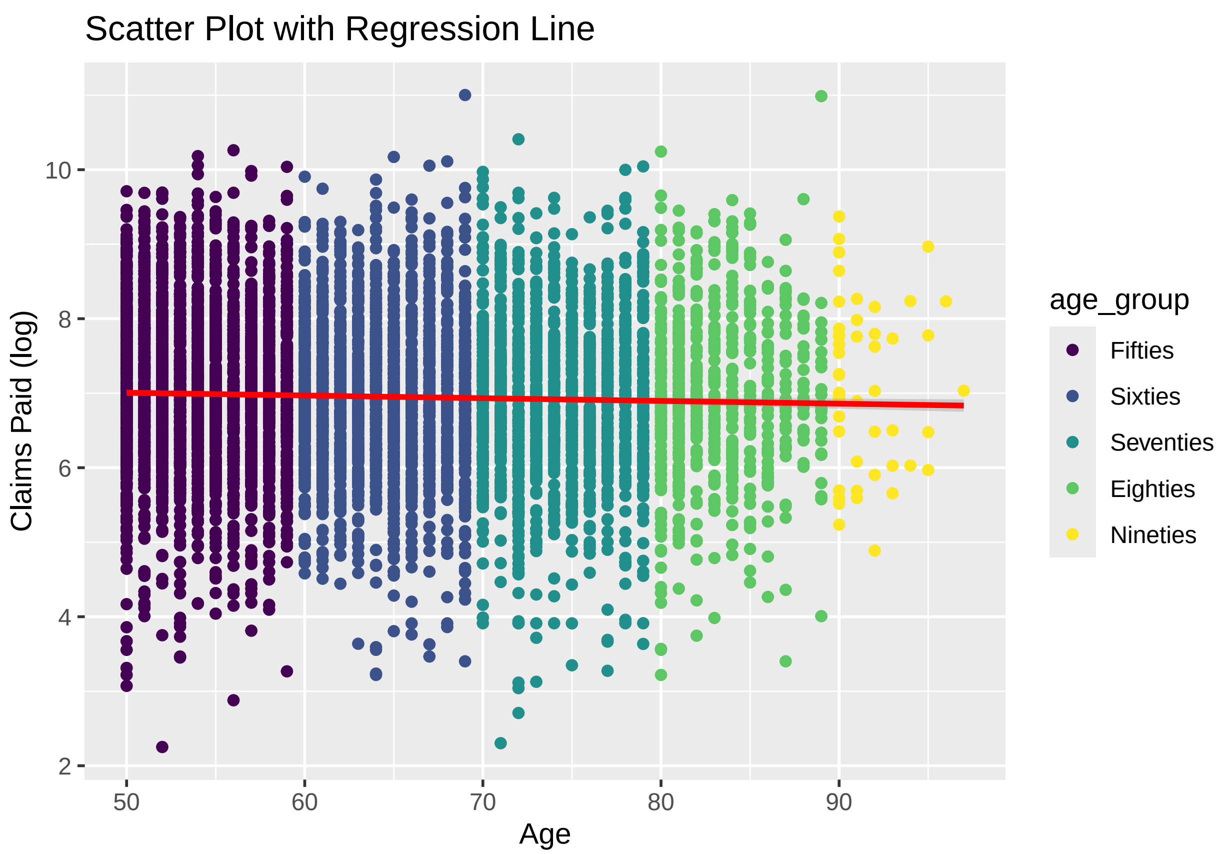

```r
# Linear regression model with log-transformed PAID
logClaimsModel <- lm(logPaid ~ AGE, data = autoClaims)
summary(logClaimsModel)
```

```
##
## Call:
## lm(formula = logPaid ~ AGE, data = autoClaims)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.7475 -0.6927 -0.0431  0.7101  4.1238
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.189060   0.078865  91.157   <2e-16 ***
## AGE         -0.003658   0.001219  -3.001   0.0027 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.07 on 6771 degrees of freedom
## Multiple R-squared:  0.001329,   Adjusted R-squared:  0.001181
## F-statistic: 9.007 on 1 and 6771 DF,  p-value: 0.002699
```

Scatter Plot with Regression Line

```r
# Scatter plot with regression line
ggplot(autoClaims, aes(x = AGE, y = logPaid, color = age_group)) +
  geom_point() +
  geom_smooth(method = "lm", color = "red") +
  labs(title = "Scatter Plot with Regression Line", x = "Age", y = "Claims Paid (log)")
```

