Third party Auto Insurance Claims Analysis

Data Science with R Programming - Course end project1

Student Name – Manohari Wijesooriya

10/05/2023

s

Contents

1. Problem Statement

The data gives the details of third party motor insurance claims in Sweden for the year 1977. In Sweden, all motor insurance companies apply identical risk arguments to classify customers, and thus their portfolios and their claims statistics can be combined. The data were compiled by a Swedish Committee on the Analysis of Risk Premium in Motor Insurance. The Committee was asked to look into the problem of analyzing the real influence on the claims of the risk arguments and to compare this structure with the actual tariff.

2. Data

The insurance dataset holds 7 variables and the description of these variables are given below:

Variable	Description
Kilometres	Distance driven by a vehicle, grouped into five categories. Kilometers travelled per year 1: < 1000 2: 1000-15000 3: 15000-20000 4: 20000-25000 5: > 25000
Zone	Geographical zone of a vehicle, grouped into 7 categories. 1: Stockholm, Göteborg, and Malmö with surroundings 2: Other large cities with surroundings 3: Smaller cities with surroundings in southern Sweden 4: Rural areas in southern Sweden 5: Smaller cities with surroundings in northern Sweden 6: Rural areas in northern Sweden 7: Gotland
Bonus	Driver claim experience, grouped into 7 categories. No claims bonus; equal to the number of years, plus one, since the last claim
Make	1-8 represents eight different common car models. All other models are combined in class 9.
Insured	Number of insured in policy-years
Claims	Number of claims
Payment	Total value of payments in Skr (Swedish Krona)

Data dictionary

3. Descriptive Analysis

This report is to gain basic insights into the data set and to prepare for the further analysis.

```
#import libraries
# general data visualization

library('ggplot2')
library('scales')
library('grid')
library('ggthemes')
library('gridExtra')
library('RColorBrewer')
library('corrplot')
```

```
## corrplot 0.84 loaded
```

```
library('dplyr')
```

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

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

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

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

```
library('ellipse')
```

```
##
## Attaching package: 'ellipse'
```

```
## The following object is masked from 'package:graphics':
##
##     pairs
```

```
library(ggpubr)
```

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

```
library(caret)
```

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

```
library(cluster)
library(Hmisc)
```

```
## Loading required package: survival
```

```
##
## Attaching package: 'survival'
```

```
## The following object is masked from 'package:caret':
##
##     cluster
```

```
## Loading required package: Formula
```

```
##
## Attaching package: 'Hmisc'
```

```
## The following objects are masked from 'package:dplyr':
##
##     src, summarize
```

```
## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
#get working directory
getwd()
```

```
## [1] "/home/labsuser"
```

```
# read csv file
autoclaims_data<- read.csv("SwedishMotorInsurance.csv")
head(autoclaims_data)
```

```
##   Kilometres Zone Bonus Make Insured Claims Payment
## 1          1    1     1    1  455.13    108  392491
## 2          1    1     1    2   69.17     19   46221
## 3          1    1     1    3   72.88     13   15694
## 4          1    1     1    4 1292.39    124  422201
## 5          1    1     1    5  191.01     40  119373
## 6          1    1     1    6  477.66     57  170913
```

```
#checking the structure of dataset
str(autoclaims_data)
```

```
## 'data.frame':    2182 obs. of  7 variables:
##  $ Kilometres: int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Zone      : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Bonus     : int  1 1 1 1 1 1 1 1 1 2 ...
##  $ Make      : int  1 2 3 4 5 6 7 8 9 1 ...
##  $ Insured   : num  455.1 69.2 72.9 1292.4 191 ...
##  $ Claims    : int  108 19 13 124 40 57 23 14 1704 45 ...
##  $ Payment   : int  392491 46221 15694 422201 119373 170913 56940 77487 6805992 214011 ...
```

```
# checking null values
colSums(is.na(autoclaims_data))
```

```
## Kilometres       Zone      Bonus       Make    Insured     Claims    Payment
##          0          0          0          0          0          0          0
```

```
#checking summary
summary(autoclaims_data)
```

```
##    Kilometres         Zone          Bonus            Make
##  Min.   :1.000   Min.   :1.00   Min.   :1.000   Min.   :1.000
##  1st Qu.:2.000   1st Qu.:2.00   1st Qu.:2.000   1st Qu.:3.000
##  Median :3.000   Median :4.00   Median :4.000   Median :5.000
##  Mean   :2.986   Mean   :3.97   Mean   :4.015   Mean   :4.992
##  3rd Qu.:4.000   3rd Qu.:6.00   3rd Qu.:6.000   3rd Qu.:7.000
##  Max.   :5.000   Max.   :7.00   Max.   :7.000   Max.   :9.000
##     Insured              Claims           Payment
##  Min.   :     0.01   Min.   :   0.00   Min.   :       0
##  1st Qu.:    21.61   1st Qu.:   1.00   1st Qu.:    2989
##  Median :    81.53   Median :   5.00   Median :   27404
##  Mean   :  1092.20   Mean   :  51.87   Mean   :  257008
##  3rd Qu.:   389.78   3rd Qu.:  21.00   3rd Qu.:  111954
##  Max.   :127687.27   Max.   :3338.00   Max.   :18245026
```

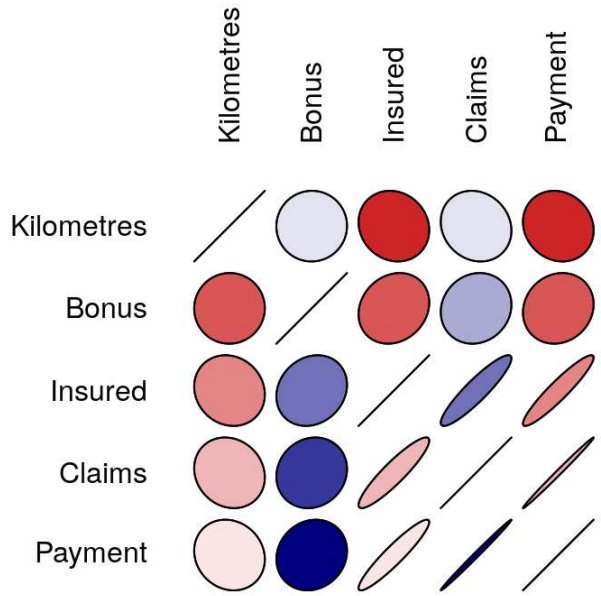
Result:

The results provide the minimum and maximum values. It also provides the mean and median values of all variables. From this you can understand the spread of data. We can see that claims and payment also have null or zero values, however the insured column does not have a zero value. This specifies that there are few entries where the car has been insured for a given period of time. However, no claim or payment has been made for that combination of car make, zone, and kilometres.

```
#keeping non categorical columns
autoclaims_data_n <- autoclaims_data[,c(-2,-4)]
#Compute Pearson's correlation
tb_corr <- cor(autoclaims_data_n, method = 'pearson')
tb_corr
```

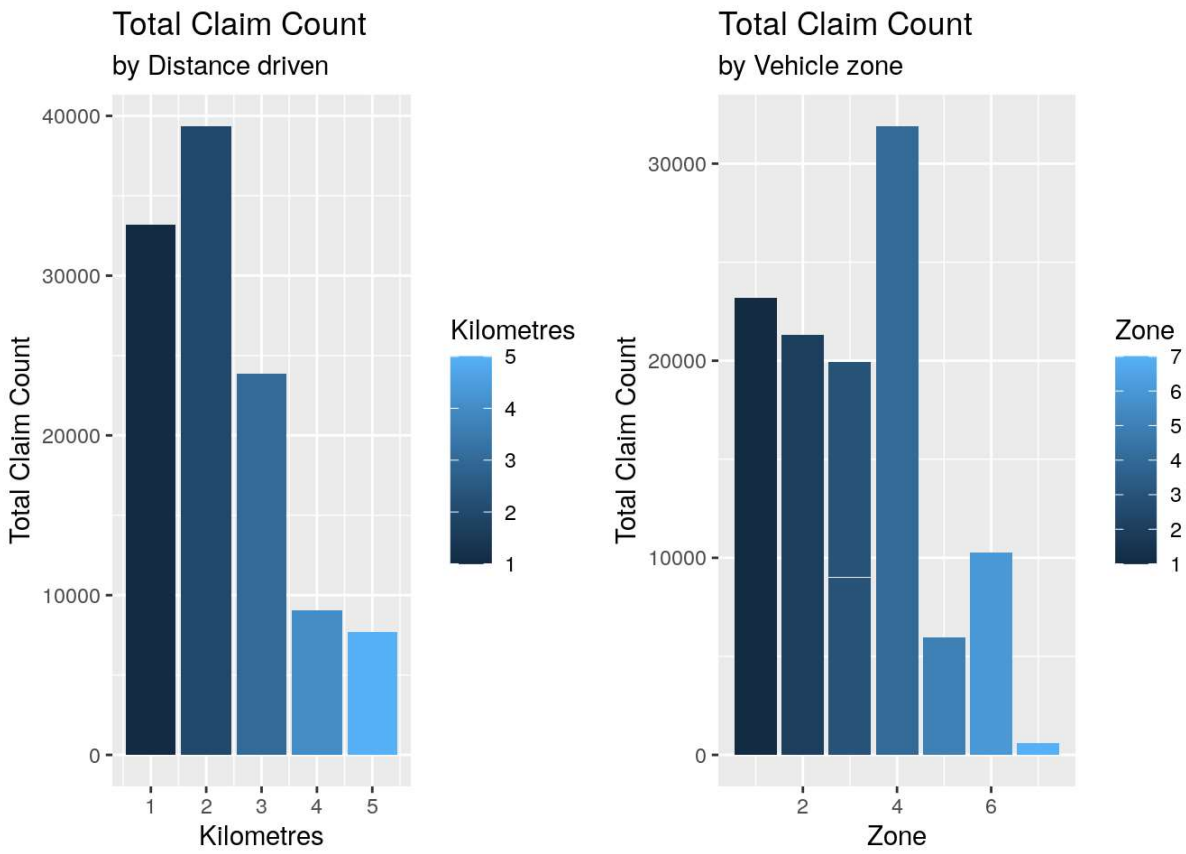
```
##              Kilometres       Bonus    Insured     Claims    Payment
## Kilometres  1.000000000 0.007226253 -0.1129903 -0.1284519 -0.1208864
## Bonus       0.007226253 1.000000000  0.1654253  0.1051024  0.1180327
## Insured    -0.112990321 0.165425256  1.0000000  0.9103478  0.9332170
## Claims     -0.128451910 0.105102362  0.9103478  1.0000000  0.9954003
## Payment    -0.120886355 0.118032691  0.9332170  0.9954003  1.0000000
```

```
require(ellipse)
plotcorr(tb_corr, col = colorRampPalette(c("firebrick3", "white", "navy"))(10))
```

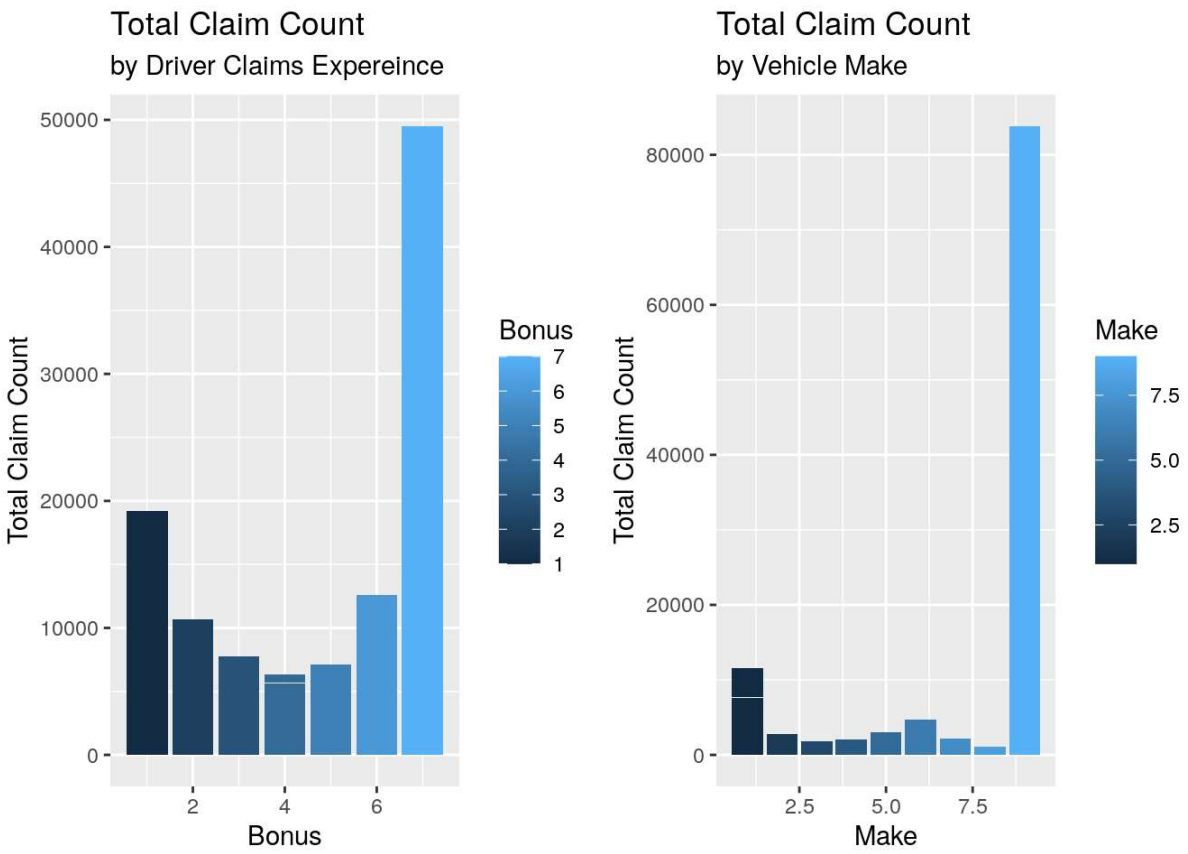


In the above plot, the length of the minor axis is computed as 1−correlation. A correlation of 1 gives a line, with the minor axis set to 0. Correlation of zero results in a circle. The intensity of the color used on the plot indicates the magnitude of the correlation. Additionally, orientation of the ellipse is used to highlight the positive or negative correlation value. For positive correlation the ellipse tilts right, with the opposite being true for negative correlation.

