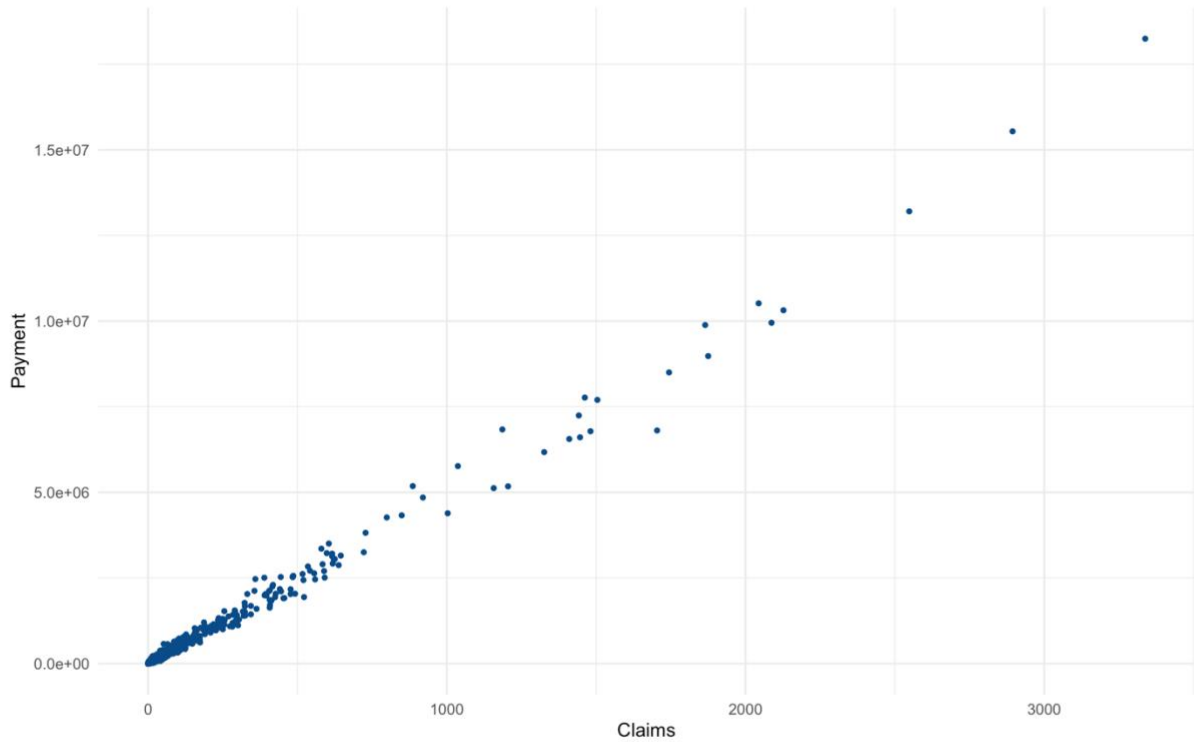
*##########    Project - 4 ######### INSURANCE ##########*

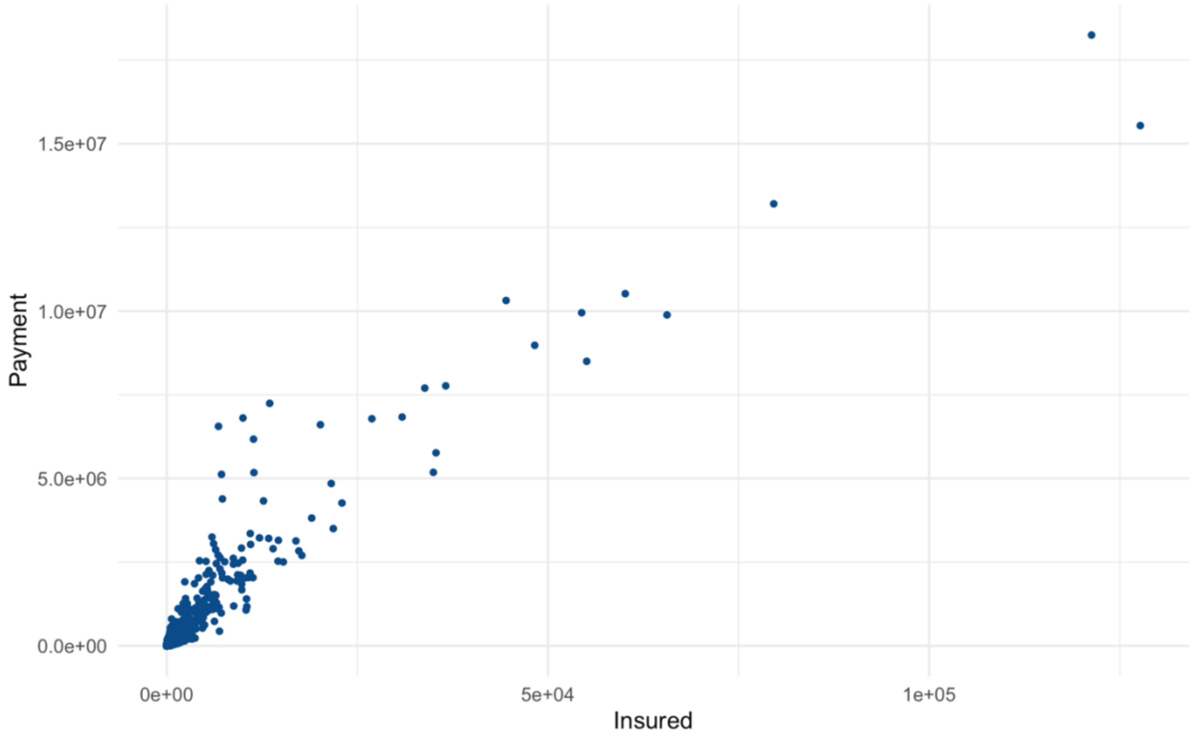
```
# Load the Insurance data to memory
ins_data <- read.csv(file.choose())
data<-ins_data
# View the data, summarise and identify the structure of data
View(ins_data)
summary(ins_data)
str(ins_data)


############            Task 1            ##################
# Descriptive analysis for the committee
# by plotting graphs with the available data
# Initial visualization to understand the data #

library(ggplot2)

ggplot(ins_data) +
 aes(x = Claims, y = Payment) +
 geom_point(size = 1L, colour = "#0c4c8a") +
 theme_minimal()
```