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

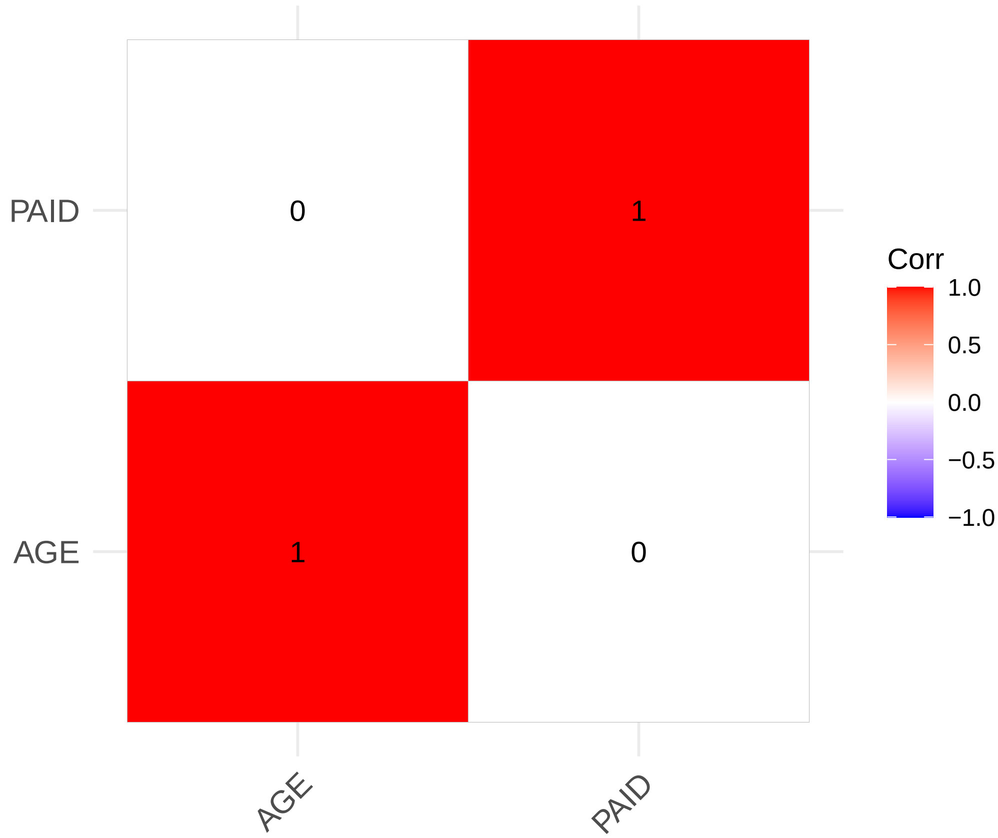


Correlation Analysis

```r
cor_matrix <- autoClaims %>%
  dplyr::select(AGE, PAID) %>%
  cor()  # Correlation matrix for numerical variables

print(cor_matrix)
```

```
##                AGE        PAID
## AGE  1.000000000 0.004188371
## PAID 0.004188371 1.000000000
```

```r
ggcorrplot(cor_matrix, lab = TRUE)  # Visualize the correlation matrix
```




```r
# Between AGE and PAID
cor(autoClaims$AGE, autoClaims$PAID)
```

```
## [1] 0.004188371
```

```r
# Correlation between AGE and log(PAID)
cor(autoClaims$AGE, autoClaims$logPaid)
```

```
## [1] -0.03644881
```

Multivariate Regression

```r
# Multivariate regression model including age group and other variables
multivariateModel <- lm(logPaid ~ AGE + GENDER + CLASS + STATE + age_group, data = autoClaims)
summary(multivariateModel)
```

```
##
## Call:
## lm(formula = logPaid ~ AGE + GENDER + CLASS + STATE + age_group,
##     data = autoClaims)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.7418 -0.6786 -0.0475  0.7079  4.2138
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)    7.161783   0.350532  20.431  < 2e-16 ***
## AGE           -0.004725   0.004616  -1.023 0.306131
## GENDERM        0.038944   0.026899   1.448 0.147727
## CLASSC11       0.044922   0.052398   0.857 0.391297
## CLASSC1A      -0.116108   0.128220  -0.906 0.365212
## CLASSC1B       0.008869   0.066523   0.133 0.893944
## CLASSC1C      -0.174947   0.178056  -0.983 0.325870
## CLASSC2       -0.193543   0.142796  -1.355 0.175341
## CLASSC6        0.048127   0.062862   0.766 0.443948
## CLASSC7       -0.019799   0.054878  -0.361 0.718275
## CLASSC71       0.023409   0.053325   0.439 0.660682
## CLASSC72       0.256317   0.123465   2.076 0.037929 *
## CLASSC7A       0.116614   0.109062   1.069 0.284999
## CLASSC7B       0.113582   0.059293   1.916 0.055458 .
## CLASSC7C       0.283816   0.126352   2.246 0.024721 *
## CLASSF1        0.123038   0.202368   0.608 0.543213
## CLASSF11      -0.059261   0.174830  -0.339 0.734648
## CLASSF6        0.044134   0.100476   0.439 0.660495
## CLASSF7       -0.300991   0.145466  -2.069 0.038570 *
## CLASSF71       0.031300   0.118874   0.263 0.792328
## STATESTATE 02  0.103574   0.089286   1.160 0.246080
## STATESTATE 03  0.005859   0.101042   0.058 0.953759
## STATESTATE 04  0.021793   0.093752   0.232 0.816192
## STATESTATE 06  0.287418   0.094106   3.054 0.002266 **
## STATESTATE 07  0.078141   0.106345   0.735 0.462496
## STATESTATE 10  0.187015   0.105439   1.774 0.076163 .
## STATESTATE 11  0.157324   0.365536   0.430 0.666923
## STATESTATE 12  0.391727   0.108343   3.616 0.000302 ***
## STATESTATE 13  0.190729   0.111963   1.703 0.088521 .
## STATESTATE 14  0.061605   0.116698   0.528 0.597586
## STATESTATE 15  0.068291   0.086622   0.788 0.430501
## STATESTATE 17  0.200930   0.096971   2.072 0.038297 *
## age_group.L    0.245736   0.170185   1.444 0.148804
## age_group.Q    0.161547   0.086019   1.878 0.060418 .
## age_group.C    0.014876   0.059179   0.251 0.801534
## age_group^4   -0.011917   0.037636  -0.317 0.751531
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.066 on 6737 degrees of freedom
## Multiple R-squared:  0.01511,    Adjusted R-squared:  0.009998
## F-statistic: 2.954 on 35 and 6737 DF,  p-value: 1.349e-08
```

