



Automobile - Price Prediction

Team

AMBIKA SHARMA

VATSAL PANCHOLI

TABLE OF CONTENTS

INTRODUCTION _____	3
DATA SET URL _____	5
DATA DESCRIPTION _____	7
BMW, AUDI & MERCEDES _____	8
DATA CLEANING _____	9
Distinct BMW Model _____	10
Distinct Audi Model _____	10
Distinct Merc Model _____	10
ANALYSIS & VISUALIZATION _____	11
Most Common BMW Model _____	11
Most Common AUDI Model _____	12
Most Common Mercedes Model _____	13
BMW Model and their prices _____	14
Audi Model and their prices _____	16
Mercedes Model and their prices _____	17
Numeric Variables associations with Sales - BMW, AUDI and MERC _____	19
Correlation Plot - BMW, AUDI and MERC _____	25
Fuel Type - BMW, AUDI and MERC _____	27
STATISTICAL SUMMARY AND FUNCTIONS _____	30
Statistical Summary: _____	30
Functions Used: _____	33

INTRODUCTION

As a Data scientist it is required to apply some data science techniques for the price of cars with the available independent variables. That should help the management to understand how exactly the prices vary with the independent variables. They can accordingly manipulate the design of the cars, the business strategy etc. to meet certain price levels. For example, there is a German automobile company such as BMW, Audi and Mercedes which aspires to enter the US market by setting up their manufacturing unit there and producing cars locally to give competition to their US and European counterparts.

They want to understand the factors affecting the pricing of cars in the American market, since those may be very different from the German market. Essentially, the company wants to know:

- Which variables are significant in predicting the price of a car?
- How well those variables describe the price of a car?

Based on various market surveys, the consulting firm has gathered a large dataset of different types of cars across the American market including the model, the year, the price of the car used to compare, transmission, mileage, fuel type, tax, mileage (mpg) and the size of the engine. Model discusses about the model's name of the car assigned by BMW, Audi, Mercedes respectively. Year specifies the model's year of the car assigned by BMW, Audi and Mercedes. Price specifies the estimated price of the car declared by BMW, Audi and Mercedes. Transmission is another name for a car's gearbox, the component that turns the gear power into something the car can use. In-short, transmission refers to the whole drivetrain including clutch, differential, and final gear shaft. Mileage of a car is the efficiency of engine equipped in a car. The aggregate number of miles travelled over in each time, length, extent, or distance is the mileage. Similarly, for the fuel type - most cars run on gasoline, a refined petroleum distillate. However, there are other types of fuel as well. Diesel is a different type of fuel obtained from

crude oil. Biodiesel is also another type of fuel that can be used. Road tax known by various names around the world, is a tax which must be paid on, or included with, a wheeled vehicle to use it on a public road. MPG, or miles per gallon, is the distance, estimated in miles, that a vehicle can travel for each gallon of fuel. MPG is additionally the essential estimation of a vehicle's eco-friendliness: The higher a vehicle's MPG, the eco-friendlier it is. Engine size is the volume of fuel and air that can be pushed through a car's cylinders and is measured in cubic centimeters (cc). For example, a car that has a 1390cc engine would be described as a 1.4 liter. Traditionally, a car with a bigger engine would generate more power than a car with a smaller engine.¹

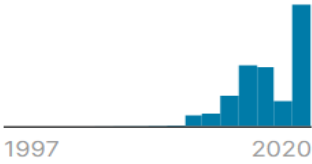
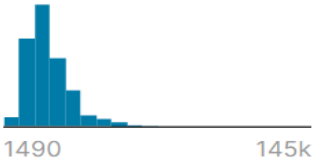
“Life is so uncertain, but the car keeps us going, isn't?”

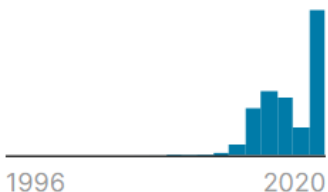
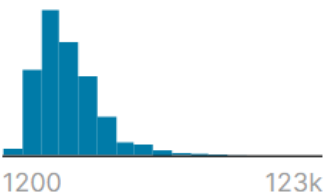
¹ <https://towardsdatascience.com/predicting-car-price-using-machine-learning-8d2df3898f16>

DATA SET URL

This dataset consists of more than 10,000 rows and 9 columns:

We will be concentrating on types of cars like Audi, BMW, Mercedes, comparing the price, model, mileage fuel type etc. in each car and accordingly check which one is more popular amongst the costumers.² We are working on three excel sheets.