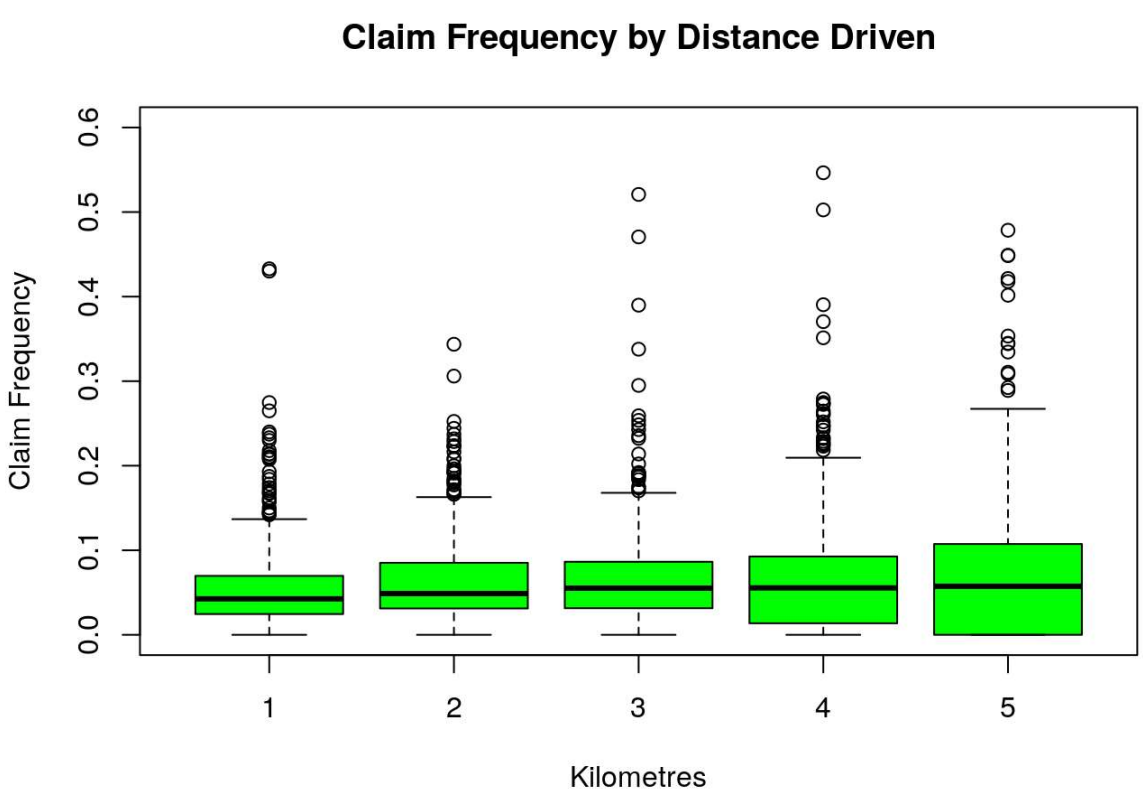
```r
#data visualisation



## Categorical features

p1 <- autoclaims_data %>%
  ggplot(aes(x = Kilometres, y = Claims, fill = Kilometres)) +
    geom_col() +
    labs(title = "Total Claim Count",
         subtitle = "by Distance driven",
              x = "Kilometres", y = "Total Claim Count")

p3 <- autoclaims_data %>%
  ggplot(aes(x = Zone, y = Claims, fill = Zone)) +
    geom_col() +
    labs(title = "Total Claim Count",
          subtitle = "by Vehicle zone",
              x = "Zone", y = "Total Claim Count")

p5 <- autoclaims_data %>%
  ggplot(aes(x = Bonus, y = Claims, fill = Bonus)) +
    geom_col() +
    labs(title = "Total Claim Count",
         subtitle = "by Driver Claims Expereince",
              x = "Bonus", y = "Total Claim Count")

p7 <- autoclaims_data %>%
  ggplot(aes(x = Make, y = Claims, fill = Make)) +
    geom_col() +
    labs(title = "Total Claim Count",
         subtitle = "by Vehicle Make",
              x = "Make", y = "Total Claim Count")


grid.arrange(p1, p3, nrow = 1)
```



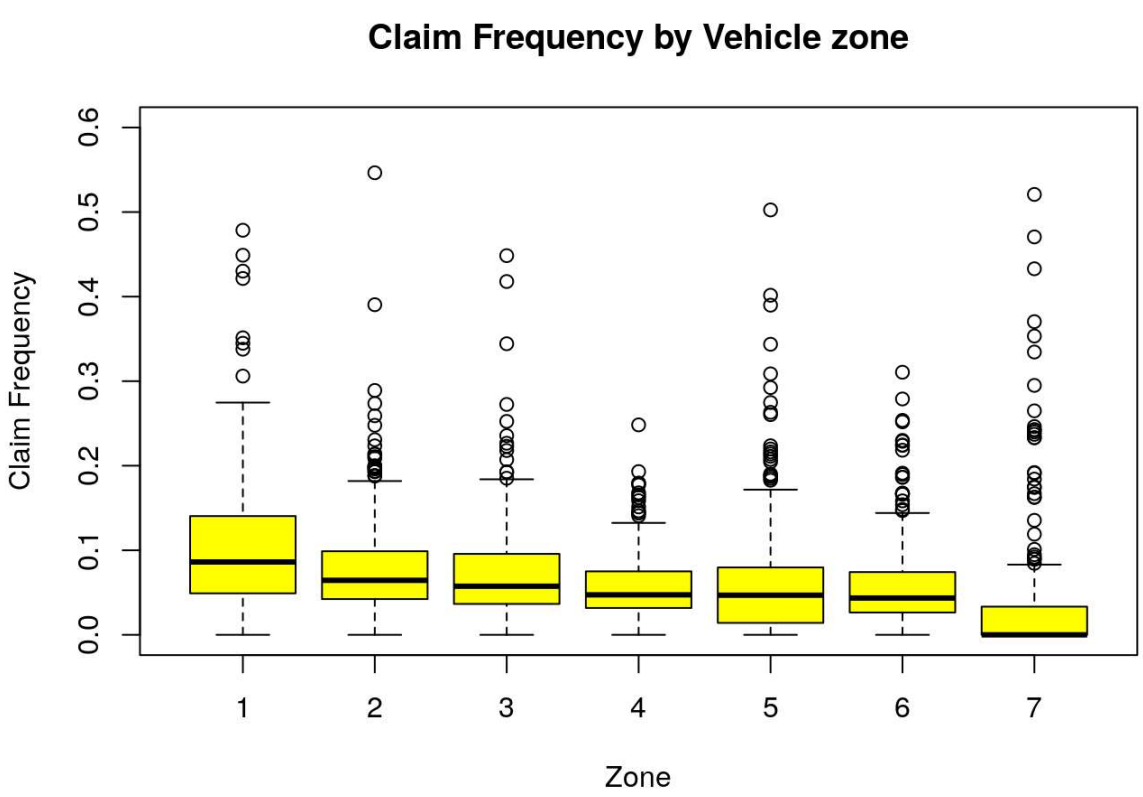

```
grid.arrange(p5, p7, nrow = 1)
```



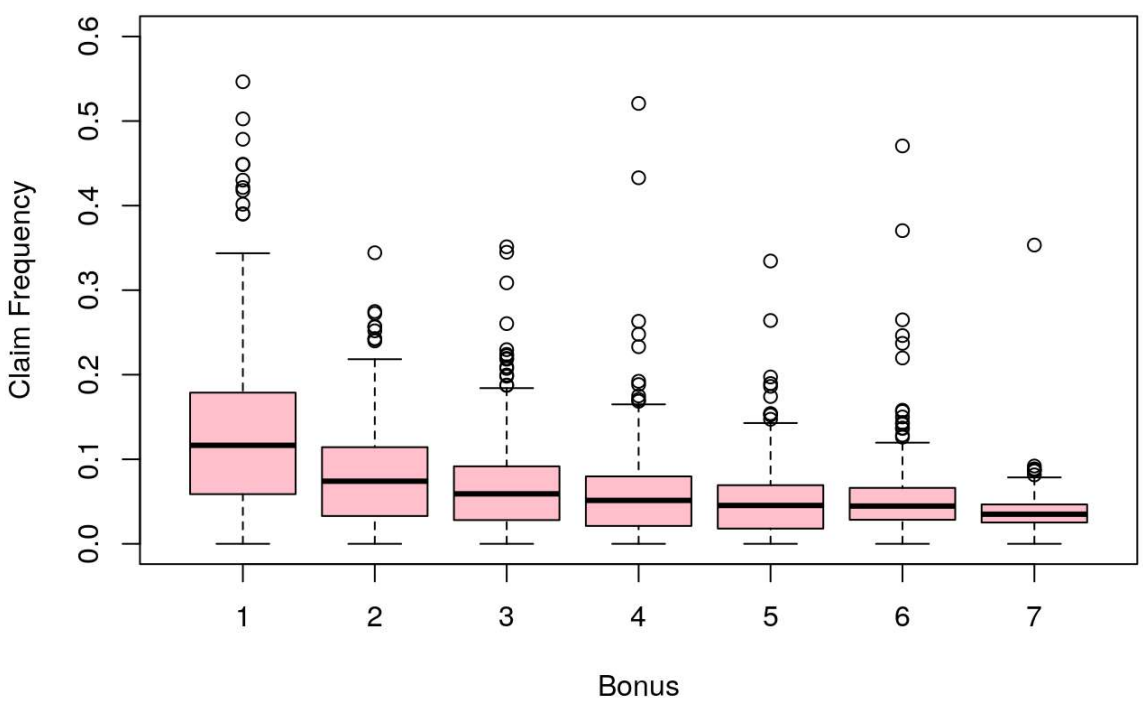

```
#visualise with boxplots

#ggplot(data = autoclaims_data, aes(Zone,Insured)) + geom_boxplot(fill = c(2:7)) +
#claim frequency
p9 <- boxplot(Claims/Insured ~ Kilometres, data=autoclaims_data, xlab="Kilometres",
        ylab="Claim Frequency", ylim = c(0, 0.6), col="green", main="Claim Frequency by Distance Driven")
```



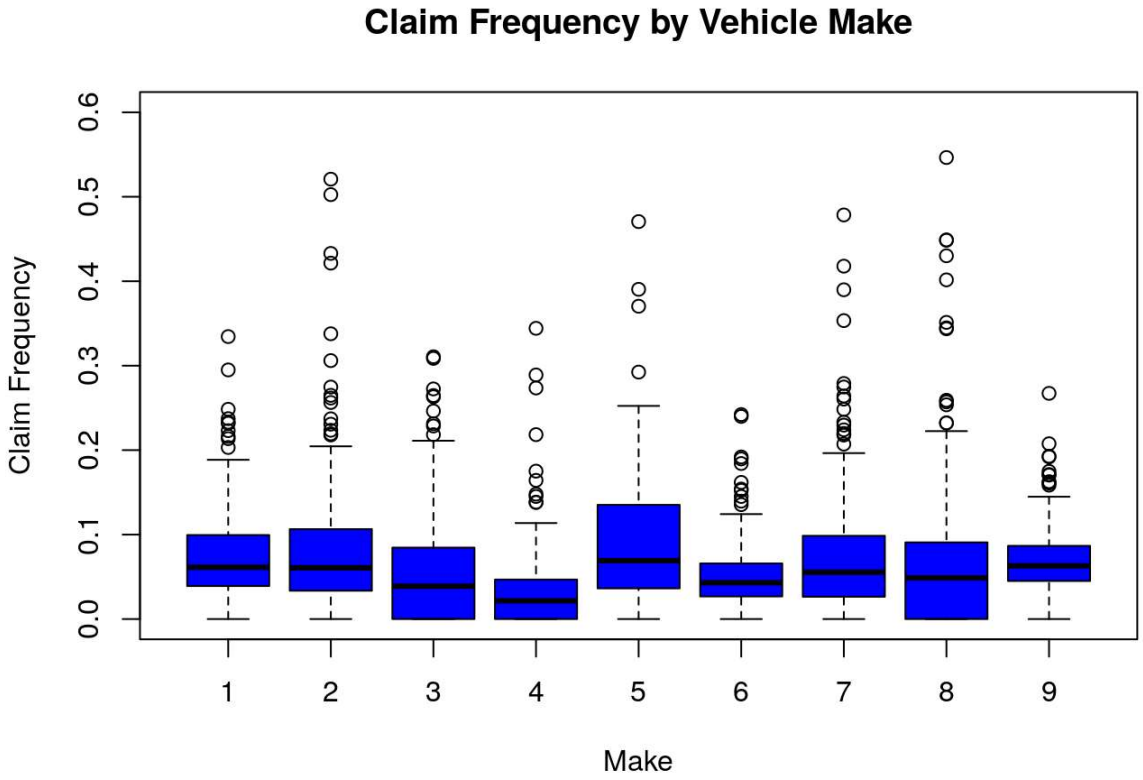

```
p10 <- boxplot(Claims/Insured ~ Zone, data=autoclaims_data, xlab="Zone",
        ylab="Claim Frequency", ylim = c(0, 0.6), col="yellow", main="Claim Frequency by Vehicle zone")
```




```
p11 <- boxplot(Claims/Insured ~ Bonus, data=autoclaims_data, xlab="Bonus",
        ylab="Claim Frequency", ylim = c(0, 0.6), col="pink", main="Claim Frequency by Driver Claims Expereince")
```




```
p12 <- boxplot(Claims/Insured ~ Make, data=autoclaims_data, xlab="Make",
      ylab="Claim Frequency", ylim = c(0, 0.6), col="blue", main="Claim Frequency by Vehicle Make")
```