Linear Model with Interaction Term

```r
# Interaction model between AGE and CLASS
interactionModel <- lm(PAID ~ AGE * CLASS, data = autoClaims)
summary(interactionModel)
```

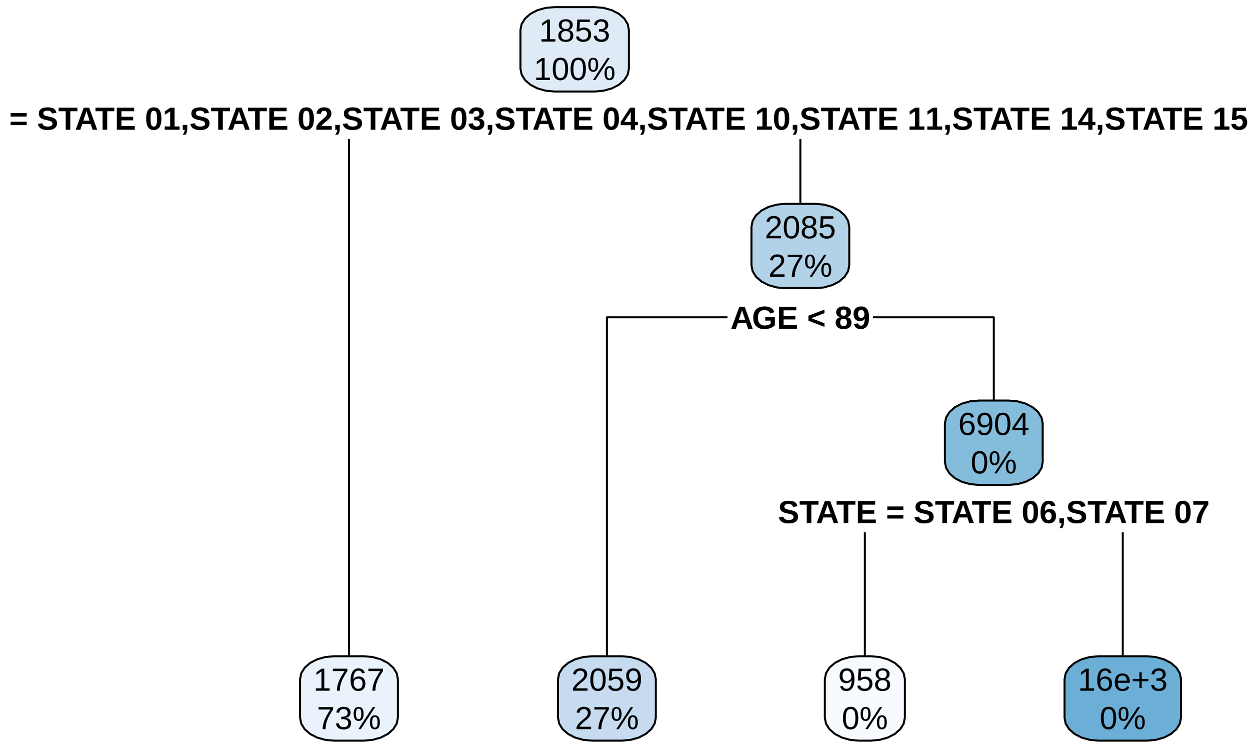
```
##
## Call:
## lm(formula = PAID ~ AGE * CLASS, data = autoClaims)
##
## Residuals:
```

```
##    Min     1Q Median     3Q    Max
##  -2796  -1305   -814    292  58007
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      835.445    665.189   1.256   0.2092
## AGE               15.696     10.411   1.508   0.1317
## CLASSC11        1519.986    812.013   1.872   0.0613 .
## CLASSC1A        1756.276   2771.983   0.634   0.5264
## CLASSC1B         622.642   1178.881   0.528   0.5974
## CLASSC1C        1006.187   4557.760   0.221   0.8253
## CLASSC2         1789.068   2427.484   0.737   0.4611
## CLASSC6        -1954.168   1492.408  -1.309   0.1904
## CLASSC7         -523.026   1049.317  -0.498   0.6182
## CLASSC71        1873.400    997.101   1.879   0.0603 .
## CLASSC72        4321.609   3049.675   1.417   0.1565
## CLASSC7A        1641.655   2738.038   0.600   0.5488
## CLASSC7B         131.751   1281.394   0.103   0.9181
## CLASSC7C        2890.117   3776.381   0.765   0.4441
## CLASSF1          970.050   3144.018   0.309   0.7577
## CLASSF11       -2413.155   3159.490  -0.764   0.4450
## CLASSF6          -80.229   3072.899  -0.026   0.9792
## CLASSF7         1281.008   3301.271   0.388   0.6980
## CLASSF71        -301.274   2782.540  -0.108   0.9138
## AGE:CLASSC11     -23.231     12.373  -1.877   0.0605 .
## AGE:CLASSC1A     -30.434     47.508  -0.641   0.5218
## AGE:CLASSC1B      -7.611     19.863  -0.383   0.7016
## AGE:CLASSC1C     -22.205     81.313  -0.273   0.7848
## AGE:CLASSC2      -37.220     40.553  -0.918   0.3587
## AGE:CLASSC6       22.783     20.060   1.136   0.2561
## AGE:CLASSC7        8.664     17.032   0.509   0.6110
## AGE:CLASSC71     -31.134     16.321  -1.908   0.0565 .
## AGE:CLASSC72     -66.594     51.813  -1.285   0.1987
## AGE:CLASSC7A     -25.399     46.429  -0.547   0.5844
## AGE:CLASSC7B       4.793     21.958   0.218   0.8272
## AGE:CLASSC7C     -43.248     68.161  -0.634   0.5258
## AGE:CLASSF1      -18.293     46.139  -0.396   0.6918
## AGE:CLASSF11      34.403     44.385   0.775   0.4383
## AGE:CLASSF6       -1.810     39.142  -0.046   0.9631
## AGE:CLASSF7      -33.312     53.454  -0.623   0.5332
## AGE:CLASSF71       2.145     46.149   0.046   0.9629
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2646 on 6737 degrees of freedom
## Multiple R-squared:  0.00585,    Adjusted R-squared:  0.0006853
## F-statistic: 1.133 on 35 and 6737 DF,  p-value: 0.2712
```