Audi used car listings.			
A model	# year	# price	A transmission
audi model	registration year	price in £	type of gearbox
A3 18%			Manual
Q3 13%			Semi-Auto
Other (7322) 69%			Other (2708)
A1	2017	12500	Manual
A6	2016	16500	Automatic

BMW used car listings.			
A model	# year	# price	A transmission
BMW model	registration year	price in £	type of gearbox
3 Series 23%			Semi-Auto
1 Series 18%			Automatic
Other (6369) 59%			Other (2527)
5 Series	2014	11200	Automatic
6 Series	2018	27000	Automatic

² <https://www.kaggle.com/danielkyrka/bmw-pricing-challenge>

model	year	price	transmission	mileage	fuelType	tax	mpg	engineSize
A1	2017	12500	Manual	15735	Petrol	150	55.4	1.4
A6	2016	16500	Automatic	36203	Diesel	20	64.2	2
A1	2016	11000	Manual	29946	Petrol	30	55.4	1.4
A4	2017	16800	Automatic	25952	Diesel	145	67.3	2
A3	2019	17300	Manual	1998	Petrol	145	49.6	1
A1	2016	13900	Automatic	32260	Petrol	30	58.9	1.4
A6	2016	13250	Automatic	76788	Diesel	30	61.4	2
A4	2016	11750	Manual	75185	Diesel	20	70.6	2
A3	2015	10200	Manual	46112	Petrol	20	60.1	1.4
A1	2016	12000	Manual	22451	Petrol	30	55.4	1.4
A3	2017	16100	Manual	28955	Petrol	145	58.9	1.4
A6	2016	16500	Automatic	52198	Diesel	125	57.6	2
Q3	2016	17000	Manual	44915	Diesel	145	52.3	2
A3	2017	16400	Manual	21695	Petrol	30	58.9	1.4
A6	2015	15400	Manual	47348	Diesel	30	61.4	2
A3	2017	14500	Automatic	26156	Petrol	145	58.9	1.4
Q3	2016	15700	Automatic	28396	Diesel	145	53.3	2
A3	2014	13900	Automatic	30516	Petrol	30	56.5	1.4
Q5	2016	19000	Automatic	37652	Diesel	200	47.1	2
Q3	2014	17000	Manual	34110	Petrol	145	47.9	1.4

model	year	price	transmission	mileage	fuelType	tax	mpg	engineSize
5 Series	2014	11200	Automatic	67068	Diesel	125	57.6	2
6 Series	2018	27000	Automatic	14827	Petrol	145	42.8	2
5 Series	2016	16000	Automatic	62794	Diesel	160	51.4	3
1 Series	2017	12750	Automatic	26676	Diesel	145	72.4	1.5
7 Series	2014	14500	Automatic	39554	Diesel	160	50.4	3
5 Series	2016	14900	Automatic	35309	Diesel	125	60.1	2
5 Series	2017	16000	Automatic	38538	Diesel	125	60.1	2
2 Series	2018	16250	Manual	10401	Petrol	145	52.3	1.5
4 Series	2017	14250	Manual	42668	Diesel	30	62.8	2
5 Series	2016	14250	Automatic	36099	Diesel	20	68.9	2
X3	2017	15500	Manual	74907	Diesel	145	52.3	2
1 Series	2017	11800	Manual	29840	Diesel	20	68.9	2
X3	2016	15500	Automatic	77823	Diesel	125	54.3	2
2 Series	2015	10500	Manual	31469	Diesel	20	68.9	2
X3	2017	22000	Automatic	19057	Diesel	145	54.3	2
3 Series	2017	16500	Manual	16570	Diesel	125	58.9	2
3 Series	2017	14250	Automatic	55594	Other	135	148.7	2
3 Series	2017	16000	Automatic	45456	Diesel	30	64.2	2
1 Series	2017	15500	Automatic	22812	Diesel	20	68.9	1.5
4 Series	2014	14000	Automatic	47348	Diesel	125	60.1	2
1 Series	2015	9700	Automatic	75124	Diesel	20	70.6	2
3 Series	2015	12600	Automatic	78957	Diesel	30	62.8	2

DATA DESCRIPTION

> str(BMW)

```
'data.frame': 10781 obs. of 9 variables:
 $ model      : chr  " 5 Series" " 6 Series" " 5 Series" " 1 Series" ...
 $ year       : int   2014 2018 2016 2017 2014 2016 2017 2018 2017 2016 ...
 $ price      : int   11200 27000 16000 12750 14500 14900 16000 16250 14250 14250 ...
 $ transmission: chr   "Automatic" "Automatic" "Automatic" "Automatic" ...
 $ mileage    : int   67068 14827 62794 26676 39554 35309 38538 10401 42668 36099 ...
 $ fuelType   : chr   "Diesel" "Petrol" "Diesel" "Diesel" ...
 $ tax        : int   125 145 160 145 160 125 125 145 30 20 ...
 $ mpg        : num   57.6 42.8 51.4 72.4 50.4 60.1 60.1 52.3 62.8 68.9 ...
 $ engineSize : num    2 2 3 1.5 3 2 2 1.5 2 2 ...
```

