

# Final Project DSC 520

Reenie Christudass

2022-08-07

## Contents

Introduction - MULTIPLE LINEAR REGRESSION . . . . .	2
Research questions: . . . . .	2
Load libraries as needed . . . . .	2
Identify data set and import . . . . .	4
Summarize the variable (univariate analysis) . . . . .	5
Identify potential predictor variable and dependent variable . . . . .	6
Convert categorical variables . . . . .	6
Data transformation and cleaning . . . . .	7
Check for number of nulls in each column and fill in NA values by finding mean . . . . .	7
Correlations between numeric variables in our new data frame . . . . .	10
plotting the scatter plot . . . . .	12
Perform Linear Regression with All Predictors . . . . .	13
Check for multicollinearity . . . . .	14
Fit the model . . . . .	14
Hypothesis Testing . . . . .	14
Visualize data . . . . .	14
Predict analysis . . . . .	14
Dataset - Selected Vehicle data . . . . .	15
1.Medical insurance ( <a href="https://www.kaggle.com/datasets/mirichoi0218/insurance">https://www.kaggle.com/datasets/mirichoi0218/insurance</a> ) . . . . .	15
2. Death rate from Cancer ( <a href="https://data.world/nrippner/cancer-trials">https://data.world/nrippner/cancer-trials</a> ) . . . . .	15
3. Vehicle data ( <a href="https://www.kaggle.com/datasets/nehalbirla/vehicle-dataset-from-cardekho">https://www.kaggle.com/datasets/nehalbirla/vehicle-dataset-from-cardekho</a> ) . . . . .	15
Visualize data . . . . .	15
References . . . . .	15

## Introduction - MULTIPLE LINEAR REGRESSION

Any organization can make better decisions by using regression techniques for predictive analysis. This statistical technique can help organizations to make better decisions. Relationships between data can transform analysis into actionable information. Linear regression can fit in one dependent and one independent variable and has significant limits. In comparison, Multiple regression can overcome and fit in single dependent and multiple independent variables. This statistical techniques is used for forecasting, time series modelling and find relationship between variable.

### Research questions:

#### Load libraries as needed

```
if(!require('foreign')) {  
  install.packages('foreign')  
  library('foreign')  
}  
  
## Loading required package: foreign  
  
if(!require('tidyverse')) {  
  install.packages('tidyverse')  
  library('tidyverse')  
}  
  
## Loading required package: tidyverse  
  
## Warning: package 'tidyverse' was built under R version 4.2.1  
  
if(!require('dplyr')) {  
  install.packages('dplyr')  
  library('dplyr')  
}  
  
## Loading required package: dplyr  
  
## Warning: package 'dplyr' was built under R version 4.2.1  
  
##  
## Attaching package: 'dplyr'  
  
## The following objects are masked from 'package:stats':  
##  
##     filter, lag  
  
## The following objects are masked from 'package:base':  
##  
##     intersect, setdiff, setequal, union
```

```

if(!require('corrr')) {
  install.packages('corrr')
  library('corrr')
}

## Loading required package: corrr

## Warning: package 'corrr' was built under R version 4.2.1

install.packages("MASS", repos="http://cran.us.r-project.org")

## Installing package into 'C:/Users/chris/AppData/Local/R/win-library/4.2'
## (as 'lib' is unspecified)

## package 'MASS' successfully unpacked and MD5 sums checked

## Warning: cannot remove prior installation of package 'MASS'

## Warning in file.copy(savedcopy, lib, recursive = TRUE): problem copying C:
## \Users\chris\AppData\Local\R\win-library\4.2\00LOCK\MASS\libs\x64\MASS.dll to C:
## \Users\chris\AppData\Local\R\win-library\4.2\MASS\libs\x64\MASS.dll: Permission
## denied

## Warning: restored 'MASS'

##
## The downloaded binary packages are in
##   C:\Users\chris\AppData\Local\Temp\RtmpyAyKXb\downloaded_packages

library(MASS)

## Warning: package 'MASS' was built under R version 4.2.1

##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
## 
##     select

install.packages("corrplot ", repos="http://cran.us.r-project.org")

## Installing package into 'C:/Users/chris/AppData/Local/R/win-library/4.2'
## (as 'lib' is unspecified)

## Warning: package 'corrplot ' is not available for this version of R
##
## A version of this package for your version of R might be available elsewhere,
## see the ideas at
## https://cran.r-project.org/doc/manuals/r-patched/R-admin.html#Installing-packages