Decision Tree Model

```r
# Decision tree model with adjusted parameters to find the best split
treeModel <- rpart(PAID ~ AGE + GENDER + CLASS + STATE + age_group,
                   data = autoClaims,
                   control = rpart.control(minsplit = 10, cp = 0.006))
```

```r
rpart.plot(treeModel)
```



Model Evaluation

```r
# Train-Test Split
# Splitting the data into training and test sets for model evaluation
set.seed(121)
trainIndex <- createDataPartition(autoClaims_dummies$PAID, p = 0.8, list = FALSE)
trainData <- autoClaims_dummies[trainIndex, ]
testData <- autoClaims_dummies[-trainIndex, ]

# Train a multivariate model on the training data
trainModel <- lm(PAID ~ ., data = trainData)
summary(trainModel)
```

```
##
## Call:
## lm(formula = PAID ~ ., data = trainData)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -2958  -1300   -798    279  57936
##
## Coefficients: (1 not defined because of singularities)
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)           6545.5665  2733.8389   2.394  0.01669 *
```

```
## AGE                  -182.9401    85.9013  -2.130  0.03325 *
## CLASS_C11            2254.3195   715.4683   3.151  0.00164 **
## CLASS_C1A             100.6986   369.6101   0.272  0.78529
## CLASS_C1B              93.6524   189.4511   0.494  0.62109
## CLASS_C1C            -242.4939   495.4142  -0.489  0.62452
## CLASS_C2             -400.2350   413.8302  -0.967  0.33351
## CLASS_C6             -377.9899   200.0675  -1.889  0.05890 .
## CLASS_C7               -5.5645   156.7908  -0.035  0.97169
## CLASS_C71              43.1341   152.7836   0.282  0.77771
## CLASS_C72             589.5818   368.7899   1.599  0.10995
## CLASS_C7A              73.9315   300.6303   0.246  0.80575
## CLASS_C7B             433.8250   170.4337   2.545  0.01094 *
## CLASS_C7C             394.9194   360.0347   1.097  0.27274
## CLASS_F1             -253.7554   556.9157  -0.456  0.64866
## CLASS_F11             229.7799   514.5627   0.447  0.65522
## CLASS_F6             -268.5065   301.9414  -0.889  0.37390
## CLASS_F7             -606.8065   407.6120  -1.489  0.13663
## CLASS_F71             -15.2205   328.7591  -0.046  0.96308
## `STATE_STATE 02`      497.9716   677.4085   0.735  0.46230
## `STATE_STATE 03`      195.3468   290.6579   0.672  0.50156
## `STATE_STATE 04`      242.1686   271.0289   0.894  0.37162
## `STATE_STATE 06`      559.4890   270.8782   2.065  0.03893 *
## `STATE_STATE 07`      452.5982   309.1792   1.464  0.14329
## `STATE_STATE 10`      301.5184   303.7725   0.993  0.32096
## `STATE_STATE 11`      -31.3648  1135.9024  -0.028  0.97797
## `STATE_STATE 12`      764.7059   314.5299   2.431  0.01508 *
## `STATE_STATE 13`      405.6317   322.4841   1.258  0.20851
## `STATE_STATE 14`      102.8975   331.5127   0.310  0.75628
## `STATE_STATE 15`      140.8134   250.6503   0.562  0.57428
## `STATE_STATE 17`      502.0219   281.2945   1.785  0.07437 .
## GENDER_M             -188.7473   462.2228  -0.408  0.68304
## age_group_Sixties    -114.9976   174.7806  -0.658  0.51060
## age_group_Seventies -214.2427   288.9306  -0.742  0.45842
## age_group_Eighties  -464.6484   451.0675  -1.030  0.30301
## age_group_Nineties  -858.7485   796.0222  -1.079  0.28073
## AGE_CLASS             -34.4038    10.7103  -3.212  0.00132 **
## AGE_GENDER              2.5711     7.1456   0.360  0.71900
## AGE_STATE              -3.7357     9.8491  -0.379  0.70448
## AGE_AGE1                    NA         NA      NA       NA
## AGE_AGE2                1.6403     0.6728   2.438  0.01479 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2711 on 5381 degrees of freedom
## Multiple R-squared:  0.01139,    Adjusted R-squared:  0.004221
## F-statistic: 1.589 on 39 and 5381 DF,  p-value: 0.01141
```

```r
# Predict on the test data
predictions <- predict(trainModel, newdata = testData)
summary(predictions)
```