> str(AUDI)

```
'data.frame': 10668 obs. of 9 variables:
 $ model      : chr   " A1" " A6" " A1" " A4" ...
 $ year       : int   2017 2016 2016 2017 2019 2016 2016 2016 2015 2016 ...
 $ price      : int   12500 16500 11000 16800 17300 13900 13250 11750 10200 12000 ...
 $ transmission: chr   "Manual" "Automatic" "Manual" "Automatic" ...
 $ mileage    : int   15735 36203 29946 25952 1998 32260 76788 75185 46112 22451 ...
 $ fuelType   : chr   "Petrol" "Diesel" "Petrol" "Diesel" ...
 $ tax        : int   150 20 30 145 145 30 30 20 20 30 ...
 $ mpg        : num   55.4 64.2 55.4 67.3 49.6 58.9 61.4 70.6 60.1 55.4 ...
 $ engineSize : num    1.4 2 1.4 2 1 1.4 2 2 1.4 1.4 ...
```

> str(MERC)

```
'data.frame': 13119 obs. of 9 variables:
 $ model      : chr   " SLK" " S Class" " SL CLASS" " G Class" ...
 $ year       : int   2005 2017 2016 2016 2016 2011 2018 2012 2019 2017 ...
 $ price      : int   5200 34948 49948 61948 73948 149948 30948 10948 139948 19750 ...
 $ transmission: chr   "Automatic" "Automatic" "Automatic" "Automatic" ...
 $ mileage    : int   63000 27000 6200 16000 4000 3000 16000 107000 12000 15258 ...
 $ fuelType   : chr   "Petrol" "Hybrid" "Petrol" "Petrol" ...
 $ tax        : int   325 20 555 325 325 570 145 265 145 30 ...
 $ mpg        : num   32.1 61.4 28 30.4 30.1 21.4 47.9 36.7 21.4 64.2 ...
 $ engineSize : num    1.8 2.1 5.5 4 4 6.2 2.1 3.5 4 2.1 ...
```







BMW, AUDI & MERCEDES

- **Model – Character** – This column discusses about the model's name of the car assigned by BMW, Audi, Mercedes respectively.
- **Year – Integer** – This column specifies the model's year of the car assigned by BMW, Audi and Mercedes.
- **Price – Integer** – This column specifies the estimated price of the car declared by BMW, Audi and Mercedes.
- **Transmission – Character** – A transmission is another name for a car's gearbox, the component that turns the gear power into something the car can use. In-short, transmission refers to the whole drivetrain including clutch, differential, and final gear shaft.
- **Mileage – Integer** – Mileage of a car is the efficiency of engine equipped in a car. The aggregate number of miles travelled over in each time, length, extent, or distance is the mileage.
- **Fuel Type – Character** – Most cars run on gasoline, a refined petroleum distillate. However, there are other types of fuel as well. Diesel is a different type of fuel obtained from crude oil. Biodiesel is also another type of fuel that can be used.
- **Tax – Integer** – Road tax known by various names around the world, is a tax which has to be paid on, or included with, a wheeled vehicle to use it on a public road.
- **Mpg – Num** – MPG, or miles per gallon, is the distance, estimated in miles, that a vehicle can travel for each gallon of fuel. MPG is additionally the essential estimation of a vehicle's eco-friendliness: The higher a vehicle's MPG, the eco-friendlier it is.
- **Engine Size – Num** – Engine size is the volume of fuel and air that can be pushed through a car's cylinders and is measured in cubic centimeters (cc). For example, a car that has a 1390cc engine would be described as a 1.4 liter. Traditionally, a car with a bigger engine would generate more power than a car with a smaller engine.

DATA CLEANING

```
> BMW <- read.csv("bmw.csv")
> AUDI <- read.csv("audi.csv")
> MERC <- read.csv("merc.csv")
```

To which it read -

Data		
 AUDI	10668 obs. of 9 variables	
 BMW	10781 obs. of 9 variables	
 MERC	13119 obs. of 9 variables	

As we are working on three excel sheets – **bmw.csv**, **merc.csv** and **audi.csv**. We have assigned each csv with unique variable – BMW, MERC, and AUDI.

```
> any(is.na(BMW))
[1] FALSE
> any(is.na(AUDI))
[1] FALSE
> any(is.na(MERC))
[1] FALSE
```

In R, missing qualities are addressed by NA (not accessible). Unusual values (e.g., division by 0) is represented by NaN (not a number). In contrast to SAS, R utilizes a similar symbol for character and numeric information.³

There appears to be no mislaid data in the given data frame for all three data sets.

So after installing packages – “magrittr” and “dplyr”

```
library(magrittr)
```

```
library(dplyr)
```

We will be able to find the unique data for models of the car, as we want to see only specific value of Models of what “BMW”, “AUDI” and “MERC” launches.