```

```

library(corrplot )

## Warning: package 'corrplot' was built under R version 4.2.1

## corrplot 0.92 loaded

install.packages("olsrr", repos="http://cran.us.r-project.org")

## Installing package into 'C:/Users/chris/AppData/Local/R/win-library/4.2'
## (as 'lib' is unspecified)

## package 'olsrr' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##   C:\Users\chris\AppData\Local\Temp\RtmpyAyKXb\downloaded_packages

library(olsrr)

## Warning: package 'olsrr' was built under R version 4.2.1

##
## Attaching package: 'olsrr'

## The following object is masked from 'package:MASS':
##
##      cement

## The following object is masked from 'package:datasets':
##
##      rivers

install.packages("ggcorrplot", repos="http://cran.us.r-project.org")

## Installing package into 'C:/Users/chris/AppData/Local/R/win-library/4.2'
## (as 'lib' is unspecified)

## package 'ggcorrplot' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##   C:\Users\chris\AppData\Local\Temp\RtmpyAyKXb\downloaded_packages

library(ggcorrplot)

## Warning: package 'ggcorrplot' was built under R version 4.2.1

## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 4.2.1

```

## Identify data set and import

```

## Set the working directory to the root of your DSC 520 directory
setwd("C:/Users/chris/dsc520/data")

## Load the `data/r4ds/heights.csv` to
df <- read.csv("C:/Users/chris/dsc520/Final_Project_Data/Car_details_v3.csv")
head(df)

##                                     name year selling_price km_driven fuel seller_type
## 1      Maruti Swift Dzire VDI 2014        450000    145500 Diesel Individual
## 2  Skoda Rapid 1.5 TDI Ambition 2014        370000    120000 Diesel Individual
## 3     Honda City 2017-2020 EXi 2006        158000    140000 Petrol Individual
## 4     Hyundai i20 Sportz Diesel 2010        225000    127000 Diesel Individual
## 5      Maruti Swift VXI BSIII 2007        130000    120000 Petrol Individual
## 6   Hyundai Xcent 1.2 VTVT E Plus 2017        440000     45000 Petrol Individual
## transmission          owner mileage engine max_power
## 1       Manual First Owner  23.4 kmpl 1248 CC     74 bhp
## 2       Manual Second Owner 21.14 kmpl 1498 CC 103.52 bhp
## 3       Manual Third Owner 17.7 kmpl 1497 CC     78 bhp
## 4       Manual First Owner 23.0 kmpl 1396 CC     90 bhp
## 5       Manual First Owner 16.1 kmpl 1298 CC  88.2 bhp
## 6       Manual First Owner 20.14 kmpl 1197 CC 81.86 bhp
##                      torque seats
## 1           190Nm@ 2000rpm      5
## 2           250Nm@ 1500-2500rpm 5
## 3           12.70@ 2,700(kgm@ rpm) 5
## 4 22.4 kgm at 1750-2750rpm 5
## 5 11.50@ 4,500(kgm@ rpm) 5
## 6           113.75nm@ 4000rpm 5

```

### Summarize the variable (univariate analysis)

```

## Summarize the variable (univariate analysis)
summary(df)

##             name              year      selling_price      km_driven
##  Length:8128      Min.   :1983      Min.   : 29999      Min.   :    1
##  Class :character  1st Qu.:2011     1st Qu.: 254999     1st Qu.: 35000
##  Mode  :character  Median :2015     Median : 450000     Median : 60000
##                  Mean   :2014     Mean   : 638272     Mean   : 69820
##                  3rd Qu.:2017     3rd Qu.: 675000     3rd Qu.: 98000
##                  Max.   :2020     Max.   :10000000     Max.   :2360457
##
##             fuel            seller_type      transmission      owner
##  Length:8128      Length:8128      Length:8128      Length:8128
##  Class :character  Class :character  Class :character  Class :character
##  Mode  :character  Mode  :character  Mode  :character  Mode  :character
##
##
```

```

##      mileage          engine        max_power        torque
##  Length:8128    Length:8128    Length:8128    Length:8128
##  Class :character  Class :character  Class :character  Class :character
##  Mode  :character  Mode  :character  Mode  :character  Mode  :character
##
##
##
##
##      seats
##  Min.   : 2.000
##  1st Qu.: 5.000
##  Median : 5.000
##  Mean   : 5.417
##  3rd Qu.: 5.000
##  Max.   :14.000
##  NA's   :221