4. Total Payments

The total value of payment by an insurance company is an important factor to be monitored. So the committee has decided to find whether this payment is related to number of claims and the number of insured policy years. They also want to visualize the results for better understanding.

```
require(car)
```

```
## Loading required package: car
```

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

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

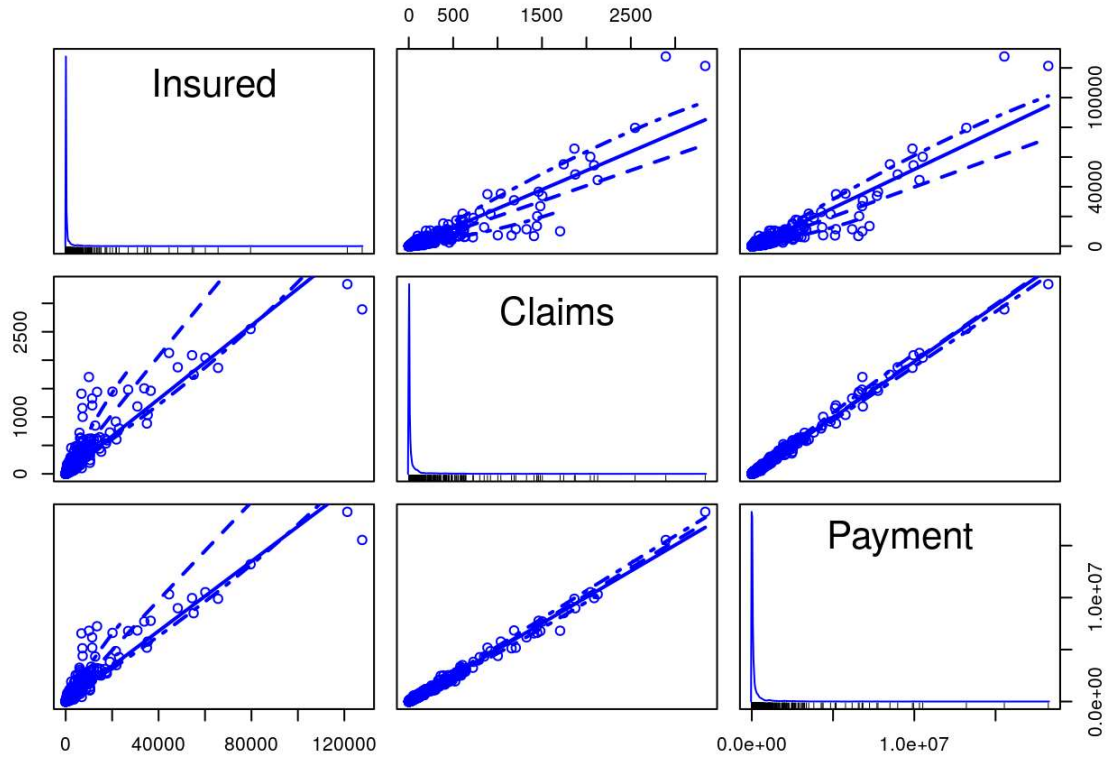
```
## The following object is masked from 'package:ellipse':
##
##     ellipse
```

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

```
require(repr)
```

```
## Loading required package: repr
```

```
# visualising impact of number of claims and the number of insured policy years on payment
options(repr.plot.width=9, repr.plot.height=9)
scatterplotMatrix(~ Insured + Claims + Payment    ,data=autoclaims_data)
```



$$
\begin{aligned}
Risk &= Claim\ frequency \times Claim\ severity \\
&= \left(\frac{Number\ of\ claims}{Insurance\ years}\right) \times \left(\frac{Claim\ cost}{Number\ of\ claims}\right) \\
&= \frac{Claimcost}{Insurance\ years},
\end{aligned}
$$

freq

```
# the values in this data set are aggregated amounts. Lets change number of claims(Clams) and the number of insured policy y
ears(Insured) to factors and evaluate impact on payment
autoclaims_data$Insured_factor <- cut(autoclaims_data$Insured, breaks = c(0, 20, 50, 100, 500, 1500,5000, 30000, 50000, 1000
00, 200000),
    labels = c("0, 20", "20, 50", "50, 100", "100, 500", "500, 1500", "1500,5000","5000, 30000","30000, 50000", "50000, 1000
00", "100000, 200000"  ))

table(autoclaims_data$Insured_factor)
```

```
##
##          0, 20         20, 50        50, 100       100, 500      500, 1500
##            523            358            283            540            237
##      1500,5000    5000, 30000   30000, 50000  50000, 100000 100000, 200000
##            140             87              7              5              2
```

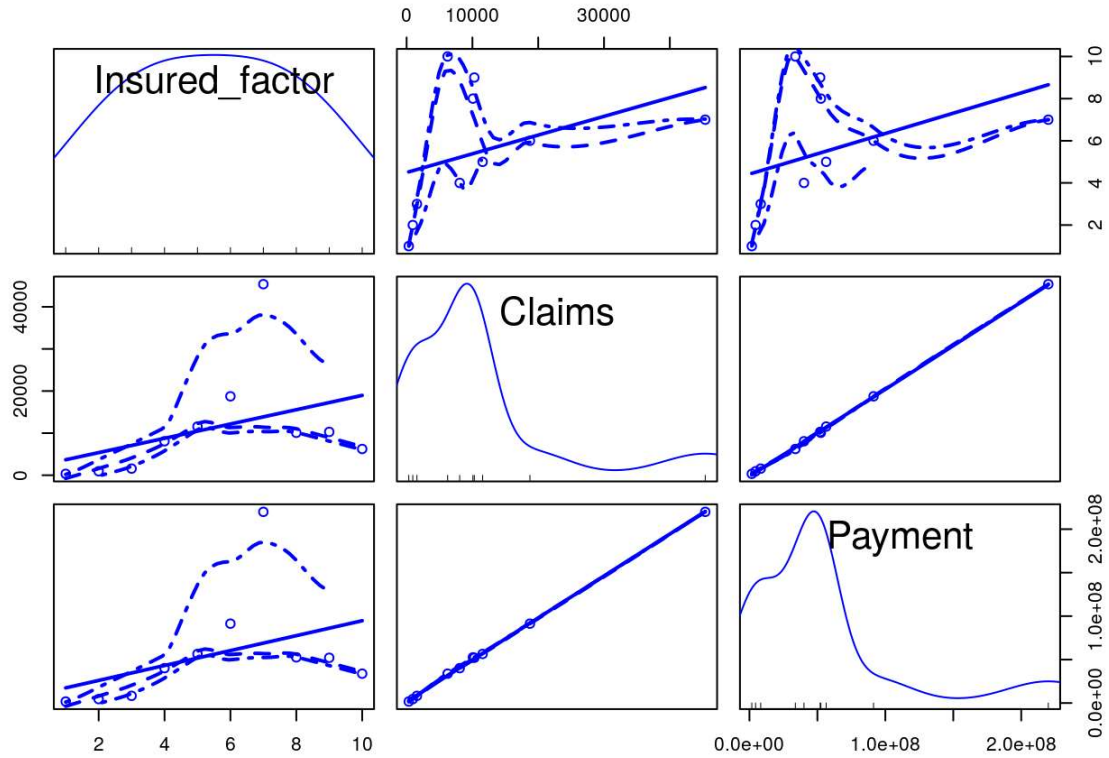
```
autoclaims_data2<- autoclaims_data[,c("Insured_factor","Insured","Claims","Payment")]
summary_3 <- autoclaims_data2 %>% group_by(Insured_factor) %>%
  summarise(across(c(Insured, Claims, Payment), sum))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

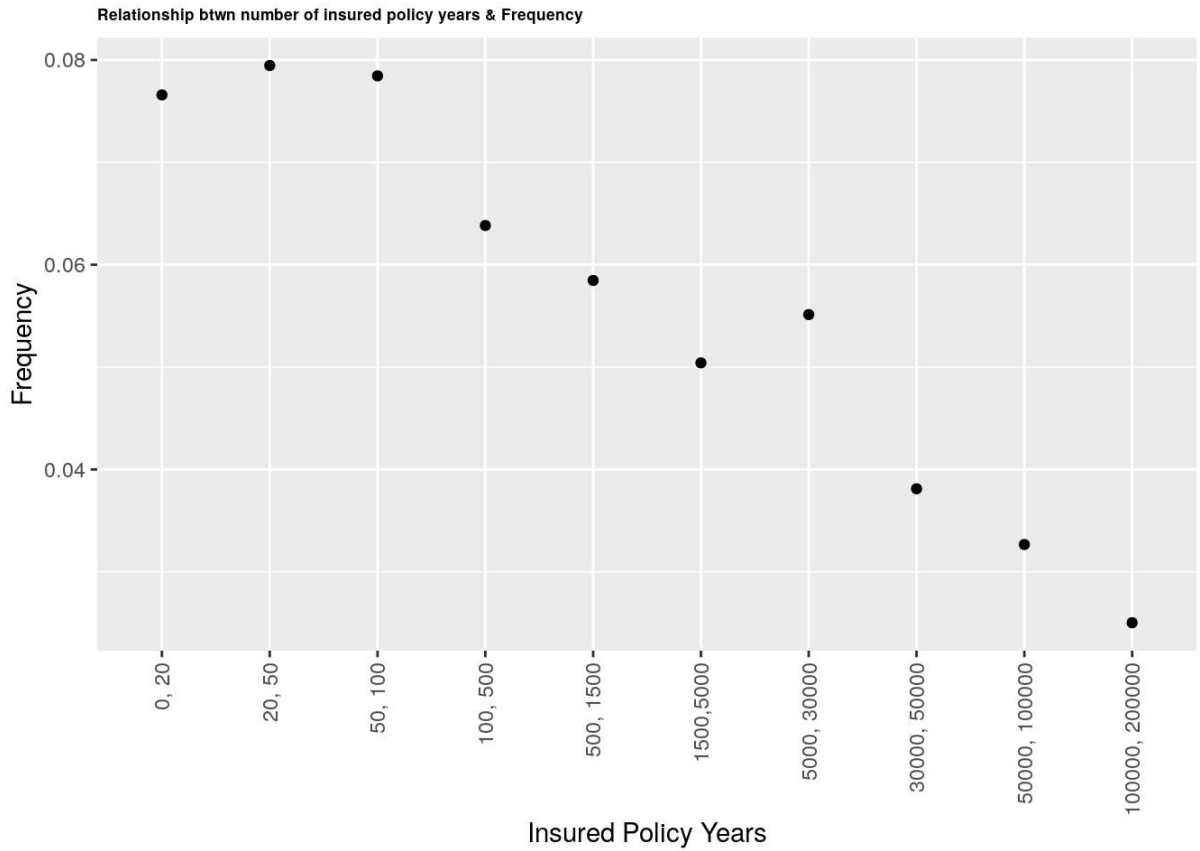
```
summary_3$Severity <- summary_3$Payment/summary_3$Claims
summary_3$Frequency <- summary_3$Claims/summary_3$Insured
summary_3$Claim_risk <- summary_3$Payment/summary_3$Insured
summary_3
```