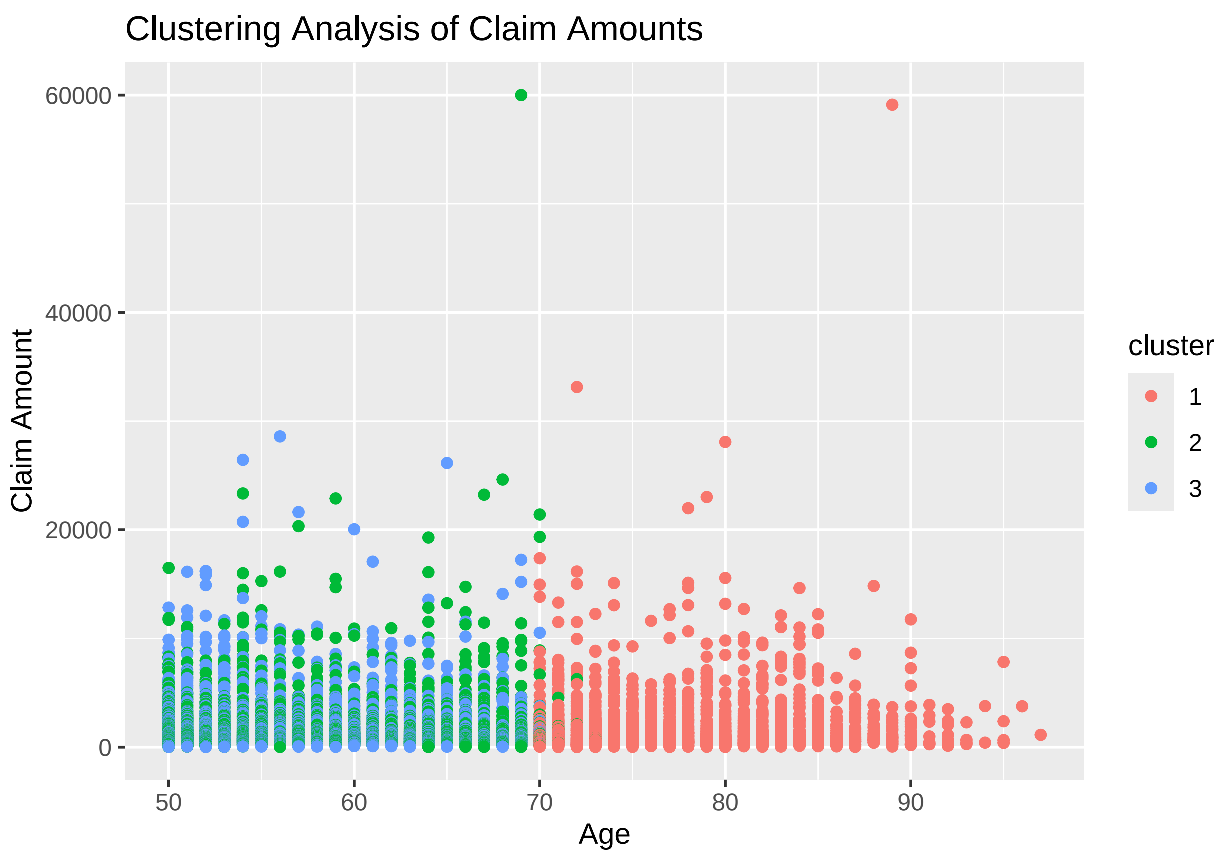
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   840.7  1666.8  1831.1  1877.7  2068.5  3053.0
```