```

Identify potential predictor variable and dependent variable

```

## One dependent variable - Price of the car
## Multiple Independent variable - year, selling_price, km_driven, fuel, seller_type, transmission,

```

Convert categorical variables

```

df <- transform(df,fuel = as.numeric(as.factor(fuel)))
df <- transform(df,seller_type = as.numeric(as.factor(seller_type)))
df <- transform(df,transmission = as.numeric(as.factor(transmission)))
df <- transform(df,owner = as.numeric(as.factor(owner)))
head(df)

##           name year selling_price km_driven fuel seller_type
## 1 Maruti Swift Dzire VDI 2014     450000    145500 2       2
## 2 Skoda Rapid 1.5 TDI Ambition 2014     370000    120000 2       2
## 3 Honda City 2017-2020 EXi 2006     158000    140000 4       2
## 4 Hyundai i20 Sportz Diesel 2010     225000    127000 2       2
## 5 Maruti Swift VXI BSIII 2007     130000    120000 4       2
## 6 Hyundai Xcent 1.2 VTVT E Plus 2017     440000    45000 4       2
##      transmission owner      mileage engine        max_power        torque
## 1             2     1 23.4 kmpl 1248 CC     74 bhp      190Nm@ 2000rpm
## 2             2     3 21.14 kmpl 1498 CC 103.52 bhp      250Nm@ 1500-2500rpm
## 3             2     5 17.7 kmpl 1497 CC     78 bhp 12.7@ 2,700(kgm@ rpm)
## 4             2     1 23.0 kmpl 1396 CC     90 bhp 22.4 kgm at 1750-2750rpm
## 5             2     1 16.1 kmpl 1298 CC    88.2 bhp 11.5@ 4,500(kgm@ rpm)
## 6             2     1 20.14 kmpl 1197 CC    81.86 bhp 113.75nm@ 4000rpm
##      seats
## 1      5
## 2      5
## 3      5
## 4      5
## 5      5
## 6      5

```

## Data transformation and cleaning

```

# Converting column km/kg to kmpl. created new column in dataframe with modified values
df$mileage_modified <- df$mileage
df$mileage_modified <- gsub("kmpL","",as.character(df$mileage_modified))
df$mileage_modified <- gsub("km/kg","",as.character(df$mileage_modified))
df$mileage_modified <- as.double(df$mileage_modified)
df$mileage_modified[endsWith(df$mileage, "km/kg")] <- as.numeric(df$mileage_modified) * 1.40

## Warning in df$mileage_modified[endsWith(df$mileage, "km/kg")] <-
## as.numeric(df$mileage_modified) * : number of items to replace is not a multiple
## of replacement length

# created new column in dataframe with modified values for engine
df$engine_modified <- gsub("CC","",as.character(df$engine))
df$engine_modified <- as.double(df$engine_modified)
# created new column in dataframe with modified values for max_power
df$max_power_modified <- gsub("bhp","",as.character(df$max_power))
df$max_power_modified <- as.double(df$max_power_modified)
head(df)

##                                     name year selling_price km_driven fuel seller_type
## 1      Maruti Swift Dzire VDI 2014     450000    145500    2        2
## 2  Skoda Rapid 1.5 TDI Ambition 2014     370000    120000    2        2
## 3      Honda City 2017-2020 EXi 2006     158000    140000    4        2
## 4      Hyundai i20 Sportz Diesel 2010     225000    127000    2        2
## 5      Maruti Swift VXI BSIII 2007     130000    120000    4        2
## 6  Hyundai Xcent 1.2 VTIV E Plus 2017     440000     45000    4        2
##   transmission owner mileage engine max_power                         torque
## 1             2     1 23.4 kmpl 1248 CC     74 bhp           190Nm@ 2000rpm
## 2             2     3 21.14 kmpl 1498 CC 103.52 bhp           250Nm@ 1500-2500rpm
## 3             2     5 17.7 kmpl 1497 CC     78 bhp           12.7@ 2,700(kgm@ rpm)
## 4             2     1 23.0 kmpl 1396 CC     90 bhp           22.4 kgm at 1750-2750rpm
## 5             2     1 16.1 kmpl 1298 CC    88.2 bhp           11.5@ 4,500(kgm@ rpm)
## 6             2     1 20.14 kmpl 1197 CC    81.86 bhp           113.75nm@ 4000rpm
##   seats mileage_modified engine_modified max_power_modified
## 1     5          23.40         1248          74.00
## 2     5          21.14         1498          103.52
## 3     5          17.70         1497          78.00
## 4     5          23.00         1396          90.00
## 5     5          16.10         1298          88.20
## 6     5          20.14         1197          81.86