# The scatter plt gives us an idea that the payment increases

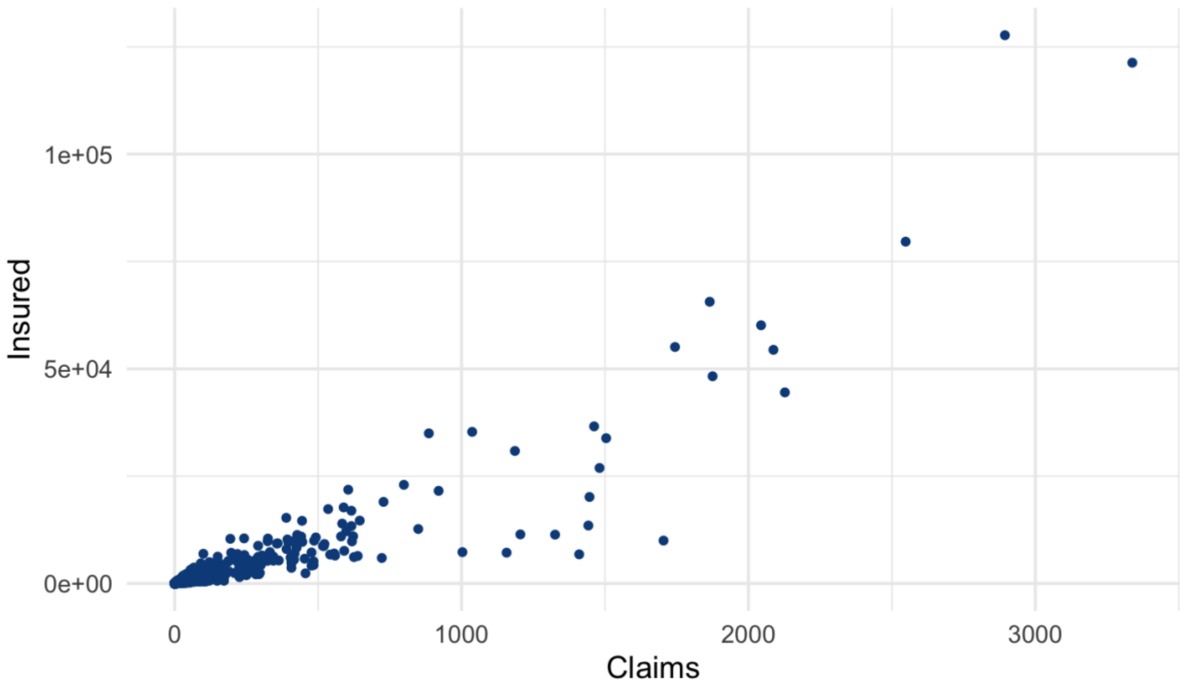
# as claims increase, also there are outliers present.

```
ggplot(ins_data) +
 aes(x = Insured, y = Payment) +
 geom_point(size = 1L, colour = "#0c4c8a") +
 theme_minimal()
```




# The scatter plot shows us that payment increases as
# claims increase, there  are outliers present

```
ggplot(ins_data) +
 aes(x = Claims, y = Insured) +
 geom_point(size = 1L, colour = "#0c4c8a") +
 theme_minimal()
```

