LAPTOP EXPERT SYSTEM

Name: Sai Sravanth Segu UTA ID:1002125503 Dataset:https://www.kaggle.com/datasets/nallaakash/laptops-cleaned-dataset/data Where it is coming from:Kaggle Info about the dataset: The dataset contains data from laptops from various companies with different features such as CPU_speed, RAM, memory_type, primary_storage, OpSys, and price, which are important factors in determining the best laptop. The results of the eda from assignment 2 show that the majority of laptops come with priced configurations that include 8/16GB RAM, a CPU speed, and Windows OS.Let's do the regression analysis again with the same data set, but this time we'll predict the best laptop based on its features.

```
df <- read.csv("laptops.csv")
str(df)
```

```
## 'data.frame':    548 obs. of  8 variables:
##  $ index          : int  0 1 2 3 4 5 6 7 8 9 ...
##  $ Company        : chr  "Apple" "Apple" "HP" "Apple" ...
##  $ cpu_speed      : num  2.3 1.8 2.5 2.7 3.1 3 2.2 1.8 1.8 1.6 ...
##  $ Ram            : int  8 8 8 16 8 4 16 8 16 8 ...
##  $ memory_type    : chr  "SSD" "Flash Storage" "SSD" "SSD" ...
##  $ primary_storage: int  128 128 256 512 256 500 256 256 512 256 ...
##  $ OpSys          : chr  "macos" "macos" "N/A" "macos" ...
##  $ Price          : int  71379 47896 30636 135195 96096 21312 114018 61736 79654 41026 ...
```

```
head(df)
```

```
##   index Company cpu_speed Ram   memory_type primary_storage   OpSys  Price
## 1     0   Apple       2.3   8           SSD             128   macos  71379
## 2     1   Apple       1.8   8 Flash Storage             128   macos  47896
## 3     2      HP       2.5   8           SSD             256     N/A  30636
## 4     3   Apple       2.7  16           SSD             512   macos 135195
## 5     4   Apple       3.1   8           SSD             256   macos  96096
## 6     5    Acer       3.0   4           HDD             500 windows  21312
```

```
summary(df)
```

```
##      index         Company            cpu_speed          Ram        
##  Min.   :  0.0   Length:548         Min.   :1.100   Min.   : 2.000  
##  1st Qu.:141.8   Class :character   1st Qu.:1.800   1st Qu.: 4.000  
##  Median :283.5   Mode  :character   Median :2.500   Median : 8.000  
##  Mean   :282.2                      Mean   :2.242   Mean   : 8.511  
##  3rd Qu.:422.2                      3rd Qu.:2.700   3rd Qu.: 8.000  
##  Max.   :561.0                      Max.   :3.600   Max.   :64.000
```

```
##  memory_type        primary_storage    OpSys               Price
##  Length:548         Min.   :  16    Length:548         Min.   : 10603
##  Class :character   1st Qu.: 256    Class :character   1st Qu.: 30716
##  Mode  :character   Median : 256    Mode  :character   Median : 47899
##                     Mean   : 454                       Mean   : 55894
##                     3rd Qu.: 512                       3rd Qu.: 71395
##                     Max.   :2048                       Max.   :324955
```

```
sum(is.na(df))
```

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

```
colnames(df)
```

```
## [1] "index"           "Company"         "cpu_speed"       "Ram"
## [5] "memory_type"     "primary_storage" "OpSys"           "Price"
```

```
datatypes <- sapply(df, typeof)
print(datatypes)
```

```
##           index         Company       cpu_speed             Ram     memory_type
##       "integer"     "character"        "double"       "integer"     "character"
## primary_storage           OpSys           Price
##       "integer"     "character"       "integer"
```

```
library(psych)
describe(df)
```

```
##                 vars   n     mean       sd  median  trimmed      mad     min
## index              1 548   282.16   162.48   283.5   282.45   208.31     0.0
## Company*           2 548     6.11     3.29     5.0     5.98     2.97     1.0
## cpu_speed          3 548     2.24     0.54     2.5     2.28     0.44     1.1
## Ram                4 548     8.51     5.14     8.0     7.83     0.00     2.0
## memory_type*       5 548     4.16     0.98     5.0     4.26     0.00     1.0
## primary_storage    6 548   454.04   381.81   256.0   408.25   189.77    16.0
## OpSys*             7 548     4.55     1.10     5.0     4.86     0.00     1.0
## Price              8 548 55893.64 34523.66 47899.0 51401.30 29305.81 10603.0
##                    max    range  skew kurtosis      se
## index              561.0    561.0 -0.01    -1.21    6.94
## Company*            17.0     16.0  0.54     0.36    0.14
## cpu_speed            3.6      2.5 -0.45    -0.95    0.02
## Ram                 64.0     62.0  3.57    27.10    0.22
## memory_type*         5.0      4.0 -0.66    -0.95    0.04
## primary_storage   2048.0   2032.0  1.65     3.14   16.31
## OpSys*               5.0      4.0 -2.34     4.11    0.05
## Price           324955.0 314352.0  1.82     7.14 1474.78
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.3.2
```

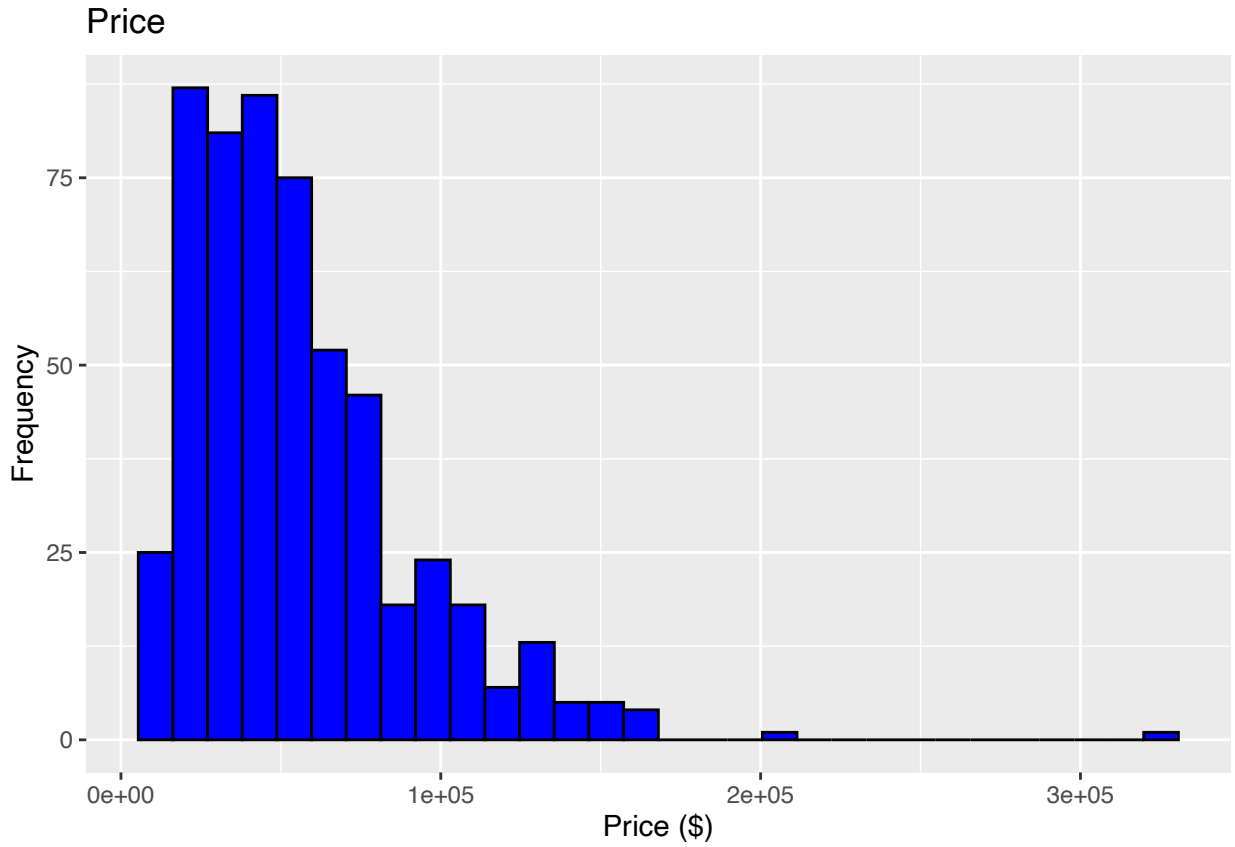
```
##
## Attaching package: 'ggplot2'
```

```
## The following objects are masked from 'package:psych':
##
##     %+%, alpha
```

```
#Correlation among the columns
cor(df[,c("Price", "cpu_speed", "Ram", "primary_storage")])
```