```
## # A tibble: 10 x 7
##    Insured_factor Insured Claims   Payment Severity Frequency Claim_risk
##    <fct>            <dbl>  <int>     <int>    <dbl>     <dbl>      <dbl>
##  1 0, 20            4256.    326   1741500    5342.    0.0766       409.
##  2 20, 50          11792.    937   4528985    4833.    0.0795       384.
##  3 50, 100         20013.   1570   8286232    5278.    0.0784       414.
##  4 100, 500       126385.   8067  40142284    4976.    0.0638       318.
##  5 500, 1500      197563.  11551  56465555    4888.    0.0585       286.
##  6 1500,5000      371759.  18742  91331394    4873.    0.0504       246.
##  7 5000, 30000    823166.  45381 219915266    4846.    0.0551       267.
##  8 30000, 50000   264378.  10077  52537933    5214.    0.0381       199.
##  9 50000, 100000  314879.  10288  52056344    5060.    0.0327       165.
## 10 100000, 200000 248980.   6232  33785188    5421.    0.0250       136.
```

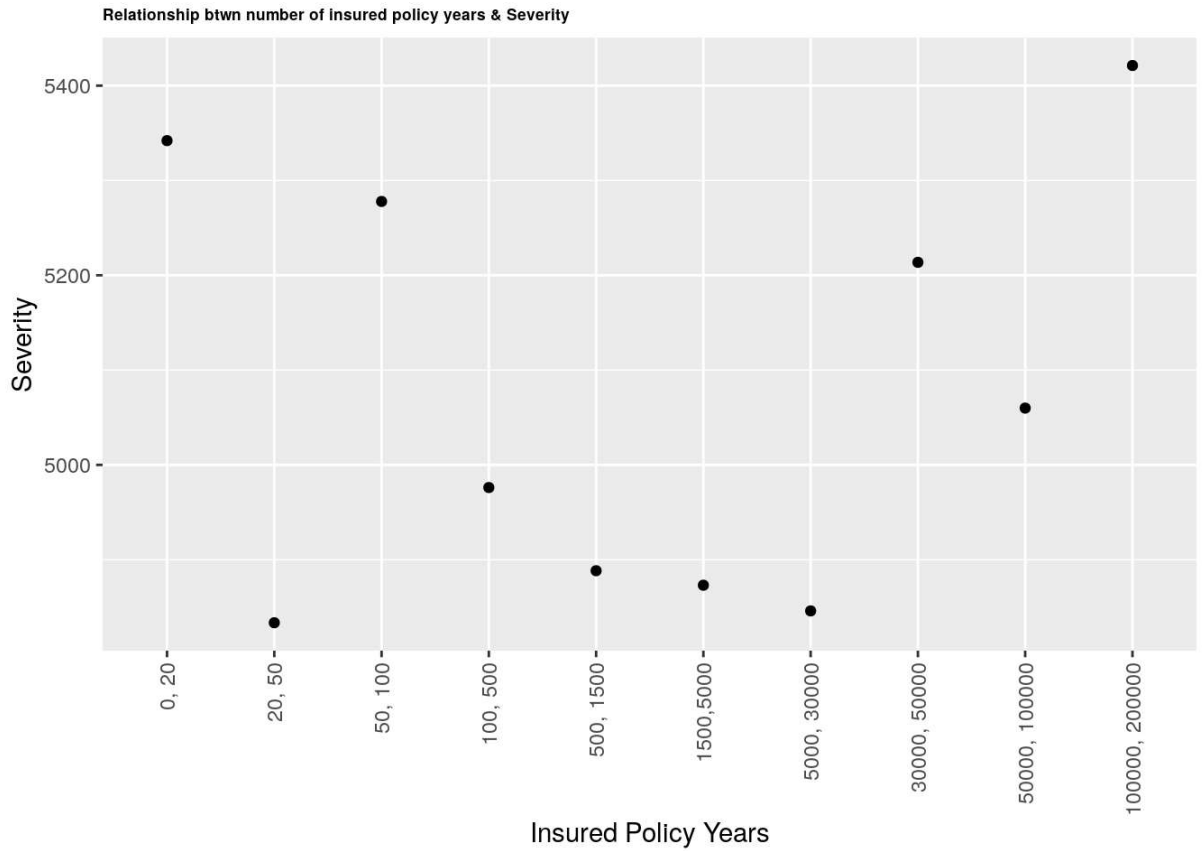
```
#Scatter Plots
options(repr.plot.width=9, repr.plot.height=9)
scatterplotMatrix(~ Insured_factor + Claims + Payment    ,data=summary_3)
```



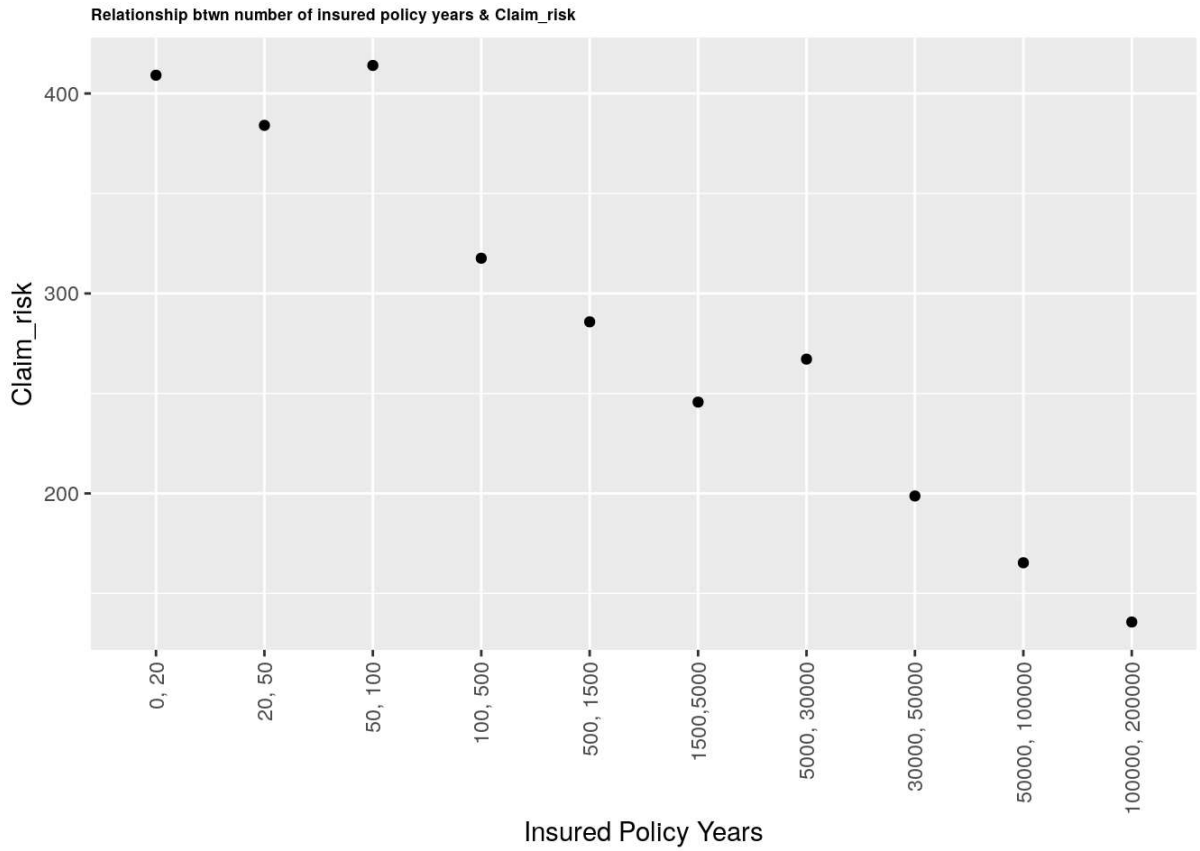

```
require(ggplot2)
require(gridExtra)
plot1<- ggplot(summary_3, aes(x = Insured_factor, y = Frequency)) + geom_point() +
    xlab('Insured Policy Years') + ylab('Frequency') +
    ggtitle('Relationship btwn number of insured policy years & Frequency') +
    theme(plot.title = element_text(size = 6.5, face = "bold"))
plot2 <- ggplot(summary_3, aes(x = Insured_factor, y = Severity)) + geom_point() +
    xlab('Insured Policy Years') + ylab('Severity') +
    ggtitle('Relationship btwn number of insured policy years & Severity') +
    theme(plot.title = element_text(size = 6.5, face = "bold"))
plot3 <- ggplot(summary_3, aes(x = Insured_factor, y = Claim_risk)) + geom_point() +
    xlab('Insured Policy Years') + ylab('Claim_risk') +
    ggtitle('Relationship btwn number of insured policy years & Claim_risk') +
    theme(plot.title = element_text(size = 6.5, face = "bold"))
plot1 + theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```




```
plot2 + theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```




```
plot3 + theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```

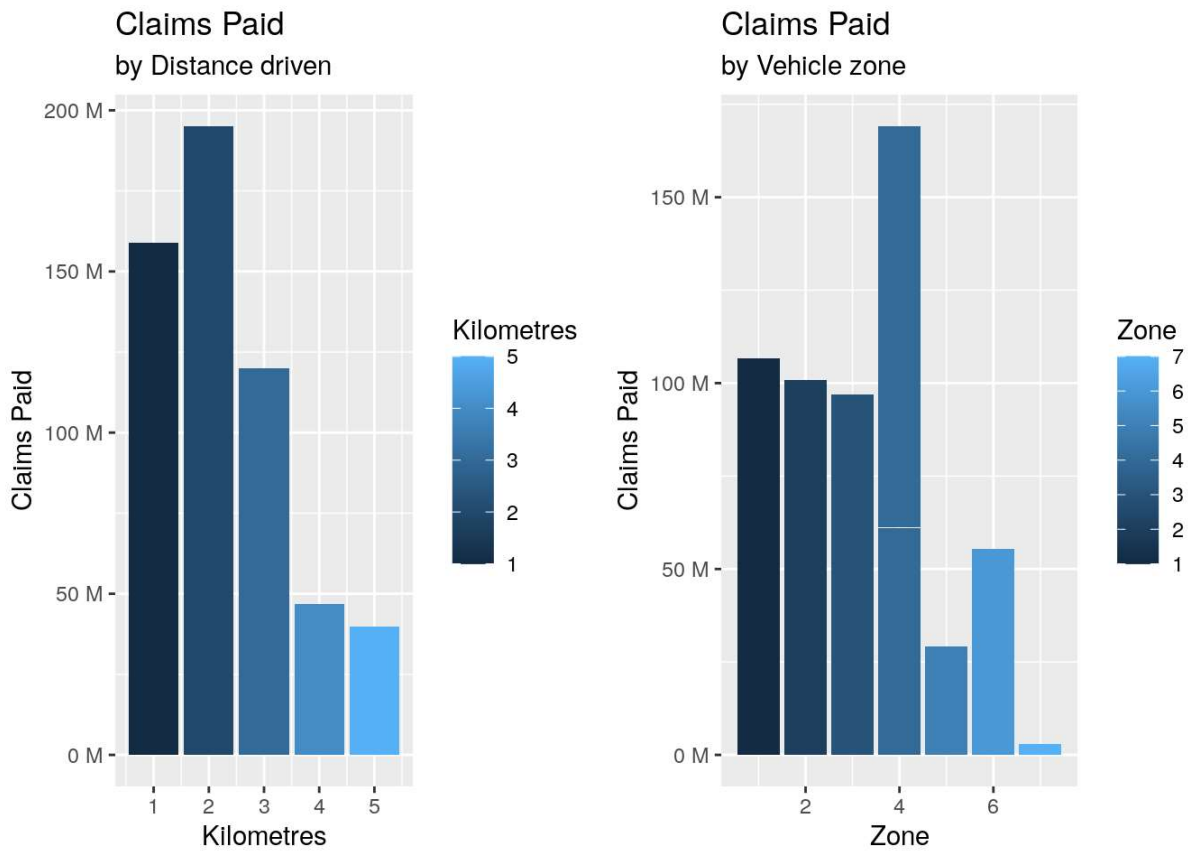



```
#plot(summary_3$Insured_factor, summary_3$Severity, type = "b",xlab="Insured Policy Years", ylab="Severity", main="Claim sev
erity by number of insured policy years")
```

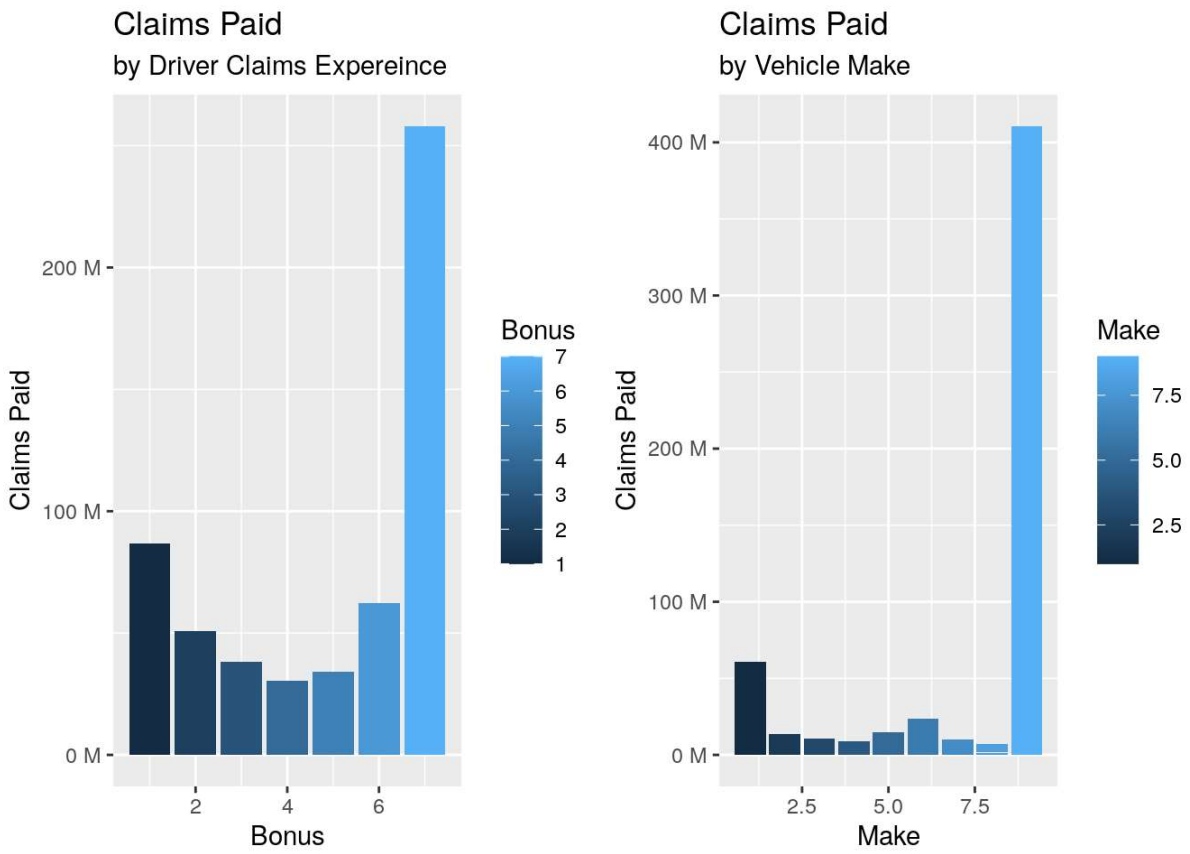
5. Variables impacting insurance payment

The committee wants to figure out the reasons for insurance payment increase and decrease. So they have decided to find whether distance, location, bonus, make, and insured amount or claims are affecting the payment or all or some of these are affecting it.