```

Check for number of nulls in each column and fill in NA values by finding mean

```
colSums(is.na(df))
```

```

##               name          year  selling_price      km_driven
## 0                  0                  0                  0
## fuel      seller_type      transmission      owner

```

```

##          0          0          0          0
## mileage      engine    max_power     torque
##          0          0          0          0
## seats  mileage_modified  engine_modified max_power_modified
##        221         225         221         216

# Return the column names containing missing observations
list_na <- colnames(df)[ apply(df, 2, anyNA) ]
list_na

## [1] "seats"           "mileage_modified"   "engine_modified"
## [4] "max_power_modified"

# Exclude the missing observations
library(dplyr)
# Exclude the missing observations
df_drop <- df %>%
na.omit()

head(df)

##           name year selling_price km_driven fuel seller_type
## 1 Maruti Swift Dzire VDI 2014      450000    145500    2       2
## 2 Skoda Rapid 1.5 TDI Ambition 2014      370000    120000    2       2
## 3 Honda City 2017-2020 EXi 2006      158000    140000    4       2
## 4 Hyundai i20 Sportz Diesel 2010      225000    127000    2       2
## 5 Maruti Swift VXI BSIII 2007      130000    120000    4       2
## 6 Hyundai Xcent 1.2 VTVT E Plus 2017      440000     45000    4       2
##   transmission owner mileage engine max_power          torque
## 1            2     1 23.4 kmpl 1248 CC     74 bhp      190Nm@ 2000rpm
## 2            2     3 21.14 kmpl 1498 CC 103.52 bhp      250Nm@ 1500-2500rpm
## 3            2     5 17.7 kmpl 1497 CC     78 bhp 12.7@ 2,700(kgm@ rpm)
## 4            2     1 23.0 kmpl 1396 CC     90 bhp 22.4 kgm at 1750-2750rpm
## 5            2     1 16.1 kmpl 1298 CC    88.2 bhp    11.5@ 4,500(kgm@ rpm)
## 6            2     1 20.14 kmpl 1197 CC    81.86 bhp    113.75nm@ 4000rpm
##   seats mileage_modified engine_modified max_power_modified
## 1     5          23.40        1248        74.00
## 2     5          21.14        1498       103.52
## 3     5          17.70        1497        78.00
## 4     5          23.00        1396        90.00
## 5     5          16.10        1298        88.20
## 6     5          20.14        1197        81.86

# Create mean
average_missing <- apply(df[,colnames(df) %in% list_na], 2, mean, na.rm = TRUE)

# Create a new variable with the mean
df_replace <- df %>%
  mutate(replace_mean_seats = ifelse(is.na(seats), average_missing[1], seats),
  replace_mean_mileage_modified = ifelse(is.na(mileage_modified), average_missing[2], mileage_modified),
  replace_mean_engine_modified = ifelse(is.na(engine_modified), average_missing[3], engine_modified),
  replace_mean_max_power_modified = ifelse(is.na(max_power_modified), average_missing[4], max_power_mod