³ <https://www.statmethods.net/input/missingdata.html>

Distinct BMW Model

```
> BMW$model %>% unique()
[1] " 5 Series" " 6 Series" " 1 Series"
[4] " 7 Series" " 2 Series" " 4 Series"
[7] " X3"      " 3 Series" " X5"
[10] " X4"      " i3"       " X1"
[13] " M4"      " X2"       " X6"
[16] " 8 Series" " Z4"       " X7"
[19] " M5"      " i8"       " M2"
[22] " M3"      " M6"       " Z3"
```

Distinct Audi Model

```
> AUDI$model %>% unique()
[1] " A1" " A6" " A4" " A3" " Q3" " Q5"
[7] " A5" " S4" " Q2" " A7" " TT" " Q7"
[13] " RS6" " RS3" " A8" " Q8" " RS4" " RS5"
[19] " R8" " SQ5" " S8" " SQ7" " S3" " S5"
[25] " A2" " RS7"
```

Distinct Merc Model

```
> MERC$model %>% unique()
[1] " SLK" " S Class" " SL CLASS"
[4] " G Class" " GLE Class" " GLA Class"
[7] " A Class" " B Class" " GLC Class"
[10] " C Class" " E Class" " GL Class"
[13] " CLS Class" " CLC Class" " CLA Class"
[16] " V Class" " M Class" " CL Class"
[19] " GLS Class" " GLB Class" " X-CLASS"
[22] "180" " CLK" " R Class"
[25] "230" "220" "200"
```

Unique () returns a vector, data frame or array with duplicate elements/rows removed.⁴ The above categorization is based on models of BMW, AUDI, and MERC, where it returns unique value of the model. The data for model is cleaned by using unique.

⁴ <https://www.rdocumentation.org/packages/base/versions/3.6.2/topics/unique>

ANALYSIS & VISUALIZATION

Most Common BMW Model

```
install.packages("ggplot")
```

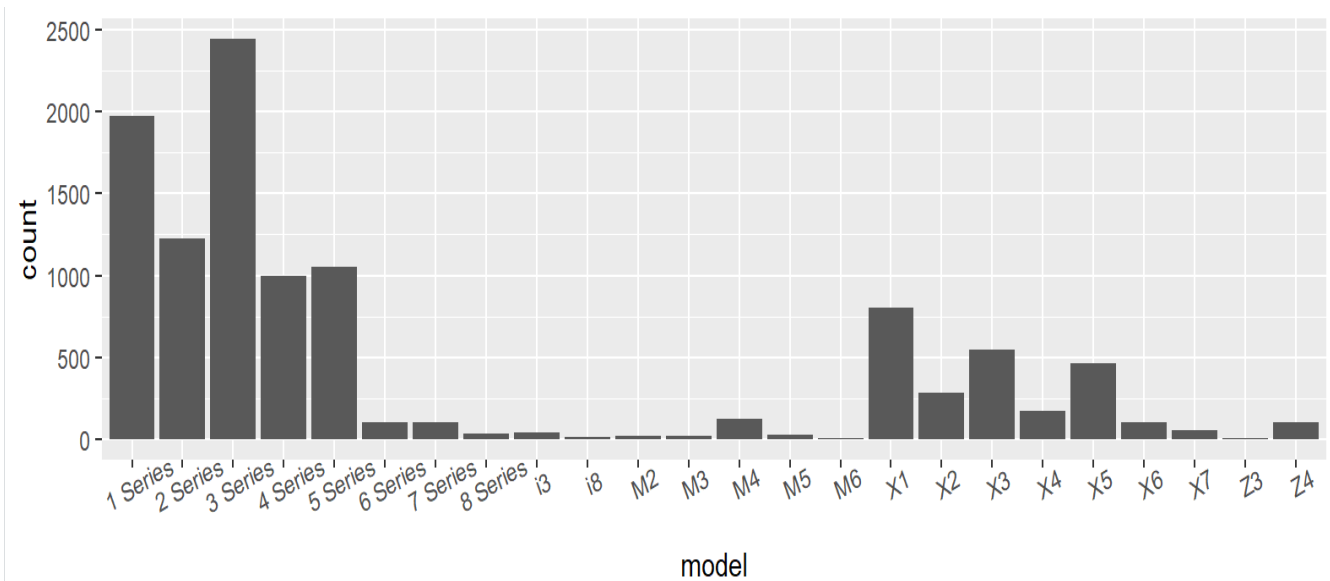
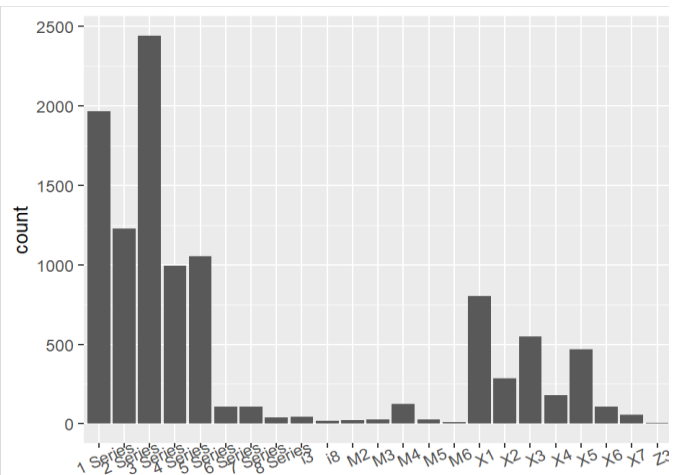
```
library(ggplot2)
```

```
ggplot(BMW, aes(x= model))+ geom_bar()+theme(axis.text.x = element_text(angle = 25))
```

```
> install.packages("ggplot")
WARNING: Rtools is required to build R packages but
is not currently installed. Please download and ins
tall the appropriate version of Rtools before procee
ding:

https://cran.rstudio.com/bin/windows/Rtools/
Installing package into 'C:/Users/asharm49/Document
s/R/win-library/4.1'
(as 'lib' is unspecified)
Warning in install.packages :
  package 'ggplot' is not available for this version
of R

A version of this package for your version of R migh
t be available elsewhere,
see the ideas at
https://cran.r-project.org/doc/manuals/r-patched/R-a
dmin.html#Installing-packages
> library(ggplot2)
> ggplot(BMW, aes(x= model))+
+   geom_bar()+
+   theme(axis.text.x = element_text(angle = 25))
```