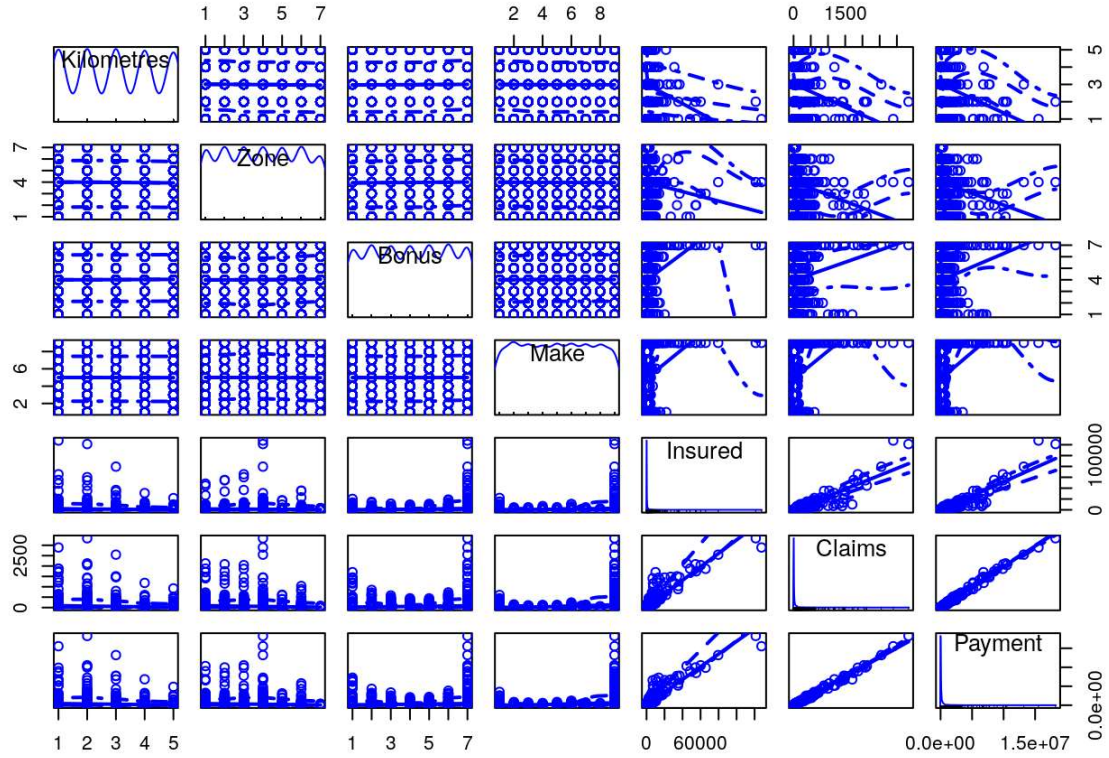
```
p2 <- autoclaims_data %>%
  ggplot(aes(x = Kilometres, y = Payment, fill = Kilometres)) +
    geom_col() +
    scale_y_continuous(labels = label_number(suffix = " M", scale = 1e-6)) + # millions
    labs(title = "Claims Paid ",
          subtitle = "by Distance driven",
              x = "Kilometres", y = "Claims Paid")

p4 <- autoclaims_data %>%
  ggplot(aes(x = Zone, y = Payment, fill = Zone)) +
    geom_col() +
    scale_y_continuous(labels = label_number(suffix = " M", scale = 1e-6)) + # millions
    labs(title = "Claims Paid",
         subtitle = "by Vehicle zone",
              x = "Zone", y = "Claims Paid")

p6 <- autoclaims_data %>%
  ggplot(aes(x = Bonus, y = Payment, fill = Bonus)) +
    geom_col() +
    scale_y_continuous(labels = label_number(suffix = " M", scale = 1e-6)) + # millions
    labs(title = "Claims Paid ",
         subtitle = "by Driver Claims Expereince",
              x = "Bonus", y = "Claims Paid")

p8 <- autoclaims_data %>%
  ggplot(aes(x = Make, y = Payment, fill = Make)) +
    geom_col() +
    scale_y_continuous(labels = label_number(suffix = " M", scale = 1e-6)) + # millions
    labs(title = "Claims Paid ",
         subtitle = "by Vehicle Make",
              x = "Make", y = "Claims Paid")



grid.arrange(p2, p4, nrow = 1)
```



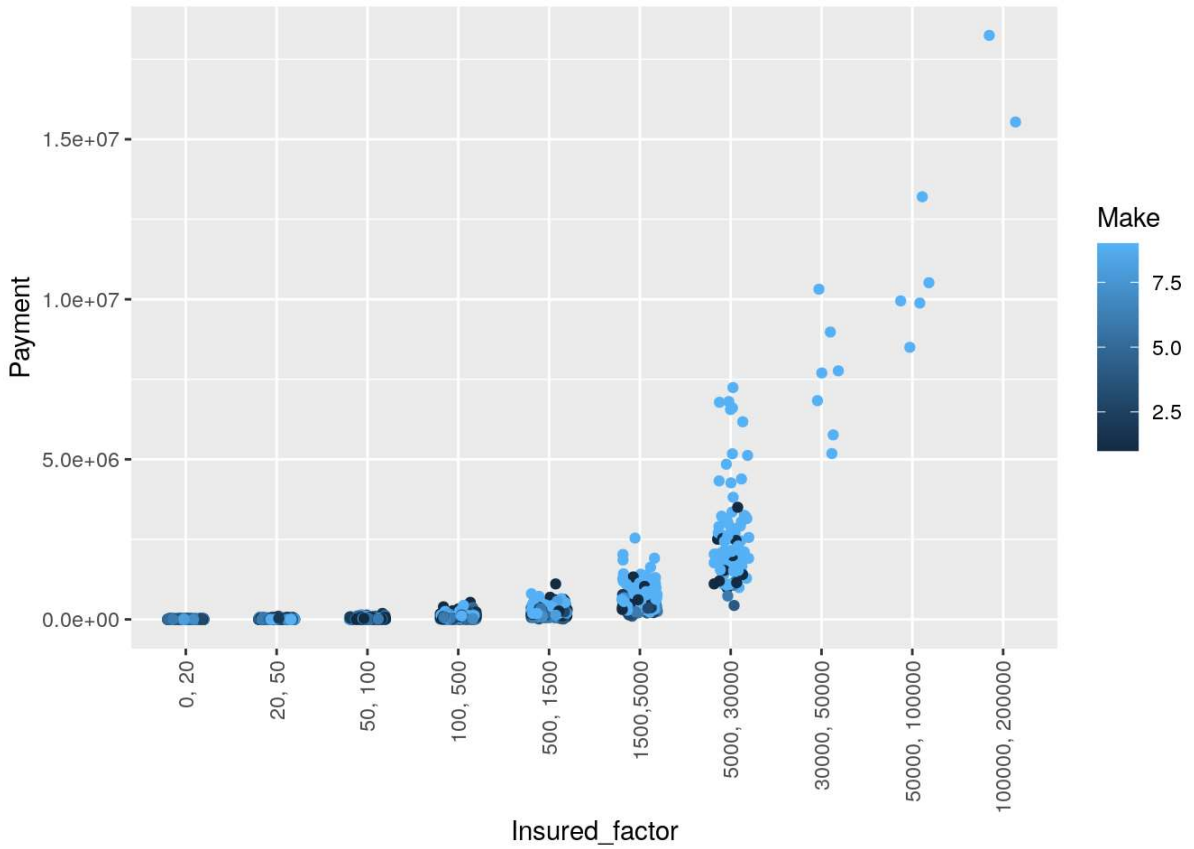

```
grid.arrange(p6, p8, nrow = 1)
```




```
require(car)
require(repr)
options(repr.plot.width=9, repr.plot.height=9)
scatterplotMatrix(~ Kilometres + Zone + Bonus + Make + Insured + Claims + Payment    ,data=autoclaims_data)
```




```
plot4 <- ggplot(autoclaims_data, aes(x = Insured_factor, y = Payment, color = Make)) +
  geom_jitter(width = .2)
plot4 + theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```



6. Deciding location for new Branch

The insurance company is planning to establish a new branch office, so they are interested to find at what location, kilometre, and bonus level their insured amount, claims, and payment get increased.

It will be good to select a location with hihger number of Insured amount, good drivers (favorable Bonus) and less Claim risk

```
#
#autoclaims_data
autoclaims_data_2 <- autoclaims_data[,c(-4)]

autoclaims_data_2$Insured_factor <- cut(autoclaims_data_2$Insured, breaks = c(0, 20, 50, 100, 500, 1500,5000, 30000, 50000,
100000, 200000),
    labels = c("0, 20", "20, 50", "50, 100", "100, 500", "500, 1500", "1500,5000","5000, 30000","30000, 50000", "50000, 1000
00", "100000, 200000"  ))
autoclaims_data_2$Severity   <- autoclaims_data_2$Payment/autoclaims_data_2$Claims
autoclaims_data_2$Frequency  <- autoclaims_data_2$Claims/autoclaims_data_2$Insured
autoclaims_data_2$Claim_risk <- autoclaims_data_2$Payment/autoclaims_data_2$Insured
head(autoclaims_data_2)
```

```
##   Kilometres Zone Bonus Insured Claims Payment Insured_factor Severity
## 1          1    1     1  455.13    108  392491       100, 500 3634.176
## 2          1    1     1   69.17     19   46221        50, 100 2432.684
## 3          1    1     1   72.88     13   15694        50, 100 1207.231
## 4          1    1     1 1292.39    124  422201      500, 1500 3404.847
## 5          1    1     1  191.01     40  119373       100, 500 2984.325
## 6          1    1     1  477.66     57  170913       100, 500 2998.474
##    Frequency Claim_risk
## 1 0.23729484   862.3712
## 2 0.27468556   668.2232
## 3 0.17837541   215.3403
## 4 0.09594627   326.6823
## 5 0.20941312   624.9568
## 6 0.11933174   357.8131
```

```
#replacing NA s in Sevrity with value 0
autoclaims_data_2 <- autoclaims_data_2 %>% mutate(Severity = ifelse(is.na(Severity), 0, Severity))

#converting these variables to factor to use in group by statement.
autoclaims_data_2$Zone_f <- as.factor(autoclaims_data_2$Zone)
autoclaims_data_2$Kilometres_f <- as.factor(autoclaims_data_2$Kilometres)
autoclaims_data_2$Bonus_f <- as.factor(autoclaims_data_2$Bonus)

#summary report
#autoclaims_data3<- autoclaims_data[,c("Insured_factor","Insured","Claims","Payment")]
autoclaims_data6 <- autoclaims_data_2[, c("Zone_f" , "Kilometres_f" , "Bonus_f" , "Insured" , "Claims" , "Payment" , "Severi
ty" , "Frequency" ,"Claim_risk")]

summary_5 <- as.data.frame(autoclaims_data6 %>% group_by(Zone_f ) %>%
  summarise(across(c(Insured, Claims, Payment , Severity, Frequency, Claim_risk), sum)))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
format(summary_5, scientific = FALSE, big.mark = ',')
```

```
##   Zone_f    Insured Claims     Payment    Severity Frequency Claim_risk
## 1      1 326,394.10 23,174 106,633,468 1,463,024.3  32.61842 161,310.04
## 2      2 387,916.78 21,302 100,775,278 1,347,868.4  25.03624 111,987.36
## 3      3 429,331.99 19,938  96,878,519 1,514,008.4  22.74706 112,860.09
## 4      4 847,154.83 31,913 169,177,603 1,655,847.9  18.11293  95,291.90
## 5      5 120,442.99  5,962  29,109,577 1,327,481.7  19.59692 117,149.77
## 6      6 252,845.64 10,262  55,291,468 1,505,368.0  17.92239 107,957.10
## 7      7  19,083.75    620   2,924,768   541,679.8  14.83013  58,724.44
```