```

```
)
```

```
head(df_replace)
```

```
##           name year selling_price km_driven fuel seller_type
## 1      Maruti Swift Dzire VDI 2014        450000    145500    2        2
## 2  Skoda Rapid 1.5 TDI Ambition 2014        370000    120000    2        2
## 3      Honda City 2017-2020 EXi 2006        158000    140000    4        2
## 4      Hyundai i20 Sportz Diesel 2010        225000    127000    2        2
## 5      Maruti Swift VXI BSIII 2007        130000    120000    4        2
## 6  Hyundai Xcent 1.2 VTIV E Plus 2017        440000    45000     4        2
##   transmission owner mileage engine max_power          torque
## 1             2     1 23.4 kmpl 1248 CC     74 bhp       190Nm@ 2000rpm
## 2             2     3 21.14 kmpl 1498 CC 103.52 bhp       250Nm@ 1500-2500rpm
## 3             2     5 17.7 kmpl 1497 CC     78 bhp 12.7@ 2,700(kgm@ rpm)
## 4             2     1 23.0 kmpl 1396 CC     90 bhp 22.4 kgm at 1750-2750rpm
## 5             2     1 16.1 kmpl 1298 CC    88.2 bhp    11.5@ 4,500(kgm@ rpm)
## 6             2     1 20.14 kmpl 1197 CC    81.86 bhp    113.75nm@ 4000rpm
##   seats mileage_modified engine_modified max_power_modified replace_mean_seats
## 1     5      23.40          1248        74.00            5
## 2     5      21.14          1498        103.52           5
## 3     5      17.70          1497        78.00           5
## 4     5      23.00          1396        90.00           5
## 5     5      16.10          1298        88.20           5
## 6     5      20.14          1197        81.86           5
##   replace_mean_mileage_modified replace_mean_engine_modified
## 1                  23.40          1248
## 2                  21.14          1498
## 3                  17.70          1497
## 4                  23.00          1396
## 5                  16.10          1298
## 6                  20.14          1197
##   replace_mean_max_power_modified
## 1                  74.00
## 2                 103.52
## 3                  78.00
## 4                  90.00
## 5                  88.20
## 6                  81.86
```

```
df_new <- subset(df_replace, select = c(name, year, selling_price, km_driven, fuel, seller_type, transmission))
head(df_new)
```

```
##           name year selling_price km_driven fuel seller_type
## 1      Maruti Swift Dzire VDI 2014        450000    145500    2        2
## 2  Skoda Rapid 1.5 TDI Ambition 2014        370000    120000    2        2
## 3      Honda City 2017-2020 EXi 2006        158000    140000    4        2
## 4      Hyundai i20 Sportz Diesel 2010        225000    127000    2        2
## 5      Maruti Swift VXI BSIII 2007        130000    120000    4        2
## 6  Hyundai Xcent 1.2 VTIV E Plus 2017        440000    45000     4        2
##   transmission owner mileage replace_mean_seats
## 1             2     1 23.4 kmpl            5
```

```

## 2      2      3 21.14 kmpl      5
## 3      2      5 17.7 kmpl      5
## 4      2      1 23.0 kmpl      5
## 5      2      1 16.1 kmpl      5
## 6      2      1 20.14 kmpl      5
##   replace_mean_mileage_modified replace_mean_engine_modified
## 1              23.40          1248
## 2              21.14          1498
## 3              17.70          1497
## 4              23.00          1396
## 5              16.10          1298
## 6              20.14          1197
##   replace_mean_max_power_modified
## 1              74.00
## 2              103.52
## 3              78.00
## 4              90.00
## 5              88.20
## 6              81.86

```

## Correlations between numeric variables in our new data frame

```

## Identify columns in dataframe that are numeric

df <- select_if(df_new, is.numeric)
head(df)

## year selling_price km_driven fuel seller_type transmission owner
## 1 2014      450000    145500  2        2        2        1
## 2 2014      370000    120000  2        2        2        3
## 3 2006      158000    140000  4        2        2        5
## 4 2010      225000    127000  2        2        2        1
## 5 2007      130000    120000  4        2        2        1
## 6 2017      440000    45000   4        2        2        1
##   replace_mean_seats replace_mean_mileage_modified replace_mean_engine_modified
## 1              5                  23.40          1248
## 2              5                  21.14          1498
## 3              5                  17.70          1497
## 4              5                  23.00          1396
## 5              5                  16.10          1298
## 6              5                  20.14          1197
##   replace_mean_max_power_modified
## 1              74.00
## 2              103.52
## 3              78.00
## 4              90.00
## 5              88.20
## 6              81.86

## Correlation
## Correlation in numbers