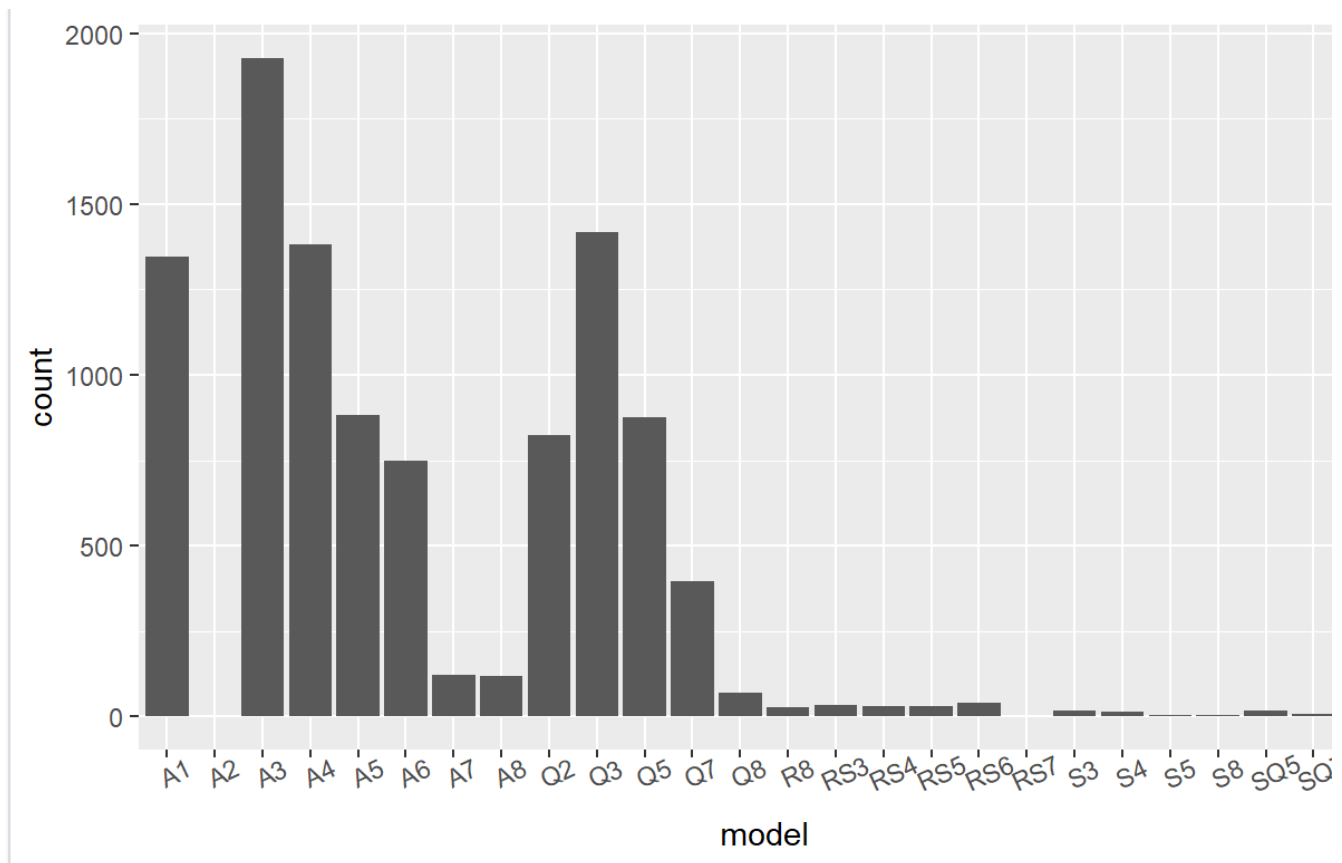
It appears that 3 Series then 1 Series are the most common models among the BMWs.