```
# Calculate RMSE on the test set
RMSE <- sqrt(mean((testData$PAID - predictions)^2))
print(paste("Root Mean Squared Error (Test Set):", RMSE))
```

```
## [1] "Root Mean Squared Error (Test Set): 2348.64933714104"
```

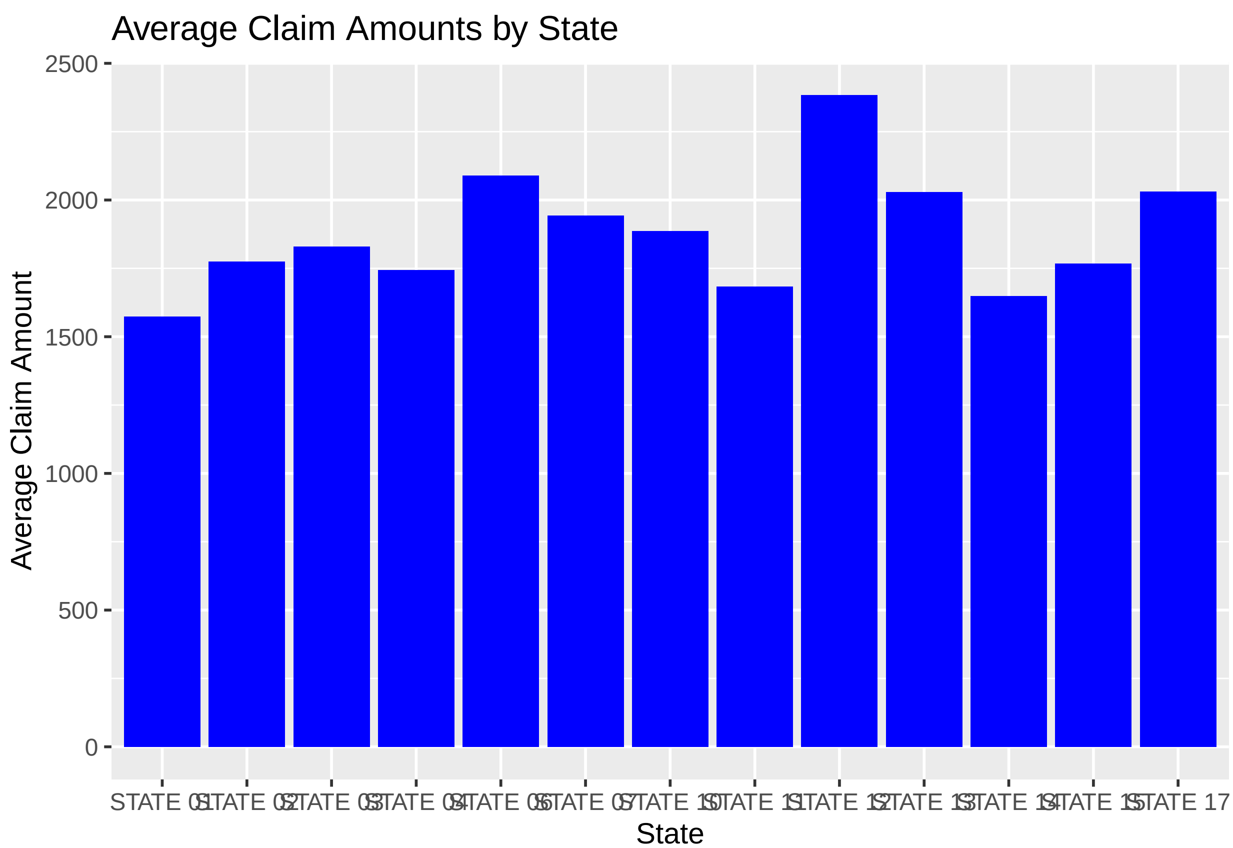
Clustering Analysis

```
# Perform K-Means Clustering
# K-Means clustering to identify groups of customers with similar claim amounts and demographics
set.seed(123)

# Perform K-means clustering using the cleaned dataset without the PAID column
numeric_columns <- autoClaims_dummies %>%
  select_if(is.numeric) %>%
  dplyr::select(-PAID)

# Apply K-means clustering on the numeric columns
kmeans_result <- kmeans(scale(numeric_columns), centers = 3)

# Add the cluster assignments to the cleaned data set
autoClaims_cleaned <- autoClaims_dummies
autoClaims_cleaned$cluster <- as.factor(kmeans_result$cluster)