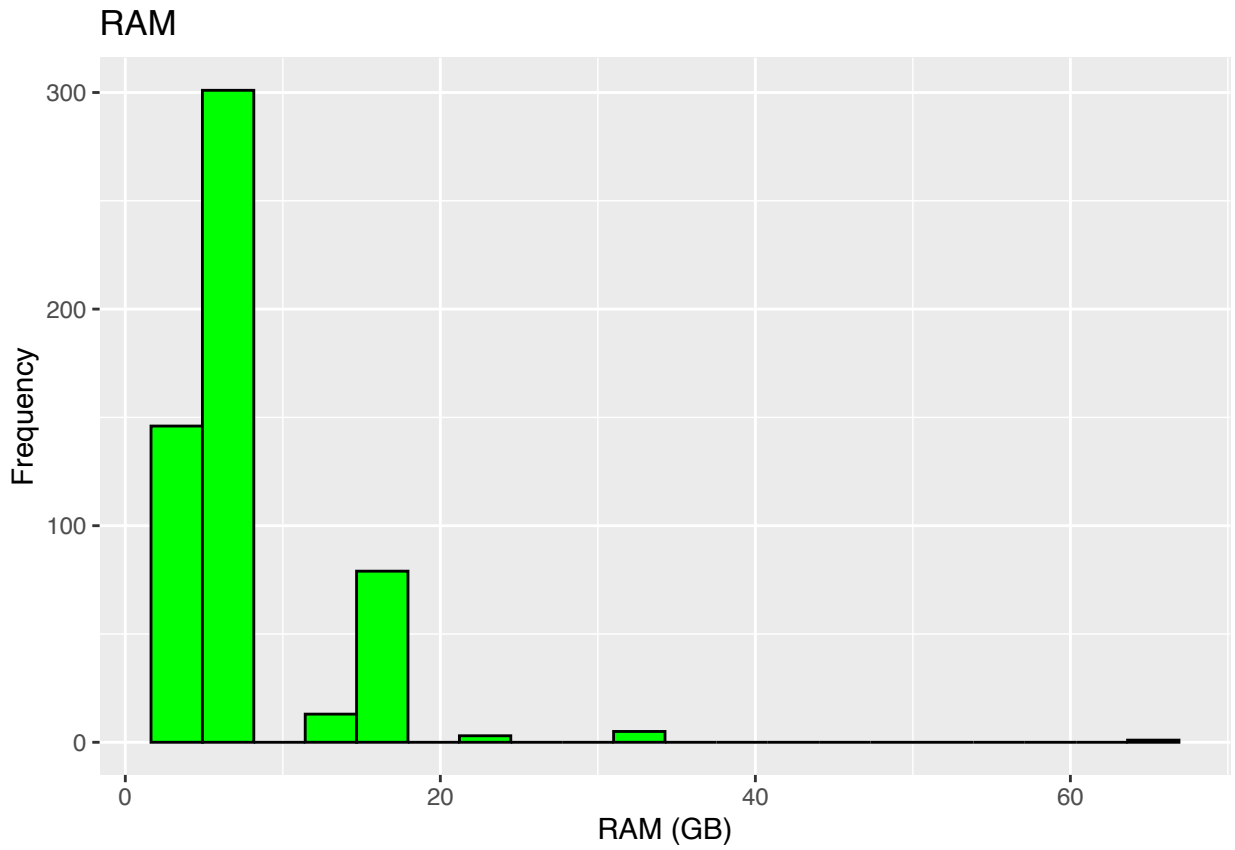
```
##                      Price  cpu_speed          Ram primary_storage
## Price            1.0000000 0.40198852  0.689834784    -0.105681838
## cpu_speed        0.4019885 1.00000000  0.295265753     0.052080206
## Ram              0.6898348 0.29526575  1.000000000    -0.004124682
## primary_storage -0.1056818 0.05208021 -0.004124682     1.000000000
```

```
#Histogram of Price
ggplot(df, aes(Price)) +
geom_histogram(bins = 30, color="black", fill="blue") +labs(title="Price", x="Price ($)", y = "Frequency")
```



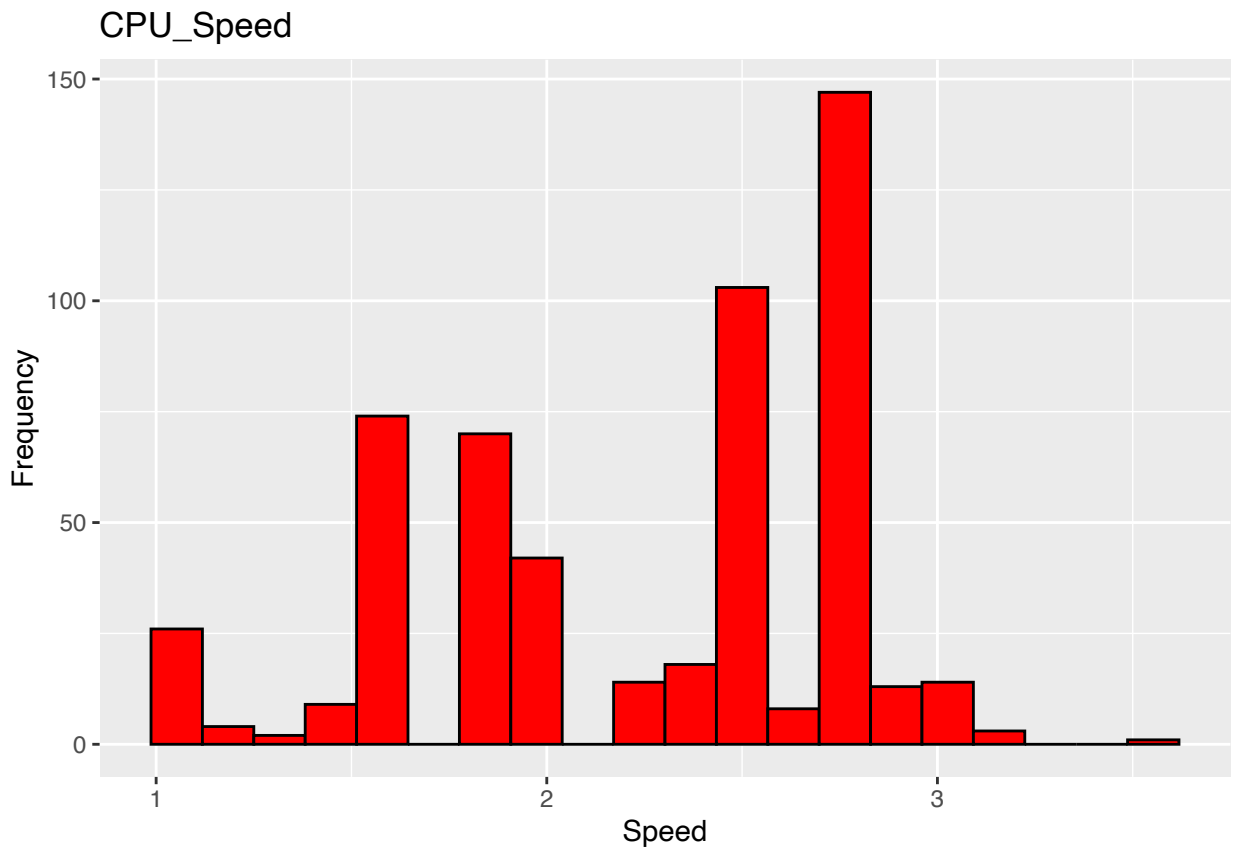
The distribution is right skewed and most laptop prices lie between $10K to $100K.There is a long tail with few high price outliers > $300K

```r
 #Histogram of RAM
ggplot(df, aes(Ram)) +
geom_histogram(bins=20, color="black", fill="green") +labs(title="RAM", x="RAM (GB)", y="Frequency")
```