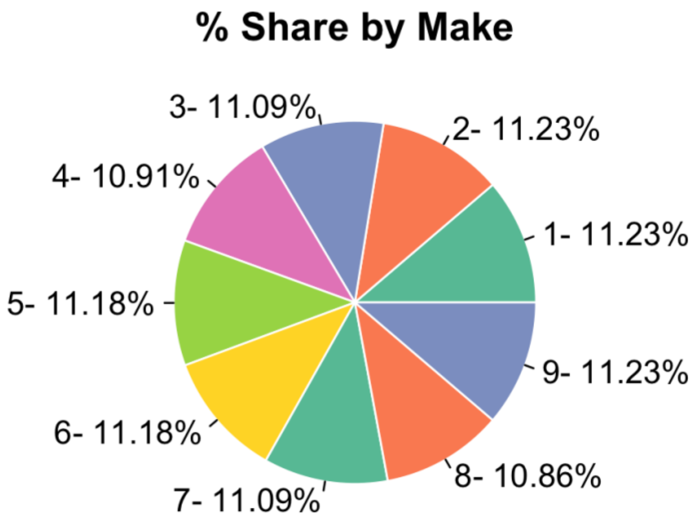



# Claims are also directly related to the Insured field
# as shown in this plot with some outliers

```
# Converting Kilometres, Zone and Make variables to Factor
ins_data$Kilometres <- as.factor(ins_data$Kilometres)
ins_data$Zone <- as.factor(ins_data$Zone)
ins_data$Make <- as.factor(ins_data$Make)

str(ins_data)

'data.frame':	2182 obs. of  7 variables:
 $ Kilometres: Factor w/ 5 levels "1","2","3","4",..: 1 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 1 ...
 $ Bonus     : int  1 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 ...

#### Data cleaning ######
#### All values less than 0 are converted to NA and then dropped ####
```

```
library(dplyr)
library(tidyr)

ins_data <- ins_data %>%
  mutate(Insured = replace(Insured, Insured<0, NA),
         Claims = replace(Claims, Claims<0, NA),
         Payment = replace(Payment, Payment<0, NA))

ins_data <- ins_data %>%
  drop_na()

str(ins_data)
summary(ins_data)

Kilometres Zone       Bonus            Make        Insured
 1:439      1:315   Min.   :1.000   1      :245   Min.   :     0.01
 2:441      2:315   1st Qu.:2.000   2      :245   1st Qu.:    21.61
 3:441      3:315   Median :4.000   9      :245   Median :    81.53
 4:434      4:315   Mean   :4.015   5      :244   Mean   :  1092.20
 5:427      5:313   3rd Qu.:6.000   6      :244   3rd Qu.:   389.78
            6:315   Max.   :7.000   3      :242   Max.   :127687.27
            7:294                   (Other):717
     Claims           Payment
 Min.   :   0.00   Min.   :       0
 1st Qu.:   1.00   1st Qu.:    2989
 Median :   5.00   Median :   27404
 Mean   :  51.87   Mean   :  257008
 3rd Qu.:  21.00   3rd Qu.:  111954
 Max.   :3338.00   Max.   :18245026

#### Data Visualization before Modeling  ####

library(RColorBrewer)

# Visualizing the percentage share of Insurance claims by Make of cars

vc <- table(ins_data$Make)
perc <- round(table(ins_data$Make)/sum(table(ins_data$Make)) * 100, 2)
pie(vc, radius = 1,
    labels = paste(names(vc),'- ',perc, '%', sep = ''),
    main = '% Share by Make',
```

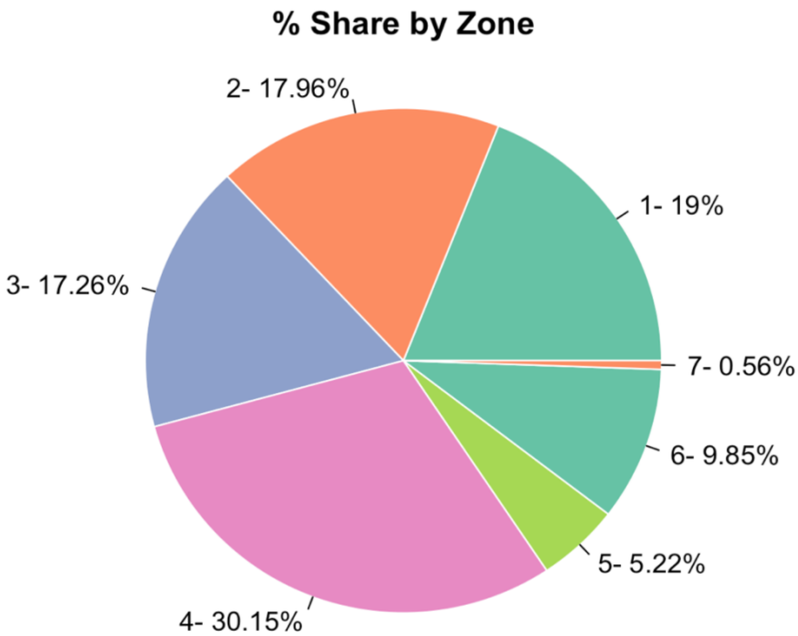
```
col = brewer.pal(6, 'Set2'),
border = 'white')
```




```
# The Make of a car in the data is equally distributed, but to get the
# correct picture we need to aggregate the payment based on the Make
# variable as will be shown below.

###################   Task 3   #################
# The below inference will help the committee to know how the distance,
# zone, make and bonus affect the inurance payment
#### Aggregating data using group  by for Zone, Kilo, Bonus and Make

zo <- ins_data %>%
  group_by(Zone) %>%
  summarise(Payment = mean(Payment), Insured = mean(Insured), Claims =
mean(Claims))%>%
  data.frame()

> zo
  Zone   Payment    Insured    Claims
1    1 338518.95 1036.17175  73.568254
2    2 319921.52 1231.48184  67.625397
3    3 307550.85 1362.95870  63.295238
```

```
4    4 537071.76 2689.38041 101.311111
5    5  93001.84  384.80188  19.047923
6    6 175528.47  802.68457  32.577778
7    7   9948.19   64.91071   2.108844
```

```
vcz <- zo$Payment
percz <- round(zo$Payment/sum(zo$Payment) * 100, 2)
pie(vcz, radius = 1,
    labels = paste(zo$Zone,'- ',percz, '%', sep = ''),
    main = '% Share by Zone',
    col = brewer.pal(5, 'Set2'),
    border = 'white')
```