```
summary_6 <- as.data.frame(autoclaims_data6 %>% group_by(Zone_f, Kilometres_f ) %>%
  summarise(across(c(Insured, Claims, Payment, Severity, Frequency,Claim_risk  ), sum)))
```

```
## `summarise()` regrouping output by 'Zone_f' (override with `.groups` argument)
```

```
format(summary_6, scientific = FALSE, big.mark = ',')
```

```
##    Zone_f Kilometres_f    Insured Claims    Payment   Severity Frequency
## 1       1            1 113,842.13  7,169 30,902,212 283,069.08  5.656198
## 2       1            2 112,304.44  8,148 38,692,904 346,853.88  6.357135
## 3       1            3  65,150.23  4,794 22,244,956 320,629.87  6.008168
## 4       1            4  20,777.63  1,566  7,409,238 237,472.12  6.507723
## 5       1            5  14,319.67  1,497  7,384,158 274,999.31  8.089201
## 6       2            1 130,747.13  6,294 28,896,696 266,661.64  4.148897
## 7       2            2 132,998.07  7,507 35,578,420 275,369.31  5.188981
## 8       2            3  77,460.12  4,409 21,058,474 273,953.85  5.085012
## 9       2            4  27,222.41  1,649  8,184,516 243,065.64  5.820746
## 10      2            5  19,489.05  1,443  7,057,172 288,817.95  4.792600
## 11      3            1 149,560.59  5,964 28,346,933 280,936.85  3.561140
## 12      3            2 144,427.66  6,940 32,879,379 330,177.29  3.957678
## 13      3            3  82,837.72  4,183 21,332,535 348,007.04  4.590923
## 14      3            4  30,550.17  1,519  7,628,655 304,441.37  4.391222
## 15      3            5  21,955.85  1,332  6,691,017 250,445.89  6.246099
## 16      4            1 279,934.65  8,619 44,563,016 331,490.43  2.655005
## 17      4            2 281,017.28 10,865 57,949,336 339,848.15  3.424067
## 18      4            3 173,991.26  7,055 37,192,080 352,500.05  3.579203
## 19      4            4  66,015.31  2,992 16,553,313 295,735.52  4.200313
## 20      4            5  46,196.33  2,382 12,919,858 336,273.74  4.254340
## 21      5            1  43,765.69  2,031  9,197,220 280,838.89  3.513416
## 22      5            2  42,038.74  2,141 10,234,924 271,811.59  4.603964
## 23      5            3  22,021.66  1,089  5,684,050 323,390.81  3.886196
## 24      5            4   7,322.34    400  2,297,679 264,046.57  3.790356
## 25      5            5   5,294.56    301  1,695,704 187,393.87  3.802992
## 26      6            1  81,889.78  2,895 16,100,355 331,049.05  2.863593
## 27      6            2  85,120.54  3,574 18,799,599 313,184.13  3.194535
## 28      6            3  52,197.90  2,218 11,782,181 264,167.68  3.622777
## 29      6            4  19,990.45    858  4,752,367 281,678.90  3.979730
## 30      6            5  13,646.97    717  3,856,966 315,288.19  4.261753
## 31      7            1   7,061.38    214    867,383  97,293.76  2.219690
## 32      7            2   6,489.99    196  1,018,425 145,001.10  1.976110
## 33      7            3   3,490.49    137    663,273 175,738.08  4.870095
## 34      7            4   1,271.73     41    138,850  34,819.60  1.904237
## 35      7            5     770.16     32    236,837  88,827.25  3.859998
##    Claim_risk
## 1  24,903.561
## 2  35,083.967
## 3  32,359.654
## 4  26,116.887
## 5  42,845.967
## 6  18,153.482
```

```
## 7  23,870.173
## 8  22,292.701
## 9  24,489.089
## 10 23,181.915
## 11 16,707.400
## 12 19,156.198
## 13 24,900.067
## 14 24,243.725
## 15 27,852.705
## 16 15,070.125
## 17 19,385.286
## 18 18,926.365
## 19 19,993.683
## 20 21,916.437
## 21 22,291.588
## 22 22,130.381
## 23 22,239.234
## 24 22,632.878
## 25 27,855.692
## 26 15,634.122
## 27 16,586.509
## 28 16,536.051
## 29 22,075.175
## 30 37,125.247
## 31  7,662.238
## 32  9,562.725
## 33 25,939.146
## 34  4,246.416
## 35 11,313.917
```

```
summary_7 <- as.data.frame(autoclaims_data6 %>% group_by(Zone_f, Bonus_f ) %>%
  summarise(across(c(Insured, Claims, Payment, Severity, Frequency,Claim_risk  ), sum)))
```

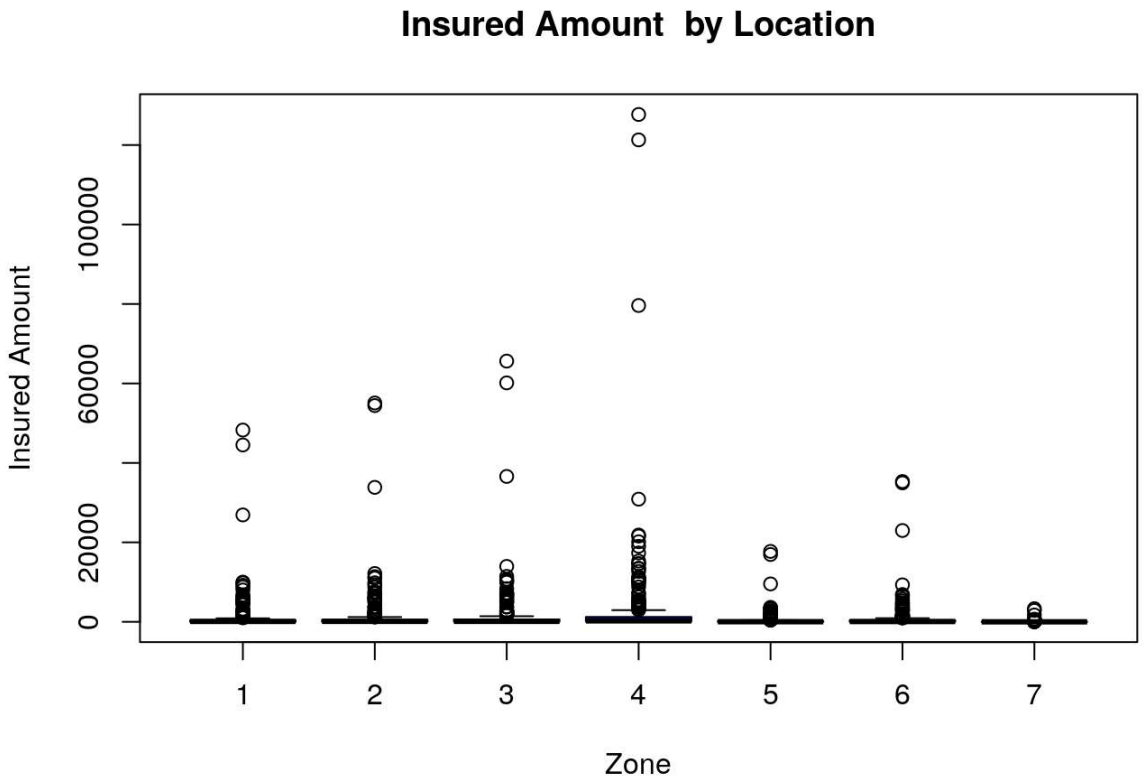
```
## `summarise()` regrouping output by 'Zone_f' (override with `.groups` argument)
```

```
format(summary_7, scientific = FALSE, big.mark = ',')
```

```
##    Zone_f Bonus_f    Insured Claims    Payment   Severity Frequency Claim_risk
## 1       1       1  25,908.04  4,805 20,114,805 170,821.67  9.355674 37,862.619
## 2       1       2  20,862.02  2,339 10,792,381 231,848.34  5.270841 31,844.061
## 3       1       3  18,130.03  1,723  8,749,561 197,007.23  5.380865 26,006.516
## 4       1       4  16,758.94  1,369  6,202,409 208,331.01  3.482523 18,241.101
## 5       1       5  20,363.31  1,504  6,250,052 195,028.91  3.581677 17,745.305
## 6       1       6  39,372.38  2,814 12,942,421 252,106.13  3.374524 19,711.396
## 7       1       7 184,999.38  8,620 41,581,839 207,880.98  2.172321  9,899.037
## 8       2       1  27,884.77  3,689 16,766,460 173,572.52  6.867939 28,164.750
## 9       2       2  23,457.75  2,026  8,924,623 171,795.98  4.331089 19,586.275
## 10      2       3  20,797.47  1,474  6,465,914 185,990.48  3.753417 15,627.092
## 11      2       4  19,423.38  1,241  6,180,504 208,822.63  2.842624 14,444.359
## 12      2       5  23,949.91  1,432  5,971,087 195,115.48  2.716958 12,529.282
## 13      2       6  44,893.23  2,498 11,579,067 201,388.72  2.631811 12,564.067
## 14      2       7 227,510.27  8,942 44,887,623 211,182.59  1.892398  9,071.535
## 15      3       1  28,628.46  3,332 14,438,146 217,200.65  6.187985 28,932.733
## 16      3       2  25,604.94  1,825  8,310,057 244,429.62  4.611053 22,618.182
## 17      3       3  22,593.23  1,326  6,238,300 223,808.51  2.829429 16,143.116
## 18      3       4  20,420.36  1,142  5,550,307 198,846.58  2.698508 13,896.931
## 19      3       5  25,269.03  1,278  5,888,425 198,101.97  2.390016 11,266.369
## 20      3       6  45,637.17  2,214 10,913,239 194,210.65  2.384739 11,295.039
## 21      3       7 261,178.80  8,821 45,540,045 237,410.47  1.645332  8,707.724
## 22      4       1  52,262.50  4,660 22,511,942 252,261.10  4.598001 24,914.454
## 23      4       2  47,053.64  2,922 15,220,299 223,583.12  3.547253 17,576.048
## 24      4       3  41,208.02  2,109 11,427,109 239,836.15  2.535006 13,939.021
## 25      4       4  37,133.83  1,678  8,184,885 226,616.83  2.156552 10,507.222
## 26      4       5  45,548.93  1,905 10,120,376 206,250.63  2.026772  9,194.866
## 27      4       6  82,952.16  3,298 17,783,713 266,024.11  1.861326 11,526.869
## 28      4       7 540,995.75 15,341 83,929,279 241,275.95  1.388018  7,633.416
## 29      5       1   8,357.26    969  4,279,402 188,297.26  5.259055 34,419.726
## 30      5       2   7,503.02    559  2,750,285 141,180.58  2.735129 13,507.710
## 31      5       3   6,480.56    391  1,575,107 121,897.08  3.064033 13,977.765
## 32      5       4   5,836.38    326  1,613,040 183,750.40  2.574192 18,535.333
## 33      5       5   7,305.42    401  2,322,237 235,666.03  1.739605 14,003.699
## 34      5       6  14,011.07    709  3,340,960 213,702.60  2.579956 13,424.471
## 35      5       7  70,949.28  2,607 13,228,546 242,987.78  1.644953  9,281.069
## 36      6       1  16,982.05  1,637  8,359,421 196,457.51  5.039922 30,402.340
## 37      6       2  15,049.95    946  4,600,796 213,921.77  2.582681 18,018.505
## 38      6       3  13,033.16    678  3,456,850 181,599.37  2.475812 12,751.166
## 39      6       4  11,300.46    526  2,708,865 205,545.40  2.489710 15,906.555
## 40      6       5  13,434.77    582  3,366,348 213,283.64  2.096545 12,773.044
## 41      6       6  25,156.36    992  5,428,177 227,210.59  1.694094  9,165.894
## 42      6       7 157,888.89  4,901 27,371,011 267,349.69  1.543623  8,939.601
```

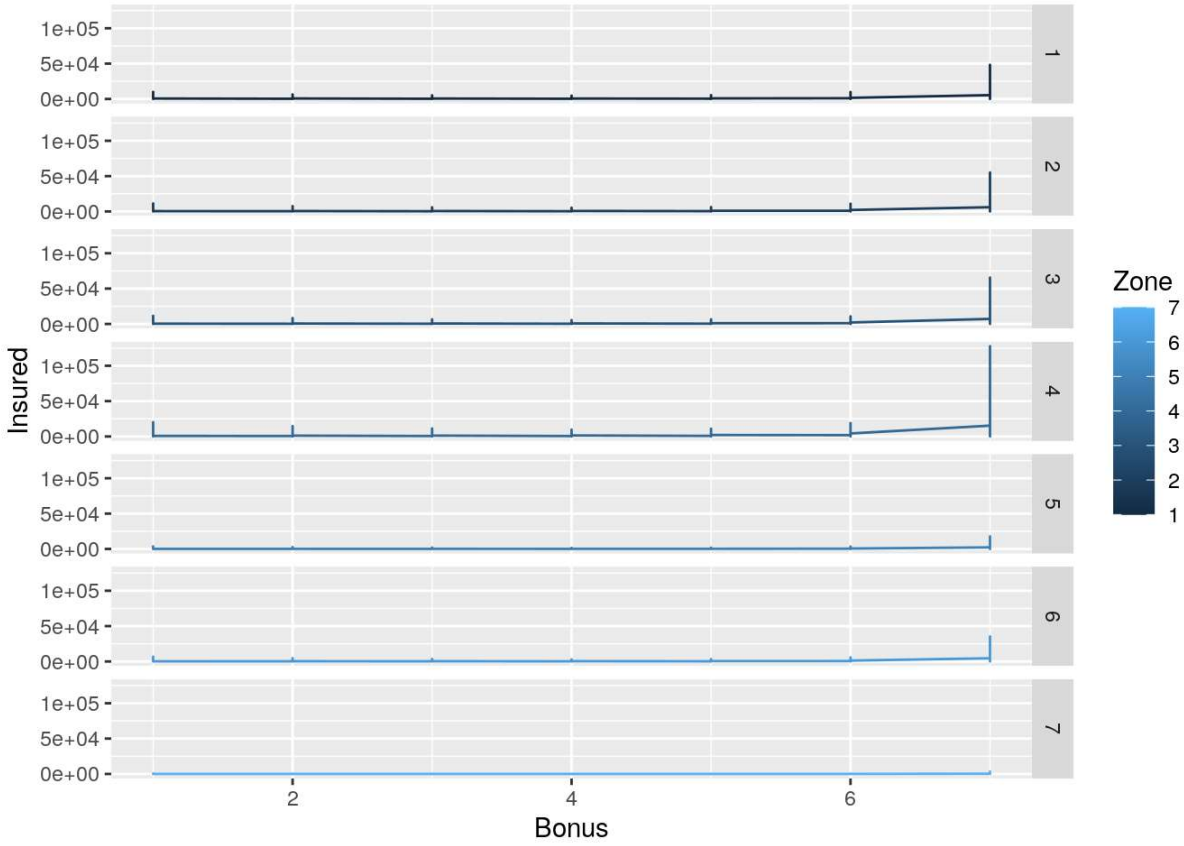
```
## 43      7       1   1,320.83     97    386,876  39,567.07  2.319608  4,613.641
## 44      7       2   1,204.22     64    356,346 114,095.08  1.620125 18,676.043
## 45      7       3     974.39     41    110,573  55,022.70  0.922369  3,423.706
## 46      7       4     846.53     27     94,407  29,813.80  4.176433 11,168.978
## 47      7       5   1,032.83     41    132,903  37,897.54  2.650869  8,244.697
## 48      7       6   1,809.90     57    295,426  63,028.81  1.968266  5,705.150
## 49      7       7  11,895.05    293  1,548,237 202,254.79  1.172460  6,892.228
```