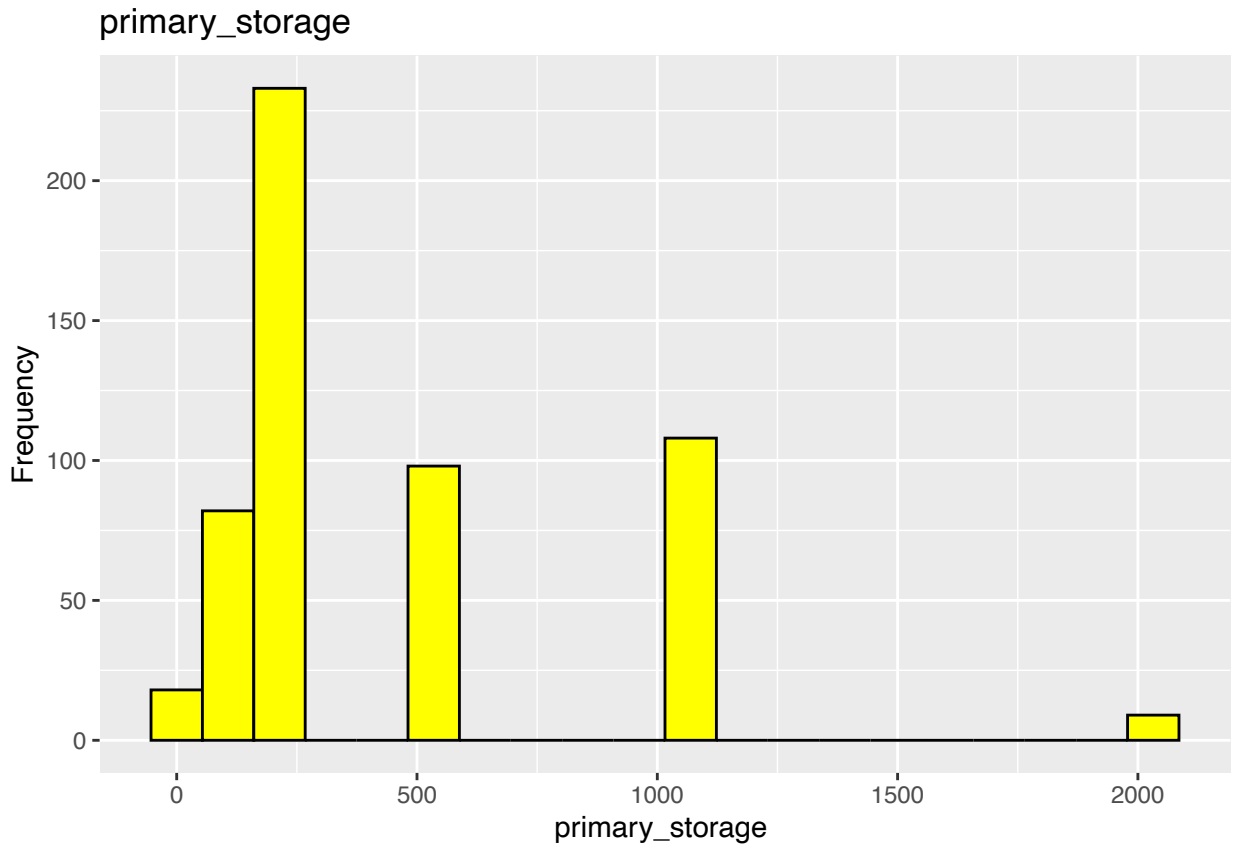
Most laptop RAM configurations range from 4GB to 16GB and highest peak is at 8GB suggesting many laptops have that RAM size Frequencies drop significantly after 16GB with very few models having >32GB RAM

```r
# Histogram of CPU speed
ggplot(df, aes(cpu_speed)) +
geom_histogram(bins=20, color="black", fill="red") +labs(title="CPU_Speed", x="Speed", y="Frequency")
```

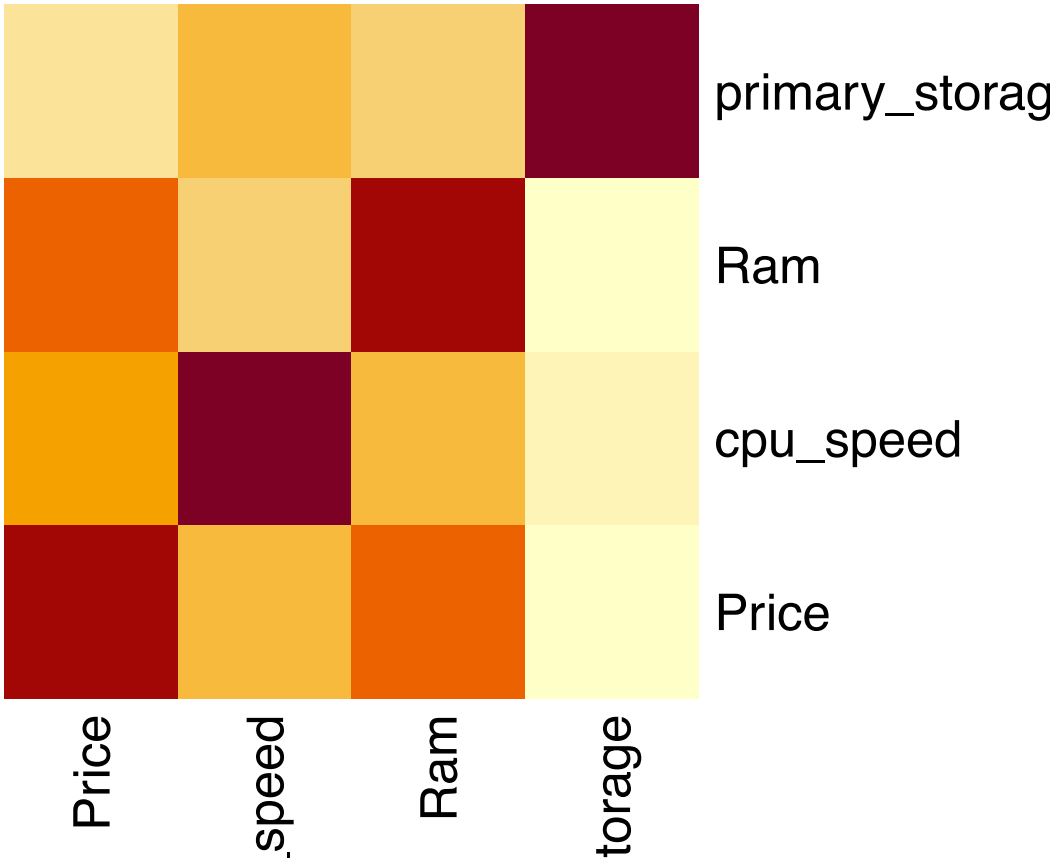


Most laptop CPUs range from 1.6 GHz to 2.8 GHz and there is a peak around 2.5 GHz showing many laptop models with that CPU speed Frequency tapers down towards the higher >3GHz speeds

```
 #Histogram of primary_storage
ggplot(df, aes(primary_storage)) +
geom_histogram(bins=20, color="black", fill="yellow") +labs(title="primary_storage", x="primary_storage
```