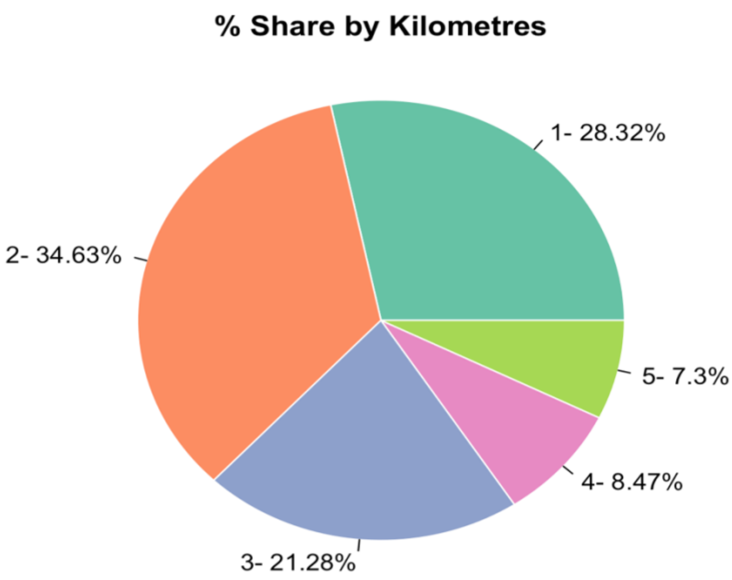



```
# Inference 1 -  Zone 4 - Rural areas in southern Sweden
# has the Highest Payment with 30.15% share

ko <- ins_data %>%
  group_by(Kilometres) %>%
  summarise(Payment = mean(Payment), Insured = mean(Insured), Claims =
mean(Claims))%>%
  data.frame()
```

```
> ko
```

```
  Kilometres  Payment  Insured  Claims
1          1 361899.35 1837.8163 75.59453
2          2 442523.78 1824.0288 89.27664
3          3 272012.58 1081.9714 54.16100
4          4 108213.41  398.9632 20.79493
5          5  93306.12  284.9475 18.04215
```

```
vck <- ko$Payment
perck <- round(ko$Payment/sum(ko$Payment) * 100, 2)
pie(vck, radius = 1,
    labels = paste(ko$Kilometres,'- ',perck, '%', sep = ''),
    main = '% Share by Kilometres',
    col = brewer.pal(5, 'Set2'),
    border = 'white')
```

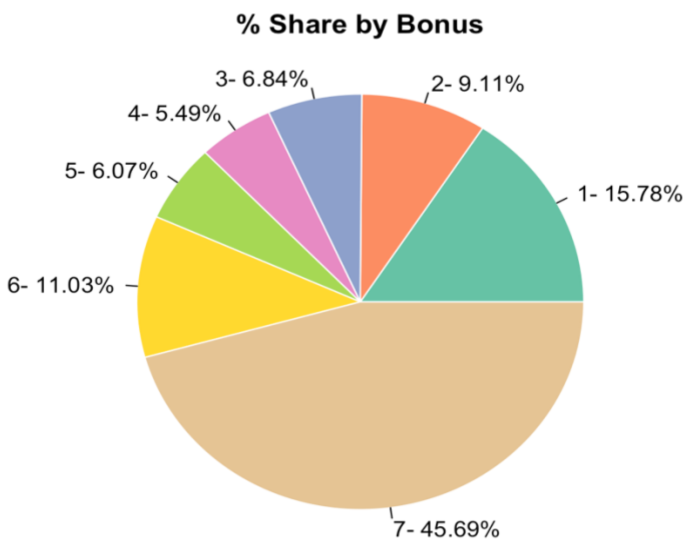



```
# Inference 2 - Kilometres 2 - 1,000 kms to 15,000 kms travelled per year
# has the Highest Payment with 34.63% share

bo <- ins_data %>%
  group_by(Bonus) %>%
  summarise(Payment = mean(Payment), Insured = mean(Insured), Claims =
mean(Claims))%>%
  data.frame()
```

```
> bo
```

```
  Bonus   Payment   Insured    Claims
1     1 282921.99  525.5502  62.50489
2     2 163316.62  451.0754  34.23397
3     3 122656.17  397.4737  24.97419
4     4  98498.12  360.3867  20.35161
5     5 108790.50  437.3936  22.82109
6     6 197723.82  805.8167  39.94286
7     7 819322.48 4620.3728 157.22222
```