# Visualize the clustering results to see how different age groups fall into clusters
ggplot(autoClaims_cleaned, aes(x = AGE, y = PAID, color = cluster)) +
  geom_point() +
  labs(title = "Clustering Analysis of Claim Amounts", x = "Age", y = "Claim Amount")
```

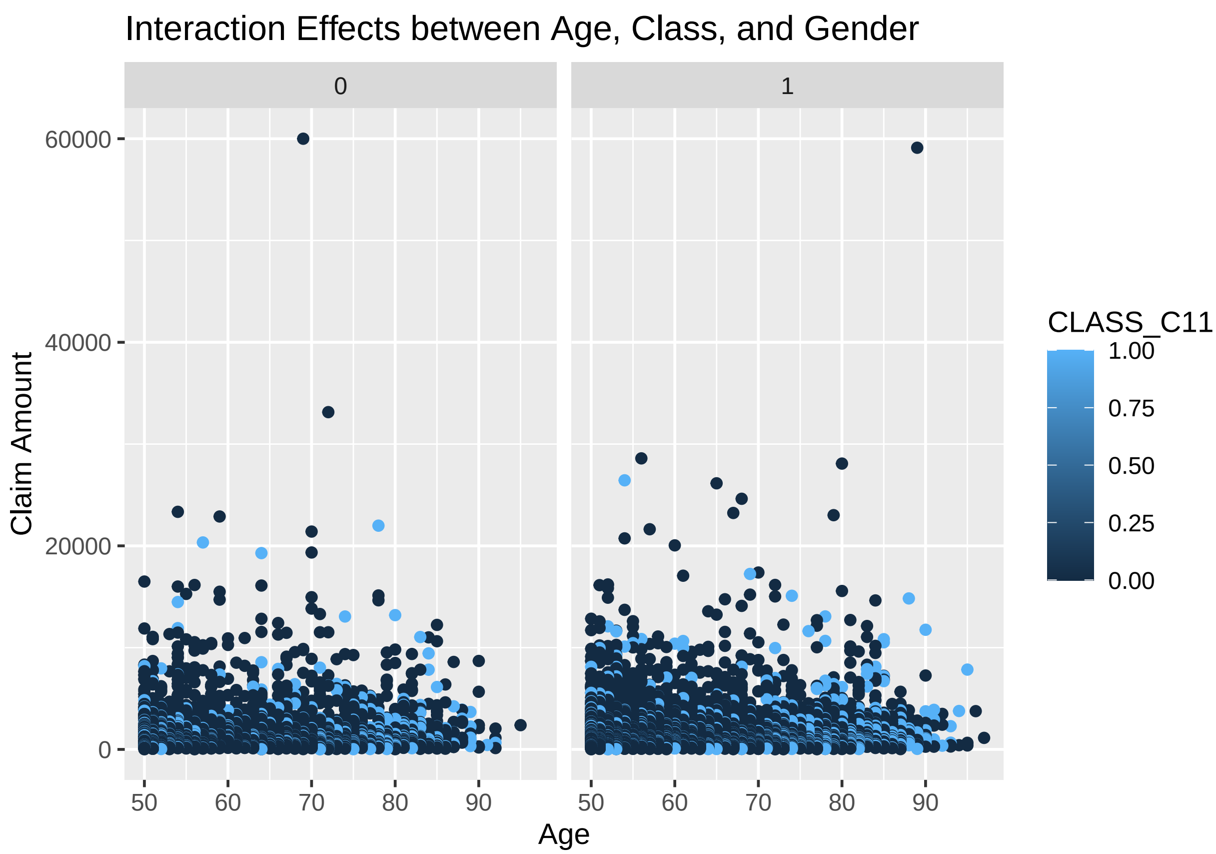


Visualization

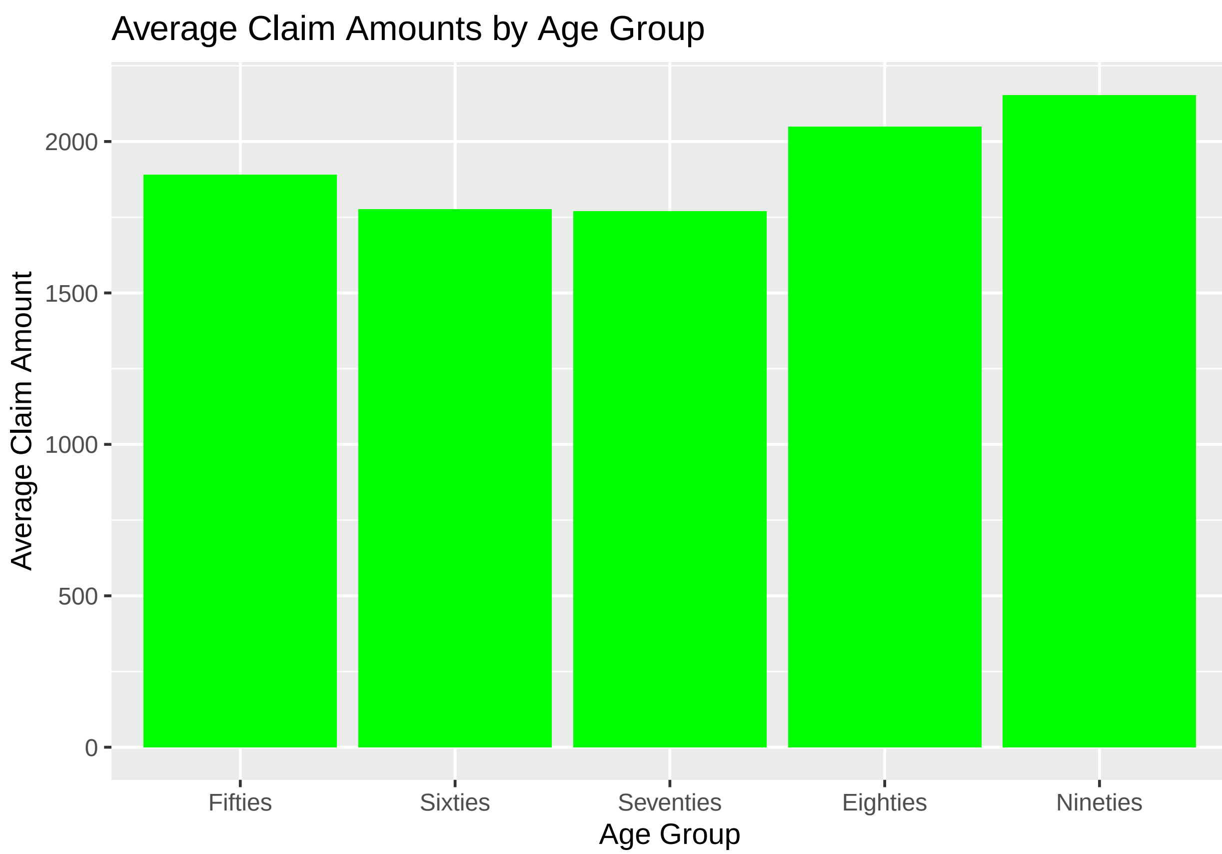
```
# Bar Charts
# Bar chart for average PAID by STATE
state_avg <- autoClaims %>%
  group_by(STATE) %>%
  summarize(mean_paid = mean(PAID))