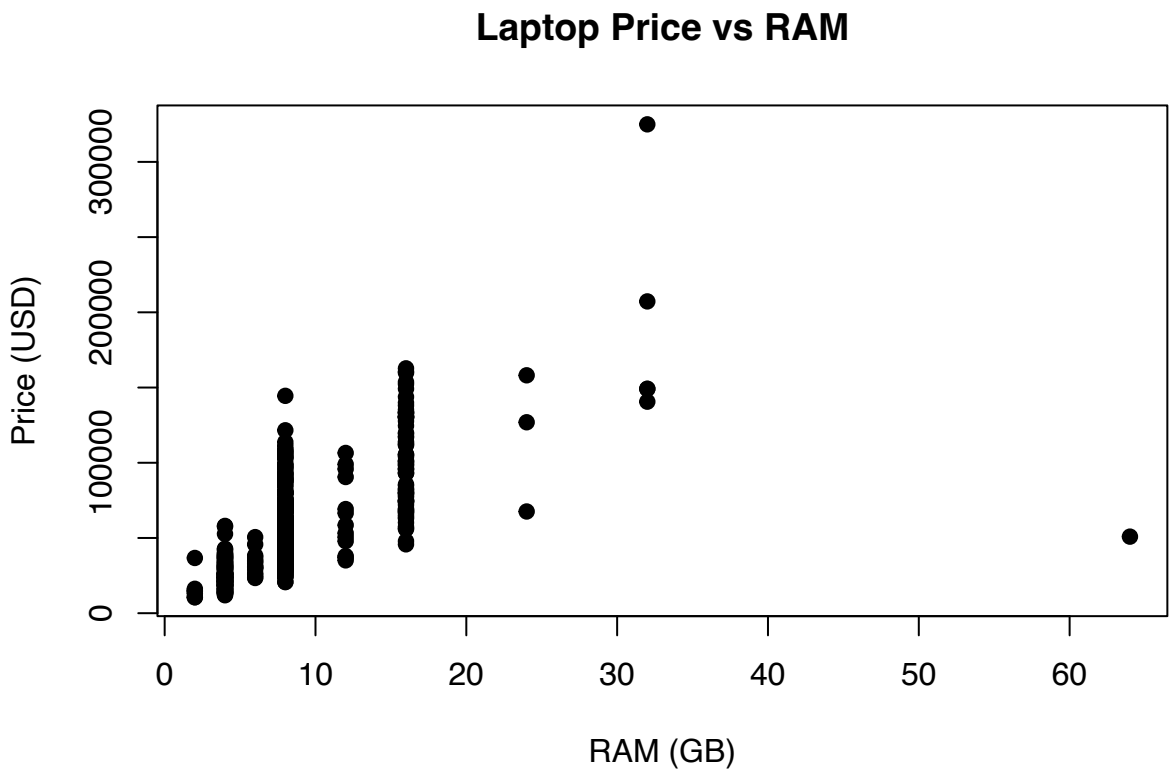



```
# Subset the desired columns
df_subset <- df[, c("Price", "cpu_speed", "Ram", "primary_storage")]

# Create correlation matrix
cor_matrix <- cor(df_subset)

# Plot heatmap without reordering
heatmap(cor_matrix,
        Rowv = NA,
        Colv = NA)
```

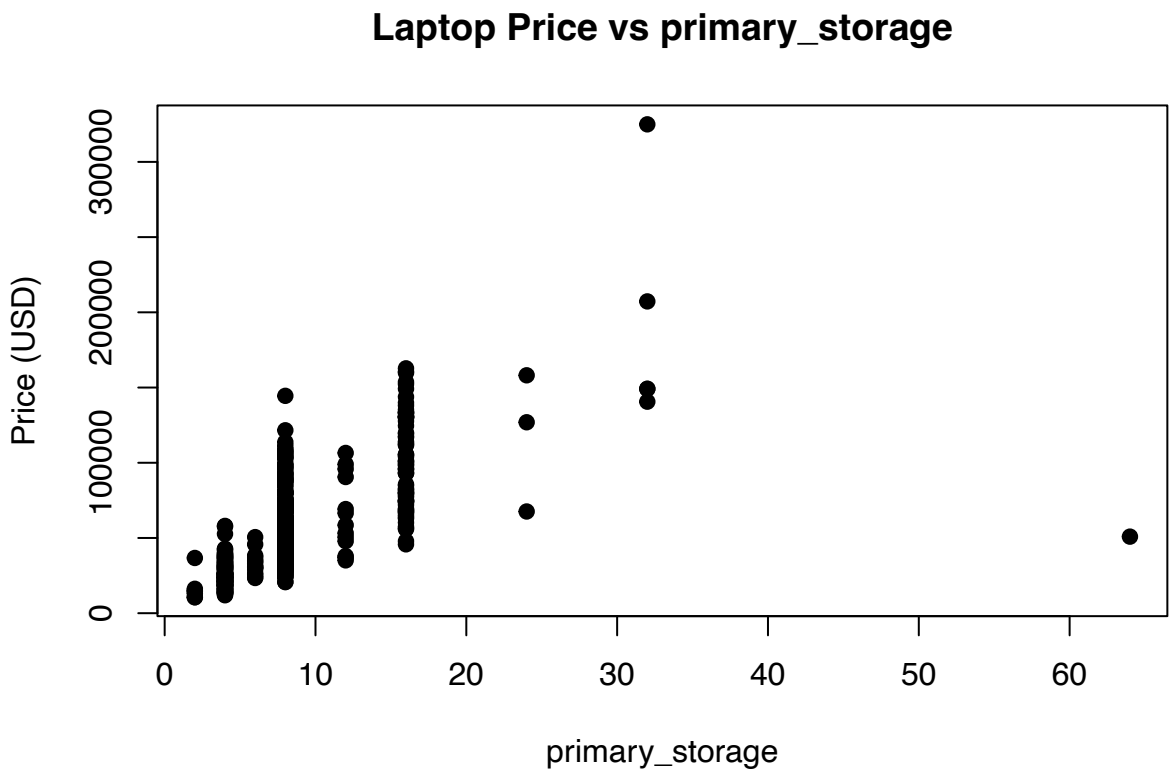


There seems to be a strong positive correlation between Price, CPU speed, RAM capacity, and Storage capacity/type. The high shades of blue indicate coefficients ~0.6-0.8.So laptops with higher priced models seem to also have faster processors, greater RAM, and more advanced storage (SSDs). These attributes tend to move together.

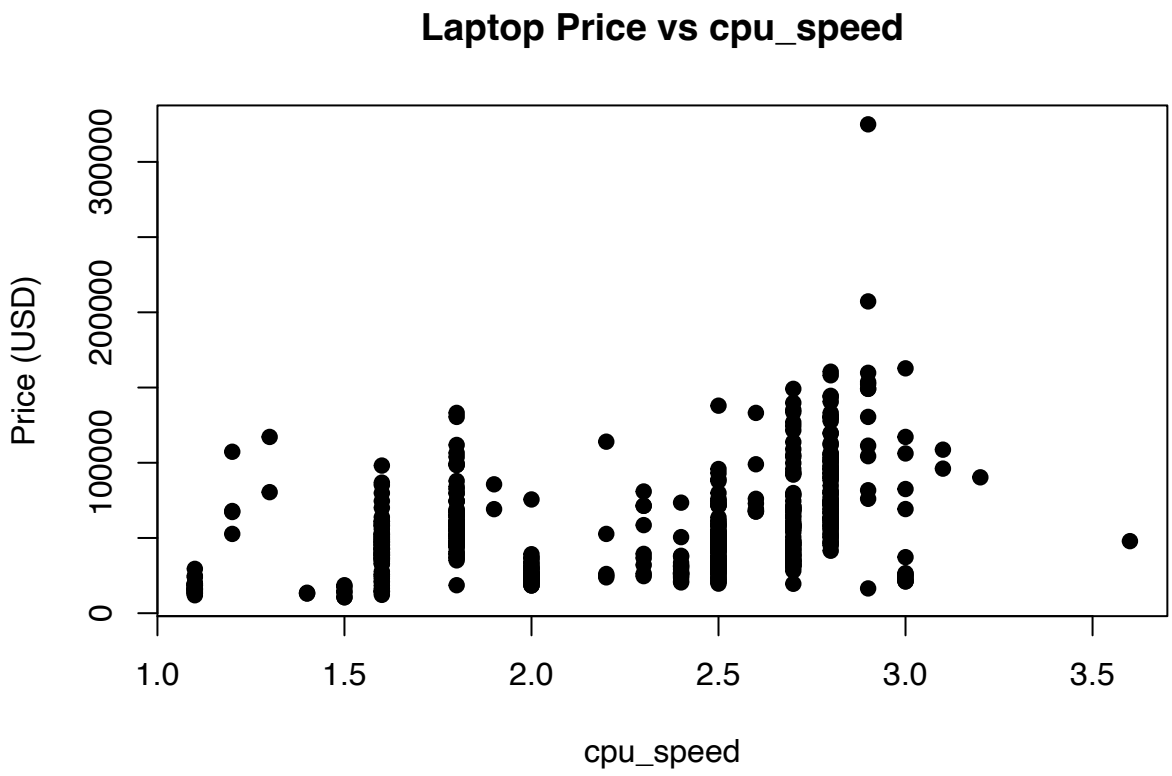
```
plot(df$Ram, df$Price,
     main="Laptop Price vs RAM",
     xlab="RAM (GB)",
     ylab="Price (USD)",
     pch=19)
```




```r
plot(df$Ram, df$Price,
     main="Laptop Price vs primary_storage",
     xlab="primary_storage",
     ylab="Price (USD)",
     pch=19)
```