R features:

- Plot Type – Bar Chart
- Library – ggplot2

Most Common AUDI Model

```
ggplot(AUDI, aes(x= model))+
+   geom_bar()+
+   theme(axis.text.x = element_text(angle = 25))
```

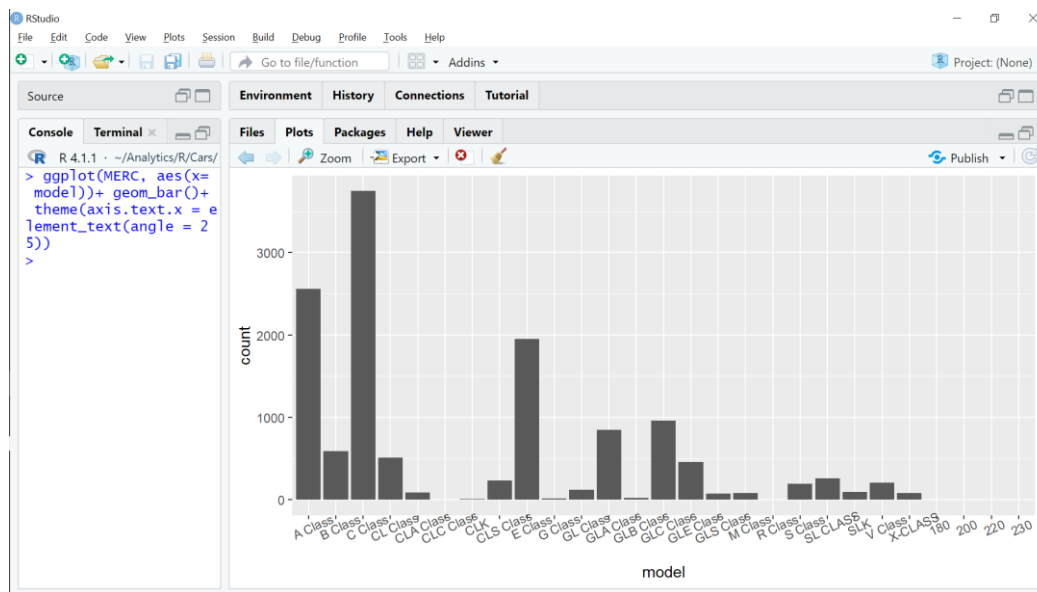


It appears that A3 then Q3 are the most common models among Audi.

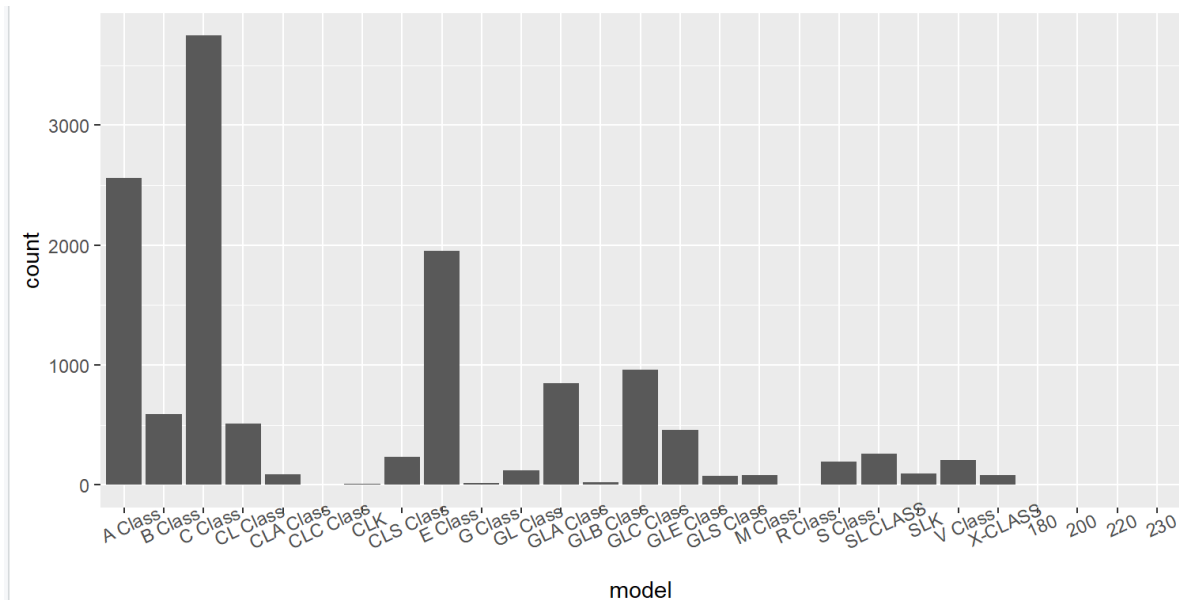
R features:

- Plot Type – Bar Chart
- Library – ggplot2

Most Common Mercedes Model



```
ggplot(MERC, aes(x= model))+
+   geom_bar()+
+   theme(axis.text.x = element_text(angle = 25))
```

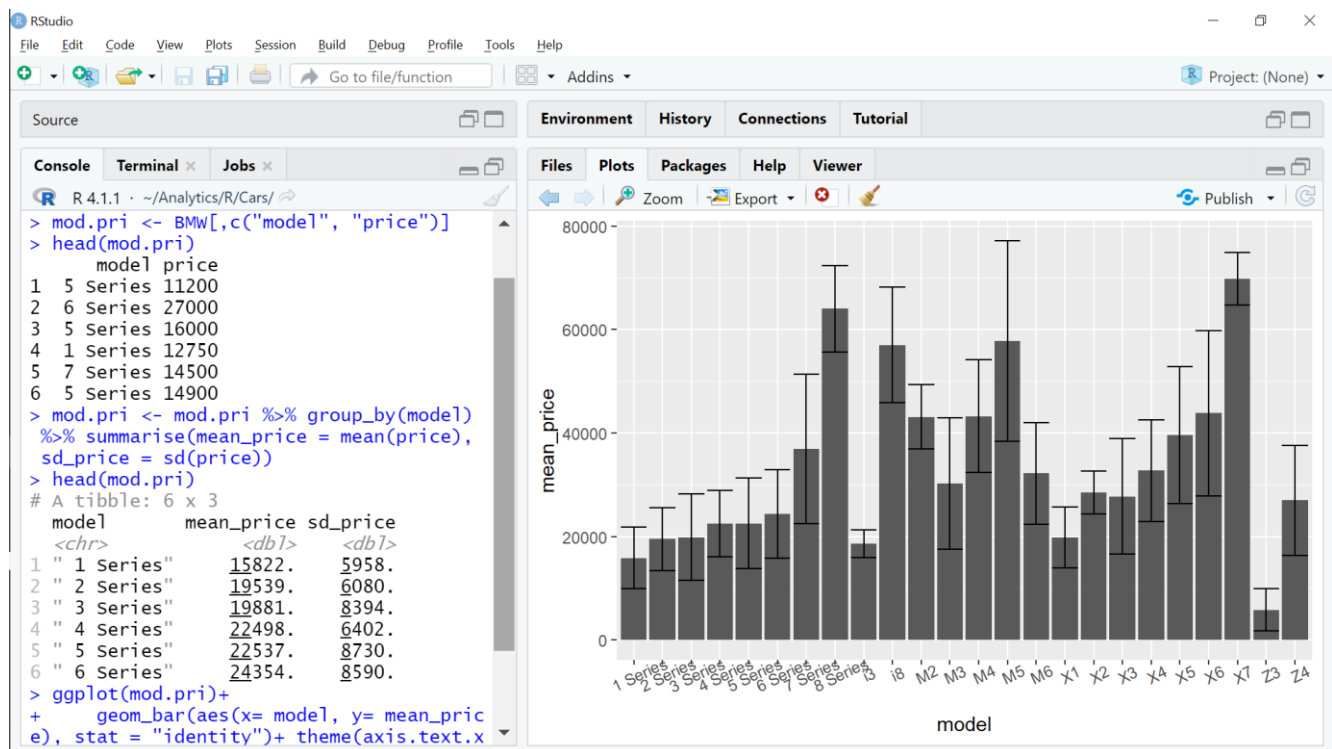


It appears that C class then A class are the most common models among Mercedes.