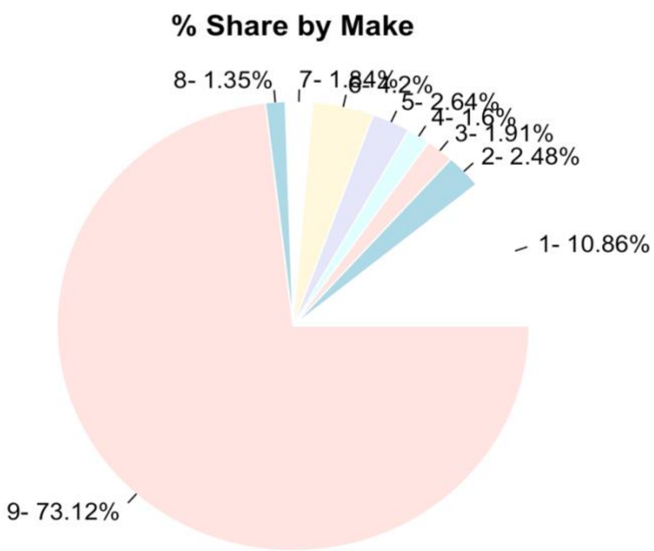
```
vcb <- bo$Payment
percb <- round(bo$Payment/sum(bo$Payment) * 100, 2)
pie(vcb, radius = 1,
    labels = paste(bo$Bonus,'- ',percb, '%', sep = ''),
    main = '% Share by Bonus',
    col = brewer.pal(7, 'Set2'),
    border = 'white')
```




```
# Inference 3- Bonus 4 (3 years no bonus claims)
# has the Least Payment of 5.49% share.

mo <- ins_data %>%
  group_by(Make) %>%
  summarise(Payment = mean(Payment), Insured = mean(Insured), Claims =
mean(Claims))%>%
  data.frame()
```

```
> mo
  Make    Payment  Insured     Claims
1    1  248975.38  977.8498  47.436735
2    2   56760.36  209.1358  11.212245
3    3   43785.92  201.4986   7.632231
4    4   36746.52  279.3529   8.676471
5    5   60519.64  220.1932  12.680328
6    6   96216.83  526.2460  19.114754
7    7   42280.04  202.4762   9.008264
8    8   31053.71  102.8262   4.654008
9    9 1676360.67 7026.9784 342.240816
```

```
vcm <- mo$Payment
percm <- round(mo$Payment/sum(mo$Payment) * 100, 2)
pie(vcm, radius = 1,
    labels = paste(mo$Make,'- ',percm, '%', sep = ''),
    main = '% Share by Make',
    border = 'white')
```




```
# Inference 4 - Make 9 - The group with not common car brands
# has the Highest Payment with 73.12% share

##############  Conclusion from the above inferences #########
```

# 1. Payments tend to be more towards people who
# drive less in a year ( 1,000 kms to 15,000 kms )

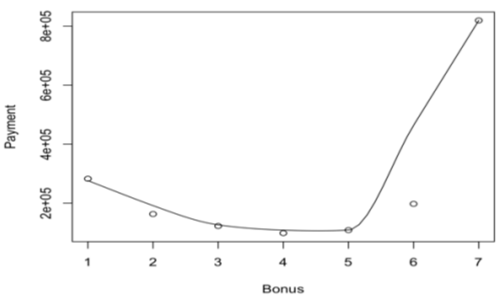
# 2. Payments in the rural areas of southern Sweden
# are comparatively more.

# 3. Bonus for people who have not made claims for last 3 years
# is more these are the people who drive their cars well with the
# least number of claims and the least payment.

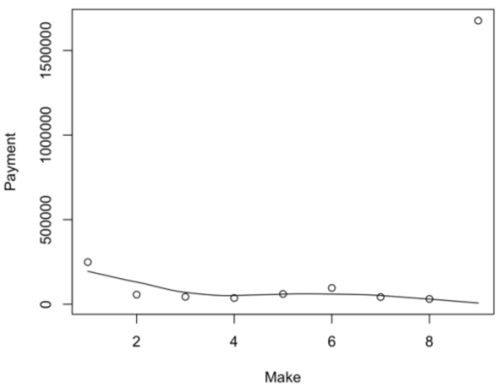
# 4. Payments made to the group with uncommon make of cars and vehicles
# is the highest with 73.12% of the payment.
# More data is sorted to deep dive into this category to make
# wise decisions for the future. one of the decisions would be to
# increase the premium of the car brands with more accidents.