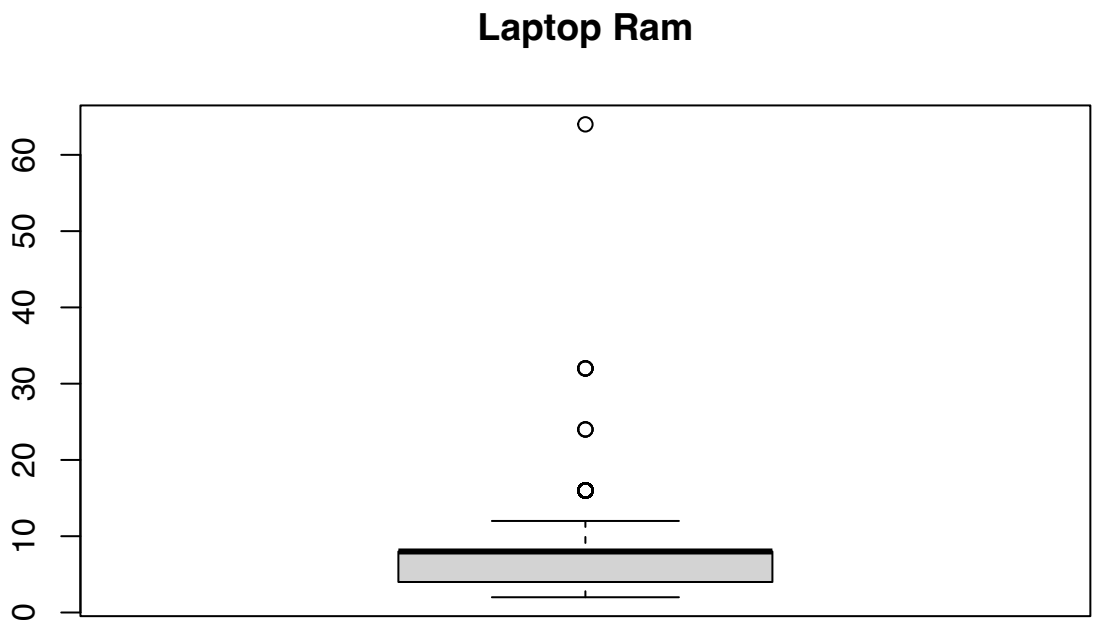



```
plot(df$cpu_speed, df$Price,
     main="Laptop Price vs cpu_speed",
     xlab="cpu_speed",
     ylab="Price (USD)",
     pch=19)
```



We can see from the scatter plots above that all of the features have a linear relationship ship where ram,ROM,battery capacity,number of rear cameras, and display size have a positive relationship but number of front cameras has a negative relationship.

```
boxplot(df$Ram, main="Laptop Ram")
```



So there are very few laptops with exceptionally high RAM capacity. These can be treated as outliers in analysis. Most laptops have modest 2-16GB RAM.

```
model <- lm(Price ~ cpu_speed +Ram +primary_storage,data = df )

summary(model)
```

```
##
## Call:
## lm(formula = Price ~ cpu_speed + Ram + primary_storage, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -233060  -13564   -3819   10222  167111
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -7386.990   4490.141  -1.645 0.100515
## cpu_speed       14432.511   1981.387   7.284 1.14e-12 ***
## Ram              4187.616    206.298  20.299  < 2e-16 ***
## primary_storage   -10.378      2.656  -3.907 0.000105 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 23680 on 544 degrees of freedom
## Multiple R-squared:  0.5321, Adjusted R-squared:  0.5295
```

```
## F-statistic: 206.2 on 3 and 544 DF,  p-value: < 2.2e-16
```

Here in the equation y is price and the x are the features. The intercept here which is -7386.990 represents the predicted value when all predictor variables are zero, which means the price will be -7386.990 when the features of the laptop are zero. The slopes of the features will be the coefficients for the respective features.The slope of a linear regression model is determined by the units of measurement used for the variables involved. The slope is the change in the dependent variable (y) caused by a one-unit increase in the independent variable (x). It represents the rate of change or the sensitivity of y to x. For example, the ram coefficient is 4187.616, which means that for every 1gb increase in ram, the price will increase by 4187.616, and similarly for other features. The features have p value less than 0.05 which means they statistically significant in predicting the price. For hypothesis testing, we have our null hypothesis which states that the responding predictor variable has no effect on the response variable. In other words, the null hypothesis asserts that the coefficients are zero.For each coefficient, the alternative hypothesis is that the corresponding predictor variable has a non-zero effect on the response variable. The alternative hypothesis states that the coefficients are not equal to zero. So, if we look at the p values, we can reject the null hypothesis and conclude that the coefficients are not zero. As a result, we can conclude that the features are important in predicting the response variable, which is price.

The coefficient of determination which is R- squared value we got is 0.5321 which means 5.21% variation of price(y) is explained by the features(x). we got the f statistic value : 206.2 and the corresponding p value is <0.05 and hence we can say that the model is significant.

Linear Equation

```
model <- lm(Price ~ cpu_speed + Ram + primary_storage, data = df)

# Get model coefficients
coefficients <- coef(model)

intercept <- coef(model)[1]
cpu_speed <- coef(model)[2]
Ram <- coef(model)[3]
primary_storage <- coef(model)[4]



# Simplify
linear_eqn <- paste0("Predicted Price = ",
                     format(coefficients[1]), " + ",
                     format(coefficients[2]), "* cpu_speed + ",
                     format(coefficients[3]), "* Ram + ",
                     format(coefficients[4]), "* primary_storage")

print(linear_eqn)
```

```
## [1] "Predicted Price = -7386.99 + 14432.51* cpu_speed + 4187.616* Ram + -10.37792* primary_storage"
```

```
model <- lm(Price ~ cpu_speed + Ram + primary_storage, data = df)

coefficients <- coef(model)

eq <- paste("Predicted Price =", round(coefficients[1],2), "+",
            round(coefficients[2],5), "* CPU Speed +",
            round(coefficients[3],5), "* RAM +",
```

```
            round(coefficients[4], 5), "* Primary Storage")

print(eq)
```

```
## [1] "Predicted Price = -7386.99 + 14432.51142 * CPU Speed + 4187.61572 * RAM + -10.37792 * Primary S
```

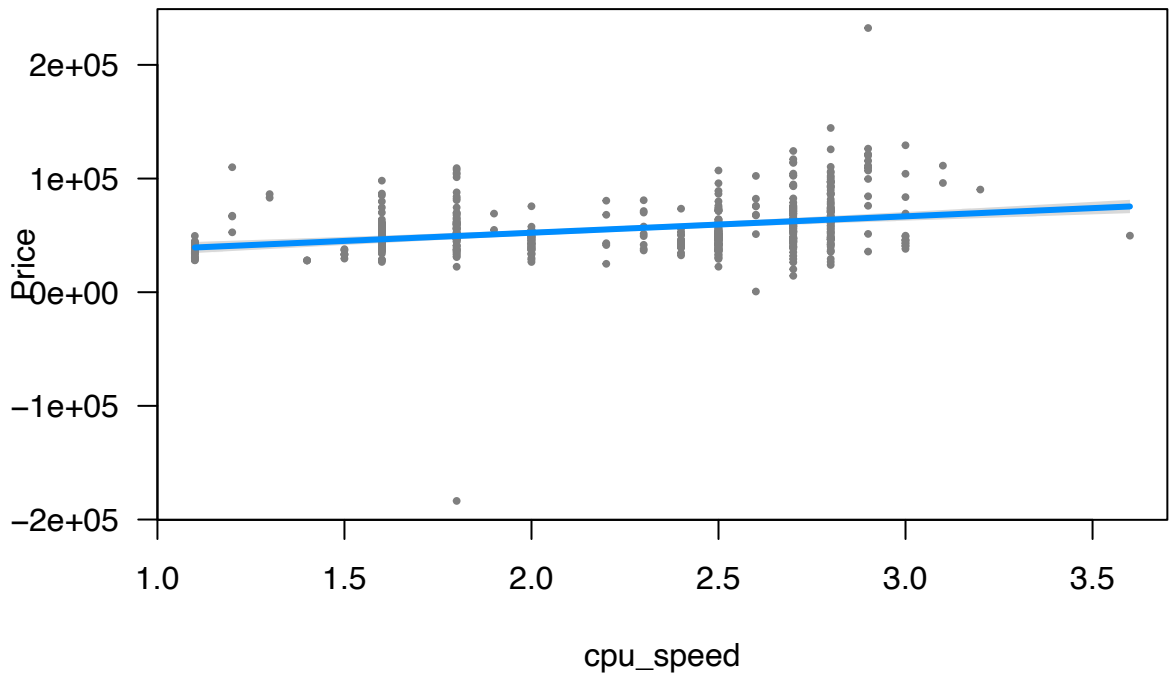
```
model.variance <- var(model$residuals)
sprintf("The variance of the model is : %f", model.variance)
```