ggplot(state_avg, aes(x = STATE, y = mean_paid)) +
  geom_bar(stat = "identity", fill = "blue") +
  labs(title = "Average Claim Amounts by State", x = "State", y = "Average Claim Amount")
```




```r
# Interaction Plots
# Interaction plot between AGE, CLASS, and GENDER
ggplot(autoClaims_cleaned, aes(x = AGE, y = PAID, color = CLASS_C11)) +
  geom_point() +
  facet_wrap(~GENDER_M) +
  labs(title = "Interaction Effects between Age, Class, and Gender", x = "Age", y = "Claim Amount")
```




```r
# Age Group Comparison
# Bar chart for average PAID by age group
age_group_avg <- autoClaims %>%
  group_by(age_group) %>%
  summarize(mean_paid = mean(PAID))

ggplot(age_group_avg, aes(x = age_group, y = mean_paid)) +
  geom_bar(stat = "identity", fill = "green") +
  labs(title = "Average Claim Amounts by Age Group", x = "Age Group", y = "Average Claim Amount")
```



Key Findings:

- Age Group Analysis: The analysis revealed that as age increases, so does the average claim amount, peaking with the Nineties age group.Specifically, average claim amounts by age group show an increase from the Fifties to the Nineties: Fifties ($1890.8), Sixties ($1776.3), Seventies ($1769.4), Eighties ($2049.0), and Nineties ($2153.9).
- State Analysis: There was variability in average claim amounts by state, with some states (like State 12) showing notably higher averages compared to others. This suggests regional differences in claim amounts.
- Impact of Demographics and Policy Type: Multivariate regression highlighted specific CLASS and STATE variables as significant. Notably, CLASS_C7B and STATE_STATE 12 emerged as significant predictors with positive coefficients of $344 and $611 respectively, indicating higher claim amounts associated with these categories.
- Correlation Analysis: The correlation between age and paid claims was very low ($0.004188371), suggesting that while age group categories show a trend in claims, age as a continuous variable alone is not a strong predictor.
- Decision Tree Insights: The decision tree analysis, highlighted `STATE` and `AGE` as critical nodes. For instance, the split at `AGE < 89` and specific states like `STATE_06` and `STATE_07` suggest that geographical and age factors are crucial in determining the claim amounts.
- Clustering Analysis: The k-means clustering identified groups of claims with similar characteristics. Three distinct clusters were observed, with the first cluster showing the highest claim amounts, particularly among older age groups. This might indicate specific risk profiles or policy characteristics within these clusters.

- Log-Transformed Regression: The log-transformed regression analysis further established the significance of age with a slight negative correlation with log-transformed claim amounts (`cor(autoClaims$AGE, autoClaims$logPaid) = -0.03644881`), which indicates that higher ages slightly decrease the claim amounts when transformed logarithmically, contrasting the findings in untransformed data.

Conclusion:

The analysis effectively identifies several key variables influencing claim amounts in auto insurance data. Age groups, certain states, and specific insurance classes significantly impact claim sizes, with older age groups generally incurring higher claims. Regional variations also affect claims, as seen with the variance across different states.