######### Task - 4 ###############
# To open a new branch office the following data and graphs will
# be very benificial

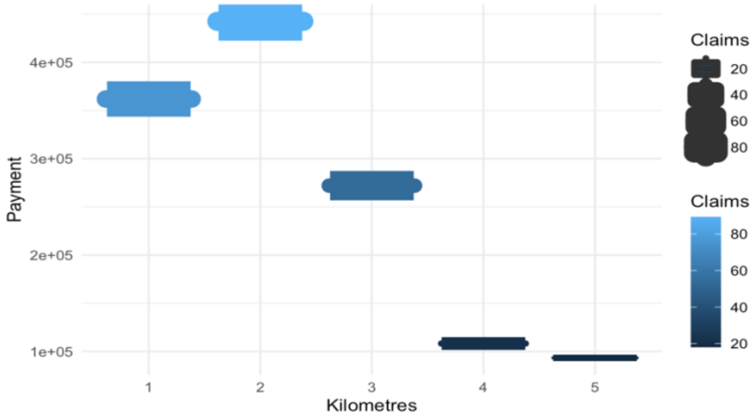
scatter.smooth(bo)



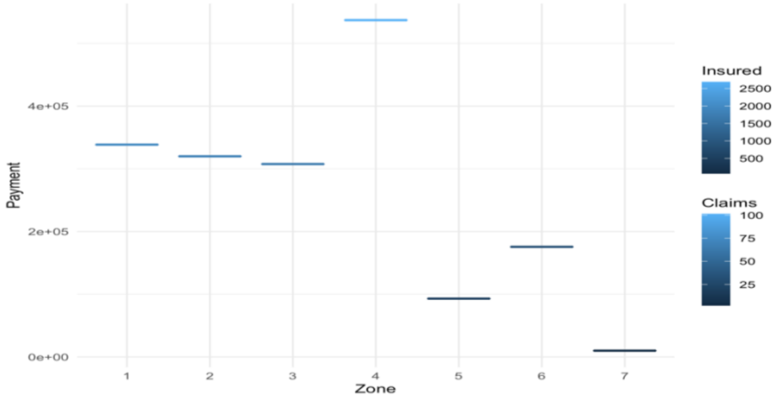

scatter.smooth(mo)



```
ggplot(ko) +
 aes(x = Kilometres, y = Payment, colour = Claims, size = Claims) +
 geom_boxplot(fill = "#0c4c8a") +
 scale_color_gradient() +
 theme_minimal()
```



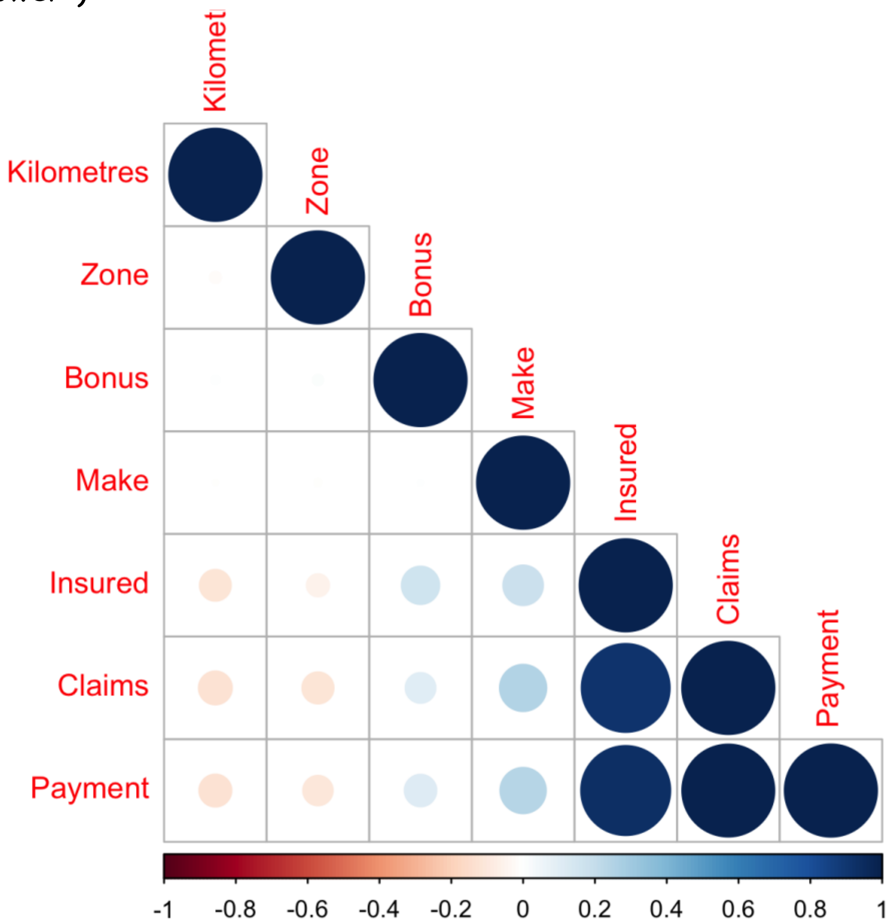

```
ggplot(zo) +
 aes(x = Zone, y = Payment, fill = Insured, colour = Claims) +
 geom_boxplot() +
 scale_fill_gradient() +
 scale_color_gradient() +
 theme_minimal()
```