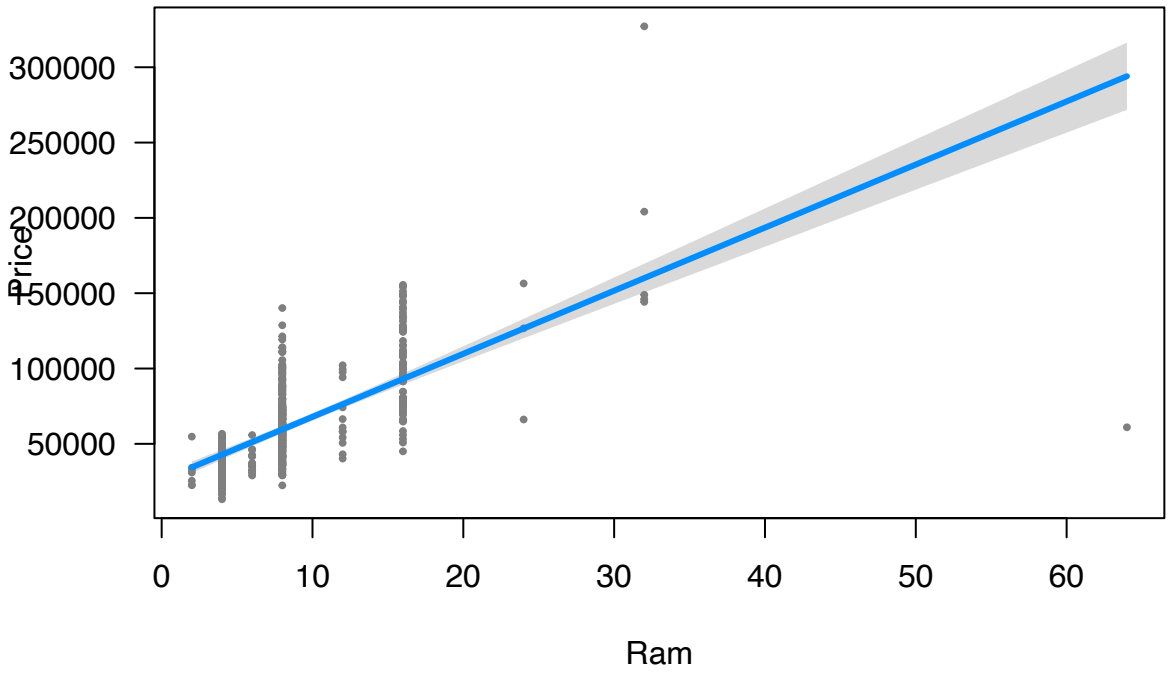
```
## [1] "The variance of the model is : 557700956.850986"
```

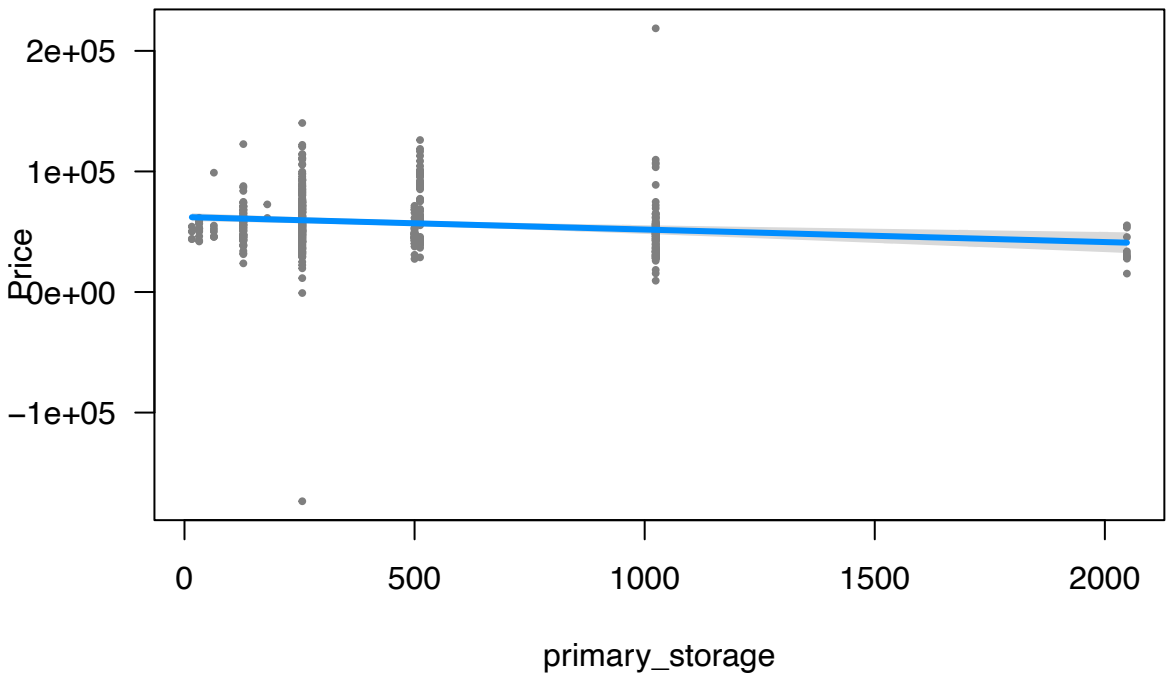
```
library(visreg)
```

```
## Warning: package 'visreg' was built under R version 4.3.2
```

```
visreg(model)
```




300000
250000
200000
150000
100000
50000
Price
0
10
20
30
40
50
60
Ram




```
library(car)
```

```
## Warning: package 'car' was built under R version 4.3.2
```

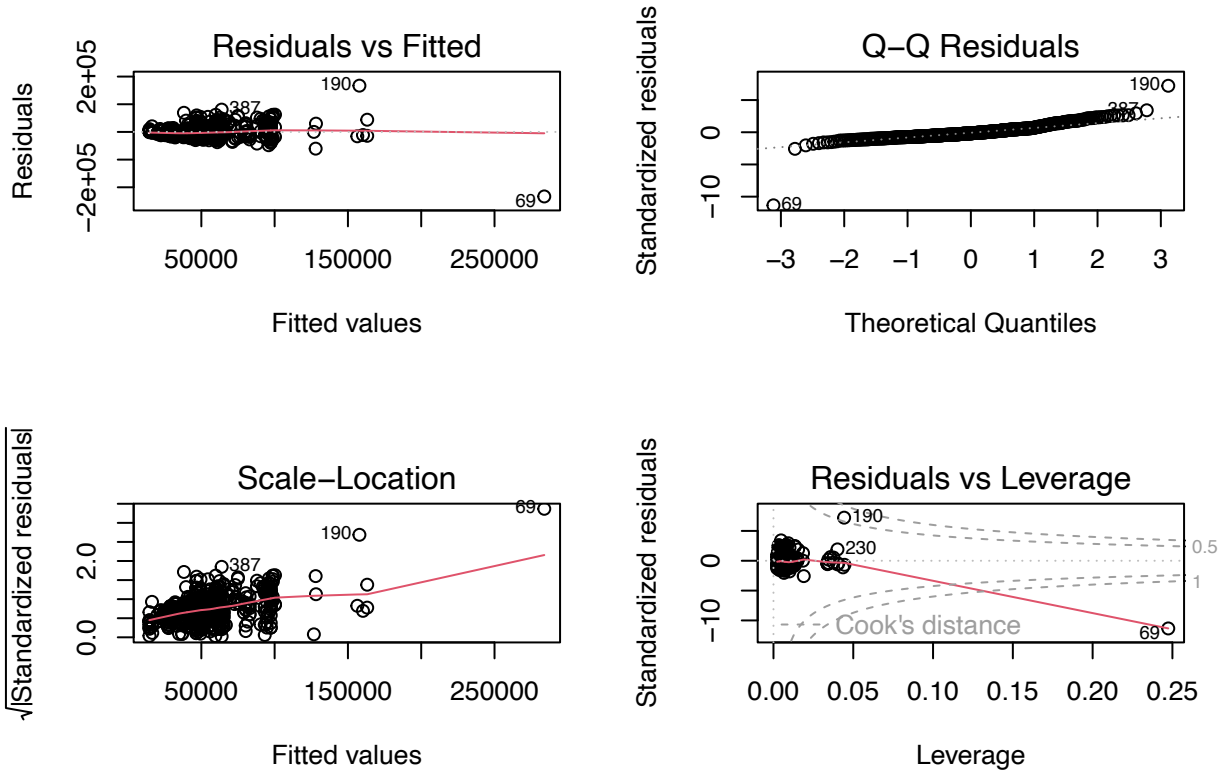
```
## Loading required package: carData
```

```
## Warning: package 'carData' was built under R version 4.3.2
```

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

```
## The following object is masked from 'package:psych':
##
##     logit
```

```
par(mfrow=c(2,2))
plot(model)
```



model assumptions: 1.linearity :The residual Vs Fitted plot shows that the line is horizontal, indicating that the linearity assumption holds.