```

```
corr <- round(cor(df), 1)
corr
```

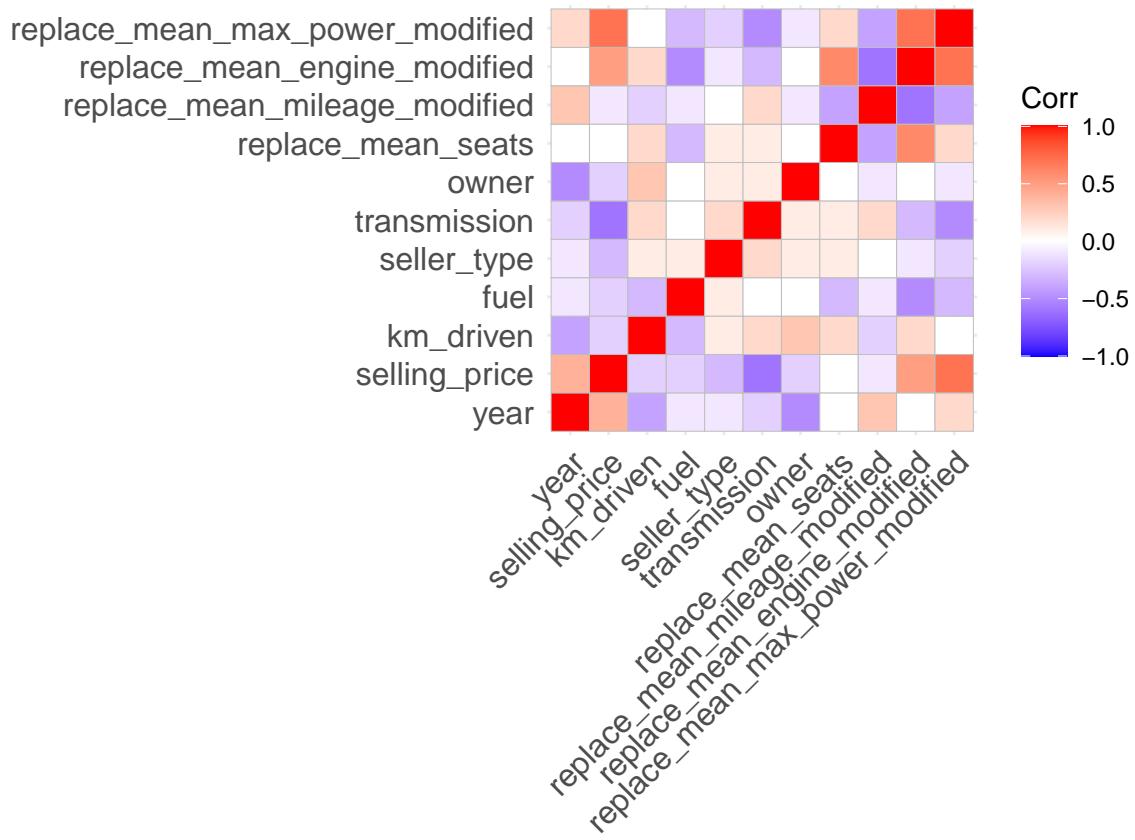
	year	selling_price	km_driven	fuel	seller_type
## year	1.0	0.4	-0.4	-0.1	-0.1
## selling_price	0.4	1.0	-0.2	-0.2	-0.3
## km_driven	-0.4	-0.2	1.0	-0.3	0.1
## fuel	-0.1	-0.2	-0.3	1.0	0.1
## seller_type	-0.1	-0.3	0.1	0.1	1.0
## transmission	-0.2	-0.6	0.2	0.0	0.2
## owner	-0.5	-0.2	0.3	0.0	0.1
## replace_mean_seats	0.0	0.0	0.2	-0.3	0.1
## replace_mean_mileage_modified	0.3	-0.1	-0.2	-0.1	0.0
## replace_mean_engine_modified	0.0	0.5	0.2	-0.5	-0.1
## replace_mean_max_power_modified	0.2	0.7	0.0	-0.3	-0.2
## transmission			owner	replace_mean_seats	
## year			-0.2	-0.5	0.0
## selling_price			-0.6	-0.2	0.0
## km_driven			0.2	0.3	0.2
## fuel			0.0	0.0	-0.3
## seller_type			0.2	0.1	0.1
## transmission			1.0	0.1	0.1
## owner			0.1	1.0	0.0
## replace_mean_seats			0.1	0.0	1.0
## replace_mean_mileage_modified			0.2	-0.1	-0.4
## replace_mean_engine_modified			-0.3	0.0	0.6
## replace_mean_max_power_modified			-0.5	-0.1	0.2
## replace_mean_mileage_modified					0.3
## year					-0.1
## selling_price					-0.2
## km_driven					-0.1
## fuel					0.0
## seller_type					0.2
## transmission					-0.1
## owner					-0.4
## replace_mean_seats					1.0
## replace_mean_mileage_modified					-0.6
## replace_mean_engine_modified					-0.4
## replace_mean_max_power_modified					0.0
## replace_mean_engine_modified					0.5
## year					0.2
## selling_price					-0.5
## km_driven					0.2
## fuel					-0.5
## seller_type					-0.1
## transmission					-0.3
## owner					0.0
## replace_mean_seats					0.6
## replace_mean_mileage_modified					-0.6
## replace_mean_engine_modified					1.0
## replace_mean_max_power_modified					0.7
## replace_mean_max_power_modified					0.2
## year					

```

## selling_price          0.7
## km_driven              0.0
## fuel                   -0.3
## seller_type             -0.2
## transmission            -0.5
## owner                  -0.1
## replace_mean_seats       0.2
## replace_mean_mileage_modified   -0.4
## replace_mean_engine_modified      0.7
## replace_mean_max_power_modified    1.0

## Correlation in graph
ggcorrplot(corr)