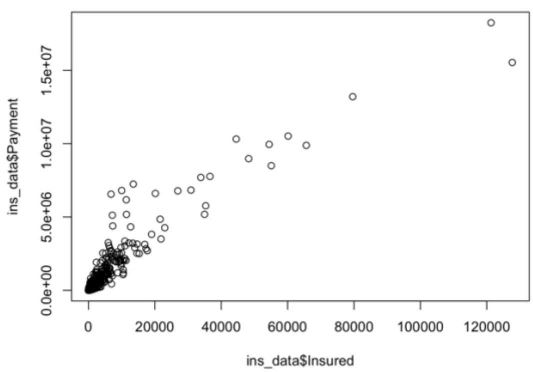

```
###### Conclusion from the above data and graphs ######
# 1. The Payment cost tends to increase exponentially when
# the insurance company registers uncommom brand of cars
# 2. The payment cost increases when the people drive lesser
# kilometers in a year.
# 3. The payments are more in rural southern Sweden
# 4. People with bonuses of 4 or more tens to make more claims.

######              Task-2          #############
###### Model for predicting the Payment ######
###### Building a Linear Regression Model ######
```

# heat map

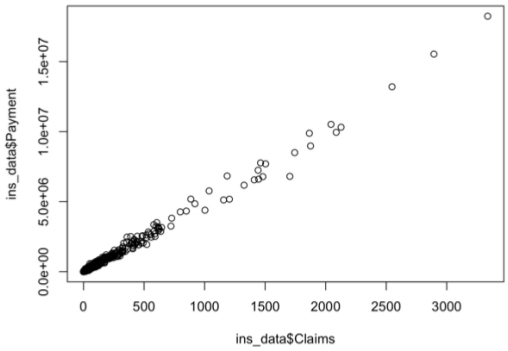
```
library(corrplot)
corrplot(cor(data), method = 'circle' ,
      type = 'lower')
```




```
###### Plotting highly correlated variables
cor(ins_data$Insured, ins_data$Payment)
cor(ins_data$Claims, ins_data$Payment)

plot(ins_data$Insured, ins_data$Payment)
```



```
plot(ins_data$Claims, ins_data$Payment)
```




```
# Splitting the data into train and test ( 70:30 ratio)
# 70 - 30
set.seed(12)
train_ind <- sample(1:nrow(ins_data), 0.70*nrow(ins_data))
train <- ins_data[train_ind , ]
test <- ins_data[ - train_ind , ]

# model development on train data :
fit <- lm(Payment ~., data = train)
summary(fit)
Call:
lm(formula = Payment ~ ., data = train)

Residuals:
    Min      1Q  Median      3Q     Max
-555810  -17058   -1913   15545  645820

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.709e+04  7.756e+03  -2.204 0.027678 *
Kilometres2  1.872e+04  5.077e+03   3.688 0.000234 ***
Kilometres3  2.304e+04  5.024e+03   4.586 4.90e-06 ***
Kilometres4  1.895e+04  5.106e+03   3.712 0.000213 ***
Kilometres5  1.936e+04  5.154e+03   3.756 0.000179 ***
Zone2        5.573e+03  5.963e+03   0.935 0.350125
Zone3        1.148e+04  5.977e+03   1.921 0.054933 .
Zone4        2.780e+04  6.056e+03   4.590 4.80e-06 ***
Zone5        9.408e+03  6.076e+03   1.548 0.121746
Zone6        2.171e+04  5.978e+03   3.632 0.000291 ***
Zone7        6.033e+03  6.097e+03   0.989 0.322599
Bonus        1.875e+03  8.182e+02   2.291 0.022105 *
```

```
Make2       -1.551e+04  6.716e+03  -2.310 0.021038 *
Make3       -1.336e+04  6.824e+03  -1.958 0.050431 .
Make4       -2.761e+04  6.891e+03  -4.007 6.46e-05 ***
Make5       -1.634e+04  6.889e+03  -2.372 0.017827 *
Make6       -1.476e+04  7.004e+03  -2.107 0.035242 *
Make7       -1.973e+04  6.976e+03  -2.829 0.004731 **
Make8       -1.163e+04  6.898e+03  -1.686 0.092037 .
Make9       -1.182e+04  7.602e+03  -1.555 0.120089
Insured      2.364e+01  7.776e-01  30.409  < 2e-16 ***
Claims      4.398e+03  2.418e+01 181.901  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 62500 on 1505 degrees of freedom
Multiple R-squared:  0.9961,	Adjusted R-squared:  0.9961
F-statistic: 1.841e+04 on 21 and 1505 DF,  p-value: < 2.2e-16
# multicollinearity

# i.e. correlation between independent variables
# independent values should be non correlated

# VIF :variance inflation factor

sub<- train %>%
  select(-Zone, -Bonus, -Make)

fit1  <- lm(Payment ~ ., data = sub)
summary(fit1)$r.squared
car::vif(fit1)

cd<-cooks.distance(fit1)

sub<- train %>%
  select(-Zone, -Bonus, -Make)

final_train <- sub[cd<(4/nrow(sub)),]
nrow(final_train)
nrow(sub)

model <- lm(Payment~ ., data = final_train)
summary(model)
```

```
Call:
lm(formula = Payment ~ ., data = final_train)

Residuals:
   Min      1Q  Median      3Q     Max 