2.independence :Durbin-Watson statistic is so close to 2, we fail to reject the null hypothesis of no autocorrelation of errors and conclude the crucial regression assumption of independent errors holds for the model.

3.normality :We can conclude from the Q_Q Residuals that points lie on the line, indicating that the assumption holds

4.homoscedasticity :We can tell from the scale-Location graph that the model's assumption holds.

```
vif(model)
```

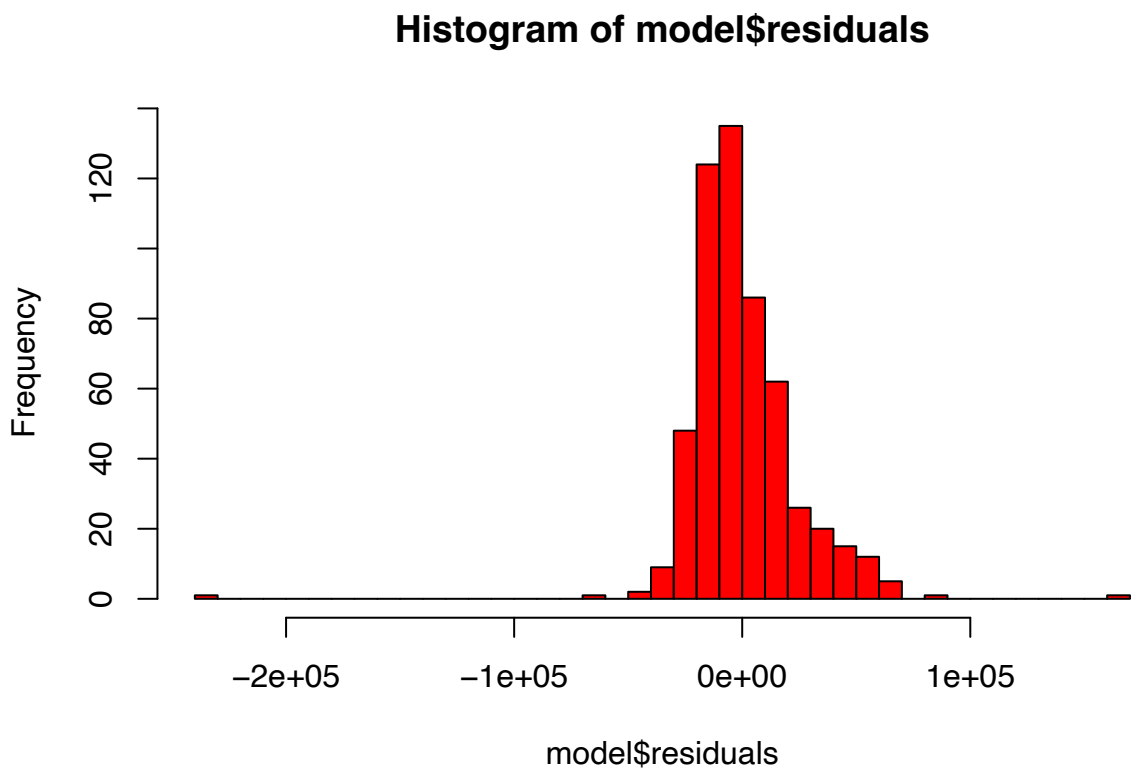
```
##       cpu_speed             Ram primary_storage
##        1.098928        1.095966        1.003139
```

The GVIF values in the above table are all less than 5, indicating that there is less multi collinearity between the features.

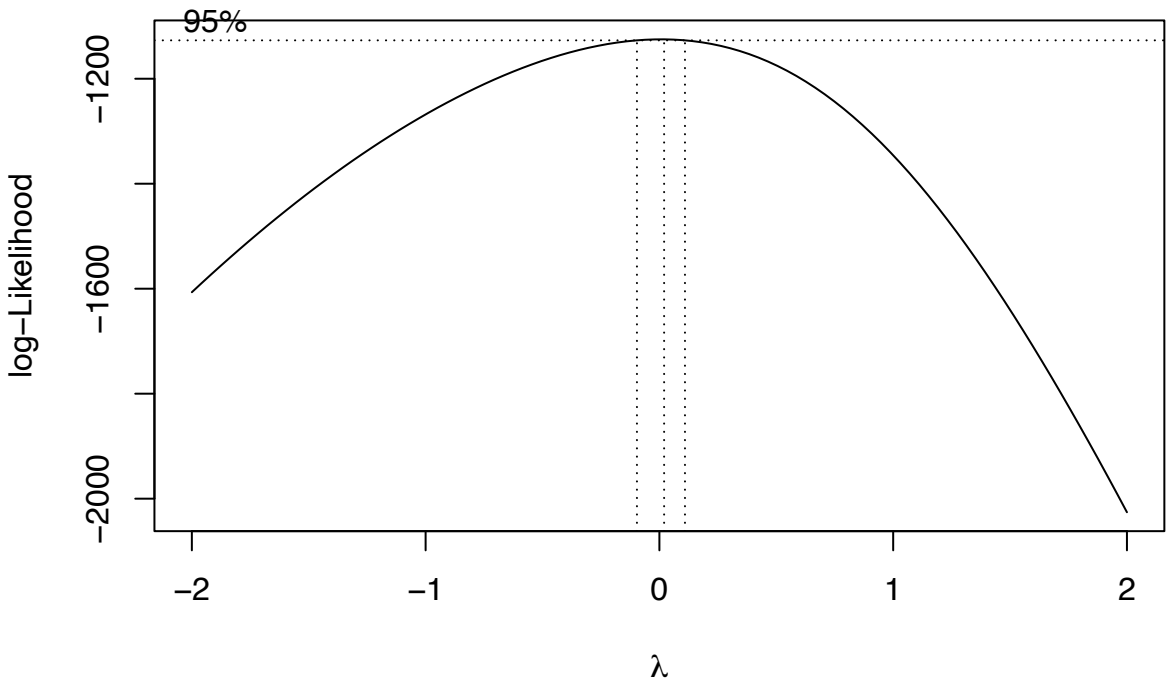
```
durbinWatsonTest(model)
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1     -0.00438831      2.007596   0.902
##  Alternative hypothesis: rho != 0
```

```r
hist(model$residuals, col="red", breaks=39)
```




```r
library(MASS)
bc <- boxcox(Ram~ Price, data=df)
```