```
#visualisations
p12 <- boxplot(Insured ~ Zone, data=autoclaims_data, xlab="Zone",
      ylab="Insured Amount",  col="blue", main="Insured Amount  by Location")
```

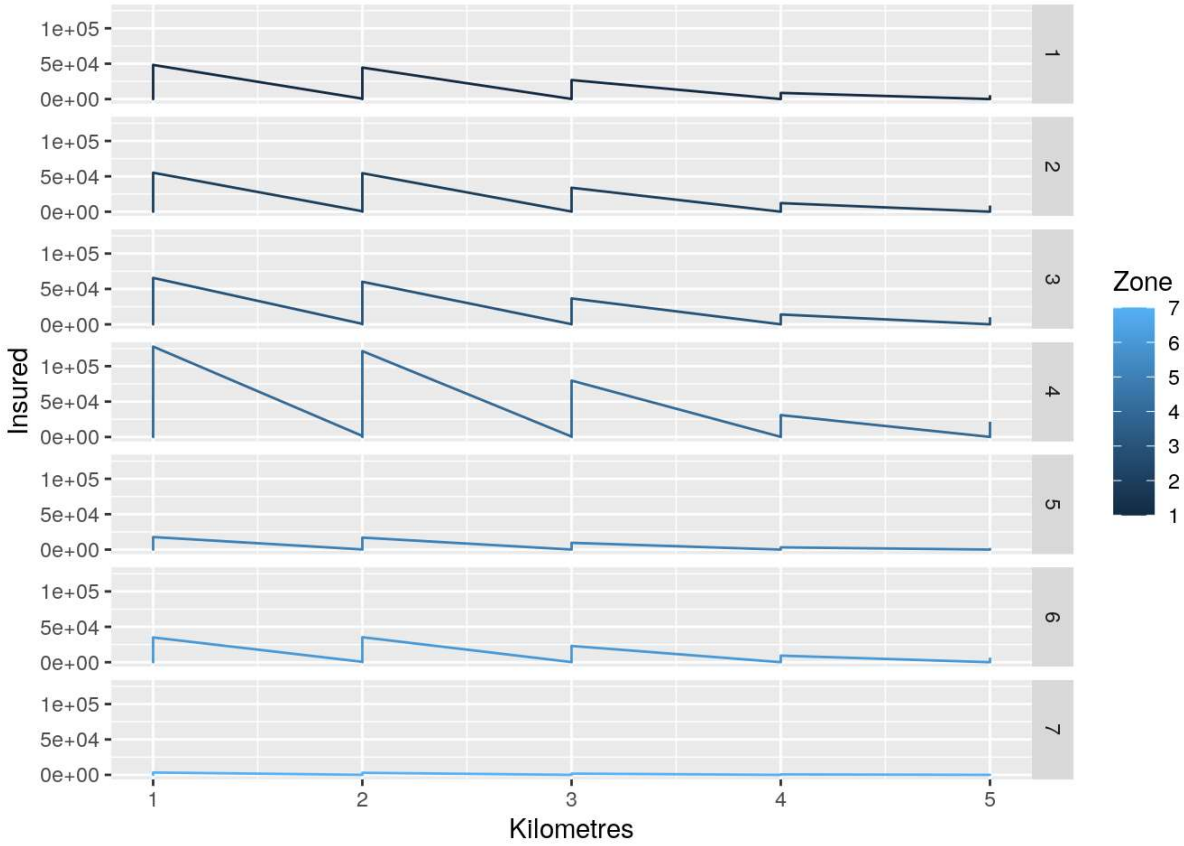


Higher Insured amounts in zones 4 and 3.

```
plot14  <- ggplot(autoclaims_data_2, aes(Bonus, Insured, col = Zone)) +             # Create ggplot2 plot
  geom_line()

plot14 + facet_grid(Zone ~ .)
```

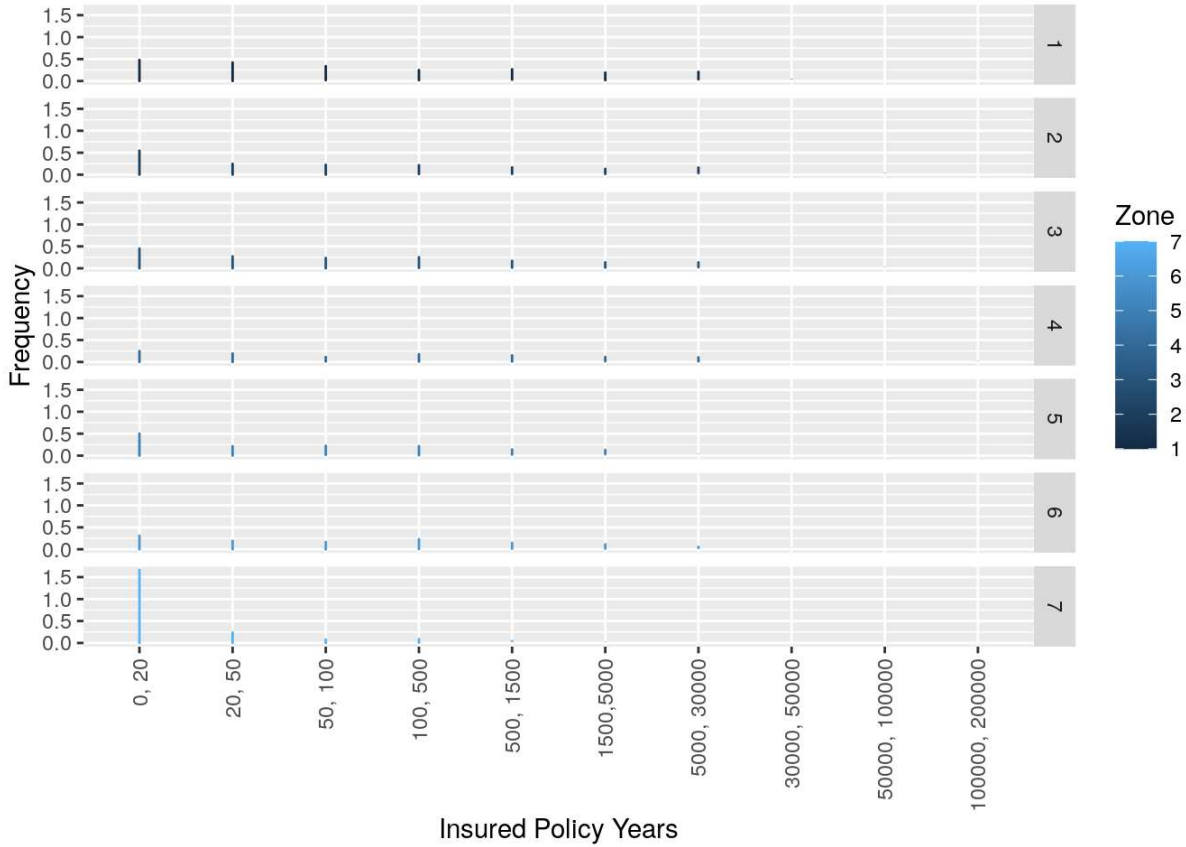



```
plot15  <- ggplot(autoclaims_data_2, aes(Kilometres, Insured, col = Zone)) +          # Create ggplot2 plot
  geom_line()

plot15 + facet_grid(Zone ~ .)
```

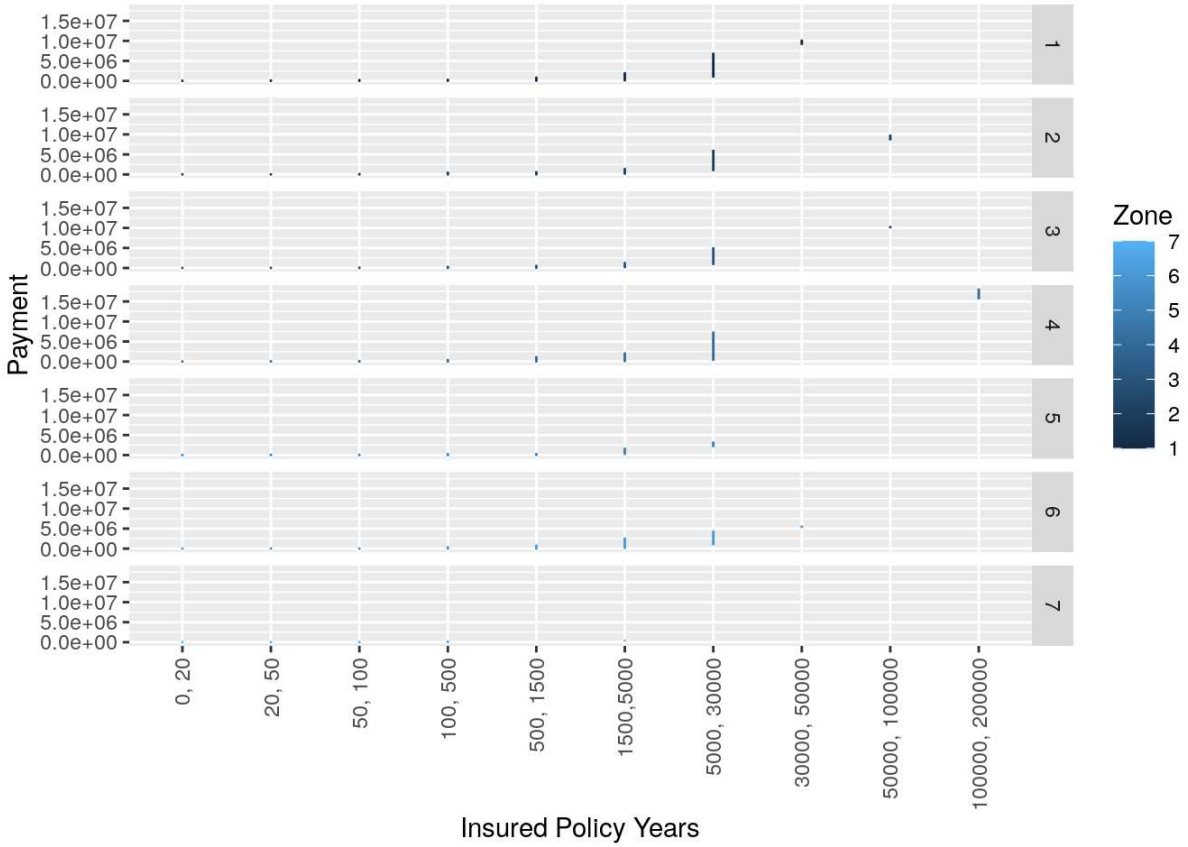


Good driver expereince in zones 4,3,2 and 1.