-134847  -10508   -3289   10721  146233 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -2328.114   1951.565  -1.193  0.23309    
Kilometres2  4518.119   2695.568   1.676  0.09393 .  
Kilometres3  5397.134   2673.913   2.018  0.04373 *  
Kilometres4  3291.457   2677.280   1.229  0.21912    
Kilometres5  7433.480   2708.828   2.744  0.00614 ** 
Insured        21.807      1.622  13.448  < 2e-16 ***
Claims       4423.894     36.946 119.740  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 31940 on 1450 degrees of freedom
Multiple R-squared:  0.988,	Adjusted R-squared:  0.9879 
F-statistic: 1.986e+04 on 6 and 1450 DF,  p-value: < 2.2e-16

# Test the moodel with the test data

sub_test <- test%>%
  select(-Zone, -Bonus, -Make)

predict <- predict(model,sub_test)

DMwR :: regr.eval(sub_test$Payment, predict)

         mae          mse         rmse         mape 
3.539491e+04 8.299516e+09 9.110168e+04          Inf 

# We now have a Linear Model that can predict the value of Payment
# with very low MAE, MSE and RMSE

######          Task-5        #################
###### Model for predicting the Claims #########
###### Building a Linear Regression Model ######
```

```
# Splitting the data into train and test ( 70:30 ratio )
# 70 - 30
set.seed(21)
train_ind <- sample(1:nrow(ins_data), 0.70*nrow(ins_data))
train <- ins_data[train_ind , ]
test <- ins_data[ - train_ind , ]

# model development on train data :
fit <- lm(Claims ~., data = train)
summary(fit)
Call:
lm(formula = Claims ~ ., data = train)

Residuals:
     Min       1Q   Median       3Q      Max
-167.100   -3.757    0.198    4.212  145.815

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  4.496e+00  1.928e+00   2.332 0.019831 *
Kilometres2 -2.171e+00  1.256e+00  -1.728 0.084184 .
Kilometres3 -4.521e+00  1.258e+00  -3.593 0.000338 ***
Kilometres4 -4.382e+00  1.257e+00  -3.486 0.000505 ***
Kilometres5 -4.554e+00  1.283e+00  -3.550 0.000397 ***
Zone2        1.063e+00  1.495e+00   0.711 0.477356
Zone3       -8.557e-01  1.505e+00  -0.568 0.569838
Zone4       -6.821e+00  1.528e+00  -4.464 8.64e-06 ***
Zone5       -2.448e+00  1.514e+00  -1.617 0.105993
Zone6       -5.337e+00  1.508e+00  -3.539 0.000413 ***
Zone7       -2.340e+00  1.561e+00  -1.499 0.133964
Bonus       -6.368e-01  2.057e-01  -3.095 0.002001 **
Make2        3.021e+00  1.683e+00   1.795 0.072786 .
Make3        2.306e+00  1.665e+00   1.385 0.166132
Make4        5.217e+00  1.701e+00   3.066 0.002205 **
Make5        3.748e+00  1.691e+00   2.217 0.026771 *
Make6        4.119e+00  1.679e+00   2.454 0.014255 *
Make7        4.147e+00  1.697e+00   2.444 0.014633 *
Make8        1.958e+00  1.702e+00   1.150 0.250132
Make9        9.741e+00  1.854e+00   5.255 1.70e-07 ***
Insured     -4.168e-03  2.114e-04 -19.712  < 2e-16 ***
```

```
Payment      2.170e-04  1.321e-06 164.308  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 15.59 on 1505 degrees of freedom
Multiple R-squared:  0.9949,	Adjusted R-squared:  0.9948
F-statistic: 1.388e+04 on 21 and 1505 DF,  p-value: < 2.2e-16
```

```
#All features are equally important and contributing  to the model


predict <- predict(fit,test)

DMwR :: regr.eval(test$Claims, predict)
```

```
       mae        mse       rmse       mape
  7.383277 282.803972  16.816777        Inf
```

```
# Similarly we now have a linear model that can predict
# the Claims for us
```