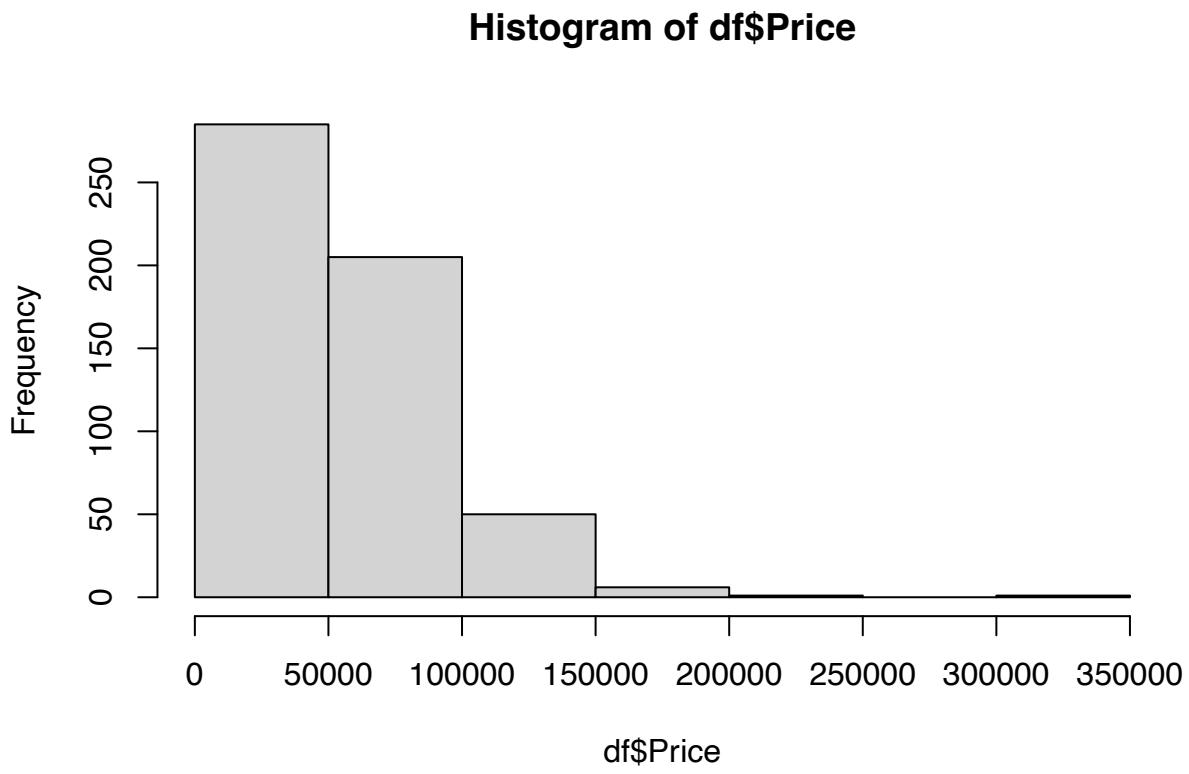

```
lambda <- bc$x[which.max(bc$y)]
```

```
lambda
```

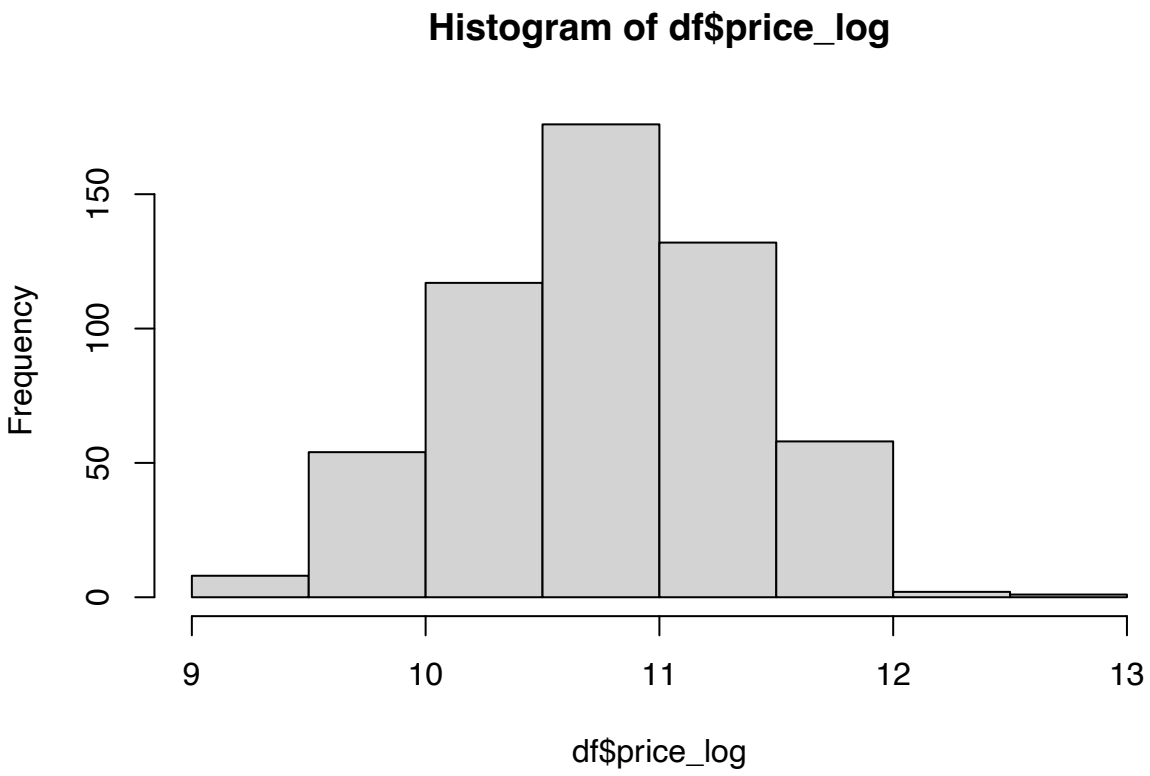
```
## [1] 0.02020202
```

Since the lambda value is near to 0.020202 we can normalize the model using the log functin

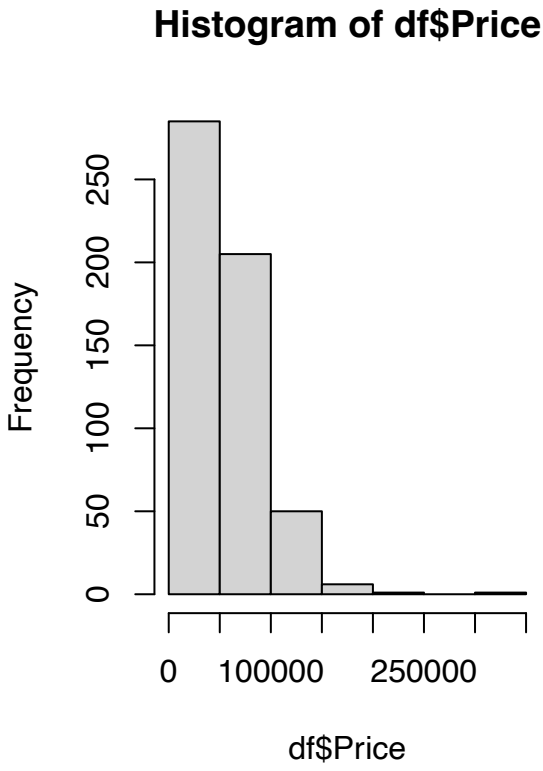
```
df$price_log = log(df$Price)

# Histogram before
hist(df$Price)
```

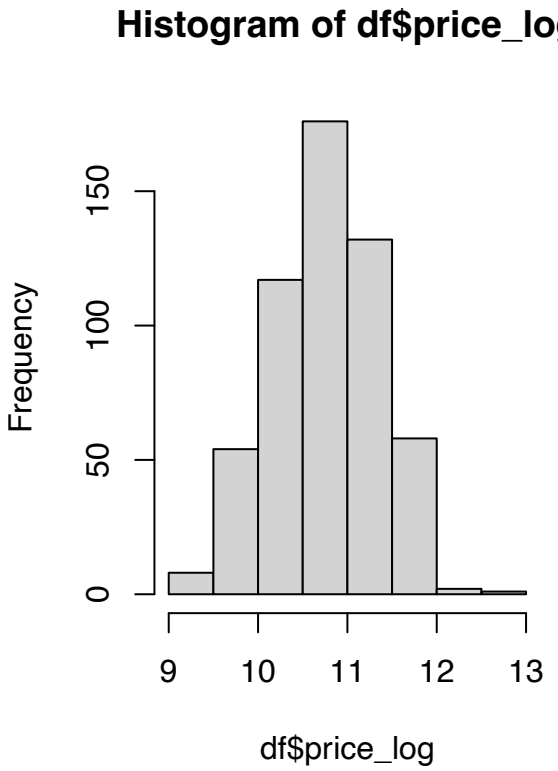



```
# Histogram after
hist(df$price_log)
```




```
# Compare histograms
par(mfrow=c(1,2))
hist(df$Price)
hist(df$price_log)
```


Histogram of df$Price
Frequency
0
50
100
150
200
250
0
100000
250000
df$Price


Histogram of df$price_log
Frequency
0
50
100
150
9
10
11
12
13
df$price_log