```



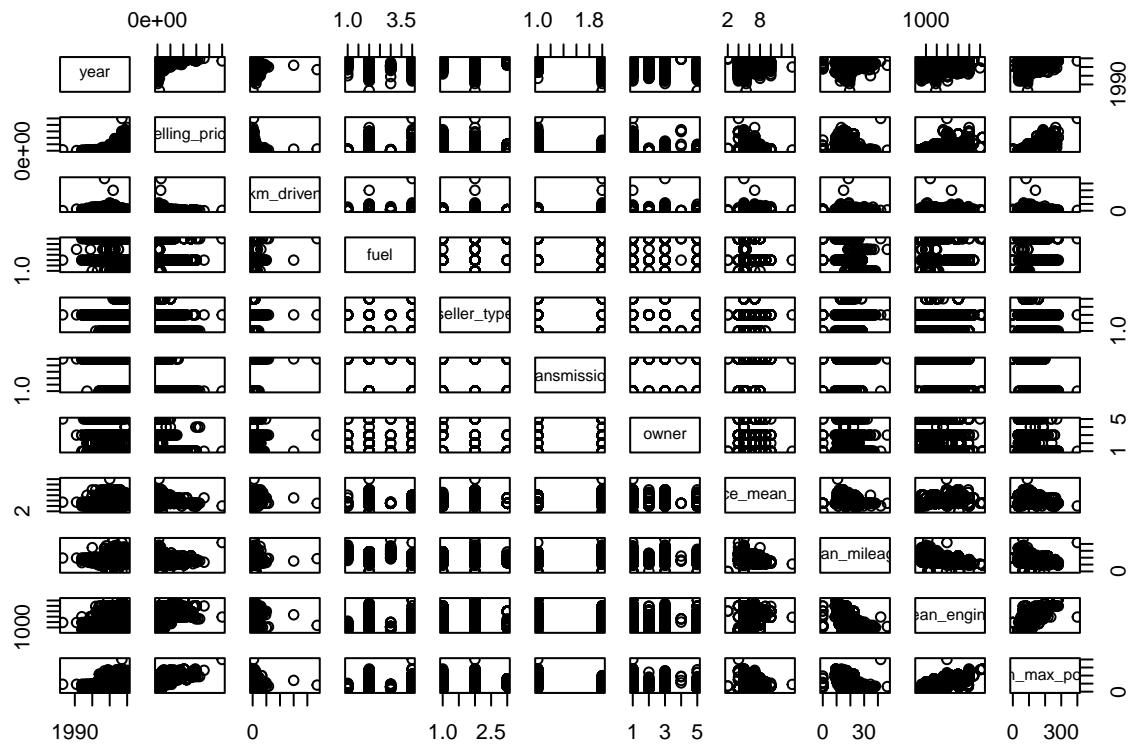
```
#CONCLUSION: Variable with highest correlation is a good predictor( > .5)
```

plotting the scatter plot

```

### Check the relationship using scatter plot
plot(df)