```
plot16  <- ggplot(autoclaims_data_2, aes(Insured_factor, Frequency, col = Zone)) + xlab('Insured Policy Years') +
# Create ggplot2 plot
  geom_line()

plot16 + facet_grid(Zone ~ .) + theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```




```
plot17  <- ggplot(autoclaims_data_2, aes(Insured_factor, Payment, col = Zone)) + xlab('Insured Policy Years') +
# Create ggplot2 plot
  geom_line()

plot17 + facet_grid(Zone ~ .) + theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```



High claim payments and high insured amounts in zones 4, 1 and 2.

Good driving record in zones 4,3,2 and 1.

High claim frequency in zone 7, 1 and 2.

7. Insurance factors Identification

The committee wants to understand what affects their claim rates so as to decide the right premiums for a certain set of situations. Hence, they need to find whether the insured amount, zone, kilometer, bonus, or make affects the claim rates and to what extent.

**Correlation analysis between variables

```
# Spearman correlation matrix between all variables
#library(Hmisc)
autoclaims_data_7 <- autoclaims_data[,c(-8)]
rcorr(as.matrix(autoclaims_data_7), type = 'spearman')
```

```
##            Kilometres  Zone Bonus  Make Insured Claims Payment
## Kilometres       1.00 -0.01  0.01  0.00   -0.33  -0.26   -0.24
## Zone            -0.01  1.00  0.01 -0.01   -0.32  -0.39   -0.36
## Bonus            0.01  0.01  1.00  0.00    0.35   0.20    0.20
## Make             0.00 -0.01  0.00  1.00    0.11   0.11    0.12
## Insured         -0.33 -0.32  0.35  0.11    1.00   0.93    0.90
## Claims          -0.26 -0.39  0.20  0.11    0.93   1.00    0.96
## Payment         -0.24 -0.36  0.20  0.12    0.90   0.96    1.00
##
## n= 2182
##
##
## P
##            Kilometres Zone   Bonus  Make   Insured Claims Payment
## Kilometres            0.5155 0.7366 0.9006 0.0000  0.0000 0.0000
## Zone       0.5155             0.5857 0.8086 0.0000  0.0000 0.0000
## Bonus      0.7366     0.5857        0.9198 0.0000  0.0000 0.0000
## Make       0.9006     0.8086 0.9198        0.0000  0.0000 0.0000
## Insured    0.0000     0.0000 0.0000 0.0000         0.0000 0.0000
## Claims     0.0000     0.0000 0.0000 0.0000 0.0000         0.0000
## Payment    0.0000     0.0000 0.0000 0.0000 0.0000  0.0000
```

The first table includes the correlation coefficients, the second their p values.

Payment is strongly correlated with both claims and insured and their p values are significant.

Insured and claims are also strongly correlated with each other which reveals a possible multicollinearity.

A few correlations involving categorical values are not statistically significant with a very high p value.

All coefficients (variable - Payment) are statistically significant since their p values are < 0.05.

Claims and insured have a strong correlation with payment, respectively 0.96 and 0.90.

Selecting variables for regression:

Insured and claims are strongly corelated. We wil consider Claims in the regression model. We are considering following variables in the regression model:

- Claims
- Kilometres
- Bonus
- Zone
- Make

```r
# Preparing categorical data for regression
autoclaims_data_d <- autoclaims_data
autoclaims_data_d$Make <- as.factor(autoclaims_data_d$Make)
autoclaims_data_d$Zone <- as.factor(autoclaims_data_d$Zone)
autoclaims_data_d$Kilometres <- as.factor(autoclaims_data_d$Kilometres)
str(autoclaims_data_d)
```

```
## 'data.frame':    2182 obs. of  8 variables:
##  $ Kilometres    : Factor w/ 5 levels "1","2","3","4",..: 1 1 1 1 1 1 1 1 1 ...
##  $ Zone          : Factor w/ 7 levels "1","2","3","4",..: 1 1 1 1 1 1 1 1 1 ...
##  $ Bonus         : int  1 1 1 1 1 1 1 1 2 ...
##  $ Make          : Factor w/ 9 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 1 ...
##  $ Insured       : num  455.1 69.2 72.9 1292.4 191 ...
##  $ Claims        : int  108 19 13 124 40 57 23 14 1704 45 ...
##  $ Payment       : int  392491 46221 15694 422201 119373 170913 56940 77487 6805992 214011 ...
##  $ Insured_factor: Factor w/ 10 levels "0, 20","20, 50",..: 4 3 3 5 4 4 4 2 7 4 ...
```

```r
# Splitting data for train and test
sample_rec <- sort(sample(nrow(autoclaims_data_d), nrow(autoclaims_data_d)*.7))
train<-autoclaims_data_d[sample_rec,]
test<-autoclaims_data_d[-sample_rec,]

# building a linear regression model to predict payment using other variables
# lm_model1 : Payment ~ Kilometres + Zone + Bonus + Make + Claims
lm_model1 <- lm(Payment ~ Kilometres + Zone +   Bonus + Make + Claims, data = train)
summary(lm_model1)
```

```
##
## Call:
## lm(formula = Payment ~ Kilometres + Zone + Bonus + Make + Claims,
##     data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -911602  -21841      30   21275 1109441
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -49894.65   10409.64  -4.793 1.80e-06 ***
## Kilometres2   9576.30    7014.67   1.365 0.172400
## Kilometres3  16251.35    6949.97   2.338 0.019500 *
## Kilometres4  23072.87    6969.44   3.311 0.000953 ***
## Kilometres5  20297.83    7117.35   2.852 0.004405 **
## Zone2        -1379.19    8246.82  -0.167 0.867204
## Zone3        13288.69    8116.76   1.637 0.101799
## Zone4        44450.69    8250.38   5.388 8.27e-08 ***
## Zone5        22286.70    8148.73   2.735 0.006311 **
## Zone6        42414.43    8332.18   5.090 4.02e-07 ***
## Zone7        27296.13    8532.74   3.199 0.001408 **
## Bonus         5813.49    1122.99   5.177 2.56e-07 ***
## Make2        -9441.94    9387.91  -1.006 0.314695
## Make3        -4060.07    9290.60  -0.437 0.662168
## Make4       -15391.33    9374.90  -1.642 0.100849
## Make5       -13683.90    9329.41  -1.467 0.142653
## Make6        -9973.51    9058.08  -1.101 0.271045
## Make7       -10715.15    9345.08  -1.147 0.251725
## Make8        -1011.46    9257.70  -0.109 0.913014
## Make9       -83495.19   10249.04  -8.147 7.78e-16 ***
## Claims        5145.07      12.51 411.434  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 86500 on 1506 degrees of freedom
## Multiple R-squared:  0.9938, Adjusted R-squared:  0.9938
## F-statistic: 1.215e+04 on 20 and 1506 DF,  p-value: < 2.2e-16
```

```
print(paste("lm_model1 AIC is: ", AIC(lm_model1)))
```

```
## [1] "lm_model1 AIC is:  39073.7029877477"
```

```
print("lm_model1 VIF is:")
```

```
## [1] "lm_model1 VIF is:"
```

```
vif(lm_model1)
```

```
##                GVIF Df GVIF^(1/(2*Df))
## Kilometres 1.041128  4        1.005051
## Zone       1.059273  6        1.004810
## Bonus      1.024802  1        1.012325
## Make       1.399676  8        1.021237
## Claims     1.463626  1        1.209804
```

lm_model1 results :

- F-statistics p-value is significant.
- Adjusted R-squared is very high.
- VIF does not reveal multicollinearity.

On the first model we only utilized those original variables included in the dataset and got a r-squared of 0.9951 which implies that 99.51% of the variation of Payment could be explained by the set of independent variables we have included. We could also observe that all of the independent variables we have included with the exception of Make is a statistically significant predictor of Payment (p-value less than 0.05 <- level of significance).

```
# lm_model2 : Payment ~ Kilometres + Zone + Bonus + Claims
lm_model2 <- lm(Payment ~ Kilometres + Zone +   Bonus + Claims, data = train)
summary(lm_model2)
```

```
##
## Call:
## lm(formula = Payment ~ Kilometres + Zone + Bonus + Claims, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -914760  -17633    4000   22680 1221724
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -62315.96    8751.45  -7.121 1.65e-12 ***
## Kilometres2  11230.66    7179.79   1.564  0.11798
## Kilometres3  15861.90    7116.10   2.229  0.02596 *
## Kilometres4  19889.22    7127.92   2.790  0.00533 **
## Kilometres5  19374.99    7288.03   2.658  0.00793 **
## Zone2        -2233.05    8442.84  -0.264  0.79144
## Zone3        13288.57    8316.74   1.598  0.11029
## Zone4        45365.37    8447.88   5.370 9.10e-08 ***
## Zone5        18610.15    8341.07   2.231  0.02582 *
## Zone6        37963.95    8523.11   4.454 9.04e-06 ***
## Zone7        20973.53    8714.58   2.407  0.01622 *
## Bonus         6335.91    1149.22   5.513 4.14e-08 ***
## Claims        5088.28      10.93 465.491  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 88670 on 1514 degrees of freedom
## Multiple R-squared:  0.9935, Adjusted R-squared:  0.9934
## F-statistic: 1.926e+04 on 12 and 1514 DF,  p-value: < 2.2e-16
```

```
print(paste("lm_model2 AIC is: ", AIC(lm_model2)))
```

```
## [1] "lm_model2 AIC is:  39141.5995610169"
```

```
print("lm_model2 VIF is:")
```

```
## [1] "lm_model2 VIF is:"
```

```
vif(lm_model2)
```

```
##                GVIF Df GVIF^(1/(2*Df))
## Kilometres 1.027687  4        1.003420
## Zone       1.034724  6        1.002849
## Bonus      1.021253  1        1.010571
## Claims     1.064158  1        1.031580
```

lm_model2 results :

- F-statistics p-value is significant.
- Adjusted R-squared is very high.
- VIF does not reveal multicollinearity.

Comparing lm_model1 and lm_model2: lm_model2 has higher AIC lm_model2 has slightly higher adjusted R Squared.

So lm_model1 is better to predict Payment.

```
# we will run the model with test data
lm_model1_test <- lm(Payment ~ Kilometres + Zone +  Bonus + Make + Claims, data = test)
summary(lm_model1_test)
```

```
##
## Call:
## lm(formula = Payment ~ Kilometres + Zone + Bonus + Make + Claims,
##     data = test)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1202662   -22291     -686    20760  1137937
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -53264.90   18962.15  -2.809 0.005123 **
## Kilometres2  15774.13   11726.31   1.345 0.179045
## Kilometres3  13242.67   12014.78   1.102 0.270794
## Kilometres4  18789.43   12293.98   1.528 0.126926
## Kilometres5  11845.76   12010.24   0.986 0.324360
## Zone2        19545.89   14161.36   1.380 0.168003
## Zone3        16960.04   14570.57   1.164 0.244865
## Zone4        75341.63   14054.85   5.361 1.16e-07 ***
## Zone5        27779.53   14805.27   1.876 0.061070 .
## Zone6        33854.79   13919.78   2.432 0.015286 *
## Zone7        23849.49   14247.25   1.674 0.094629 .
## Bonus         6593.94    1869.71   3.527 0.000451 ***
## Make2        -9265.13   15923.73  -0.582 0.560879
## Make3        -4676.04   16458.38  -0.284 0.776417
## Make4       -18640.03   16323.36  -1.142 0.253917
## Make5        -9022.79   16117.57  -0.560 0.575806
## Make6        -9357.92   17304.26  -0.541 0.588844
## Make7       -14549.65   16164.34  -0.900 0.368405
## Make8        -5553.93   16748.86  -0.332 0.740301
## Make9         9831.72   17225.82   0.571 0.568368
## Claims        4721.53      28.34 166.619  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 95280 on 634 degrees of freedom
## Multiple R-squared:  0.9865, Adjusted R-squared:  0.9861
## F-statistic:  2323 on 20 and 634 DF,  p-value: < 2.2e-16
```

```
print(paste("lm_model1_test AIC is: ", AIC(lm_model1_test)))
```

```
## [1] "lm_model1_test AIC is:  16900.0948024629"
```

```
print("lm_model1_test VIF is:")
```

```
## [1] "lm_model1_test VIF is:"
```

```
vif(lm_model1_test)
```

```
##                GVIF Df GVIF^(1/(2*Df))
## Kilometres 1.135907  4        1.016056
## Zone       1.161794  6        1.012576
## Bonus      1.018808  1        1.009360
## Make       1.679144  8        1.032923
## Claims     1.655451  1        1.286643
```

The regression equation can be written as

Payment = -14704.39 + 16839.78* Kilometres2 + 22285.52* Kilometres3 + 15017.35* Kilometres4 + 22107.93* Kilometres5 + -12761.1* Zone2 + -2312.81* Zone3 + 39940.93* Zone4 + 8851.57* Zone5 + 14551.98* Zone6 + 8405.92* Zone7 + 3778.04* Bonus + -16167.04* Make2 + -19636.57* Make3 + -27087.48* Make4 + -30199.86* Make5 + -19971.42* Make6 + -29646.97* Make7 + -17026.41* Make8 + -54497.24* Make9 + 4939.75* Claims

``