```



## Perform Linear Regression with All Predictors

```
my_model <- lm(df, formula=selling_price ~.)
summary(my_model)
```

```
##
## Call:
## lm(formula = selling_price ~ ., data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -2264019 -191685   -1122   156105  4612253 
##
## Coefficients:
## (Intercept)  Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -6.913e+07 3.401e+06 -20.325 < 2e-16 ***
## year         3.473e+04 1.698e+03  20.460 < 2e-16 ***
## km_driven    -1.074e+00 1.068e-01 -10.061 < 2e-16 ***
## fuel          -2.402e+04 7.131e+03 -3.368 0.00076 ***
## seller_type   -2.187e+05 1.362e+04 -16.055 < 2e-16 ***
## transmission -4.799e+05 1.910e+04 -25.122 < 2e-16 ***
## owner         -1.009e+04 4.766e+03 -2.117 0.03430 *  
## replace_mean_seats -4.068e+04 7.854e+03 -5.179 2.28e-07 ***
## replace_mean_mileage_modified 1.128e+04 1.998e+03  5.645 1.71e-08 ***
```

```

## replace_mean_engine_modified      1.018e+02  2.327e+01  4.373 1.24e-05 ***
## replace_mean_max_power_modified  1.250e+04  2.550e+02  49.032 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 456900 on 8117 degrees of freedom
## Multiple R-squared:  0.6792, Adjusted R-squared:  0.6788
## F-statistic:  1719 on 10 and 8117 DF,  p-value: < 2.2e-16

## CONCLUSION: Most of the variable can be participate in the multiple linear
## regression other than owner

```

Check for multicollinearity

```

##TOLERANCE & VARIANCE INFLATION FACTOR (VIF)
library("olsrr")
ols_vif_tol(my_model)

```

	Variables	Tolerance	VIF
## 1	year	0.5449152	1.835148
## 2	km_driven	0.7047259	1.418991
## 3	fuel	0.5028443	1.988687
## 4	seller_type	0.8895366	1.124181
## 5	transmission	0.6257568	1.598065
## 6	owner	0.7577952	1.319618
## 7	replace_mean_seats	0.4648983	2.151008
## 8	replace_mean_mileage_modified	0.3889383	2.571102
## 9	replace_mean_engine_modified	0.1919687	5.209182
## 10	replace_mean_max_power_modified	0.3163011	3.161545

```

## CONCLUSION: VIF above 10 shows multicollinearity. I did not notice anything in the dataset

```

Fit the model

Hypothesis Testing

$H_0 = H_a$  or  $H_0 \neq H_a$

Visualize data

Create plot to show the current vs predicted values

Predict analysis

Are there any outliers?  
 Are there missing values? Remove or find mean ?  
 Handle categorical variables?

## **Dataset - Selected Vehicle data**

### **1. Medical insurance (<https://www.kaggle.com/datasets/mirichoi0218/insurance>)**

Does the cost of the insurance is related with age, habits (smoking vs non smoking), BMI , region etc.  
Predict the cost increase / decrease on the cost of the insurance?

### **2. Death rate from Cancer (<https://data.world/nrippner/cancer-trials>)**

Who is prevailing the trails? Is it been driven by the environment conditions or wealth

### **3. Vehicle data (<https://www.kaggle.com/datasets/nehalbirla/vehicle-dataset-from-cardekho>)**

Is the price of the car increase or decrease as the mileage goes up or because of accident etc? Any other factor.

## **Visualize data**

Scatter plot - Predicted value vs Actual value - Plot the residual vs fitted Quantile-Quantile plots -determining if two data sets come from populations with a common distribution. Histogram - to find residuals

## **References**