R features:

- Plot Type – Bar Chart
- Library – ggplot2

BMW Model and their prices



```
mod.pri <- BMW[,c("model", "price")]
```

```
head(mod.pri)
```

```
  model price
```

```
1 5 Series 11200
```

```
2 6 Series 27000
```

```
3 5 Series 16000
```

```
4 1 Series 12750
```

```
5 7 Series 14500
```

```
6 5 Series 14900
```

```
mod.pri <- mod.pri %>% group_by(model) %>% summarise(mean_price = mean(price), sd_price =
sd(price))
```

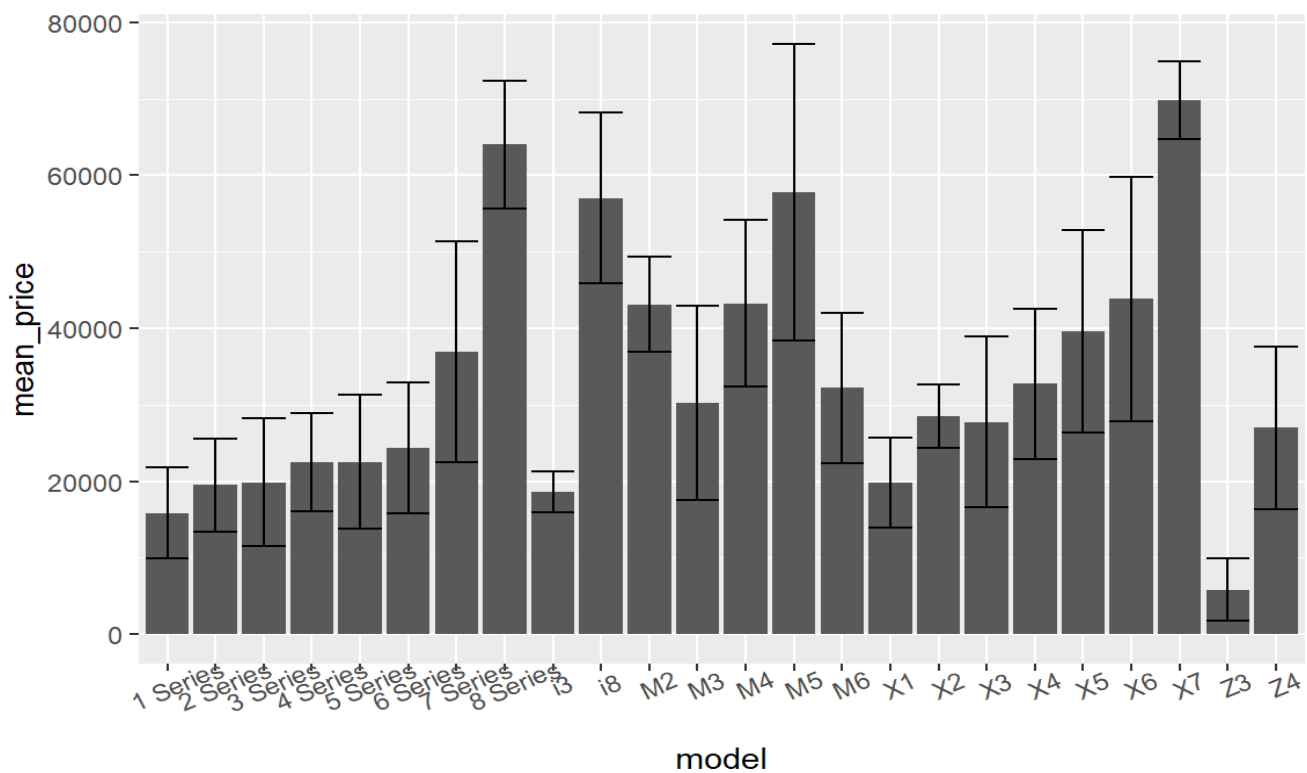
```
head(mod.pri)
```

```
# A tibble: 6 x 3
```

```
  model    mean_price sd_price
```


	<chr	<dbl	<dbl
1	" 1 Series"	15822.	5958.
2	" 2 Series"	19539.	6080.
3	" 3 Series"	19881.	8394.
4	" 4 Series"	22498.	6402.
5	" 5 Series"	22537.	8730.
6	" 6 Series"	24354.	8590.

```
ggplot(mod.pri)+ geom_bar(aes(x= model, y= mean_price), stat = "identity")+ theme(axis.text.x =
element_text(angle = 25))+ geom_errorbar(aes(x = model, ymin = mean_price - sd_price, ymax =
mean_price + sd_price))
```



Mean price of model M5 is the highest followed by X7 and 8 series in BMW.

R features:

- Plot Type – Bar Median Chart
- Library – ggplot2

Audi Model and their prices

```
mod.pri <- AUDI[,c("model", "price")]
```

```
head(mod.pri)
```

```
model price
```

```
1  A1 12500
```

```
2  A6 16500
```

```
3  A1 11000
```

```
4  A4 16800
```

```
5  A3 17300
```

```
6  A1 13900
```

```
mod.pri <- mod.pri % % group_by(model) % % summarise(mean_price = mean(price), sd_price =  
sd(price))
```

```
head(mod.pri)
```

```
# A tibble: 6 x 3
```

```
model mean_price sd_price
```

```
<chr    <dbl    <dbl
```

```
1 " A1"    14328.  4646.
```

```
2 " A2"    2490    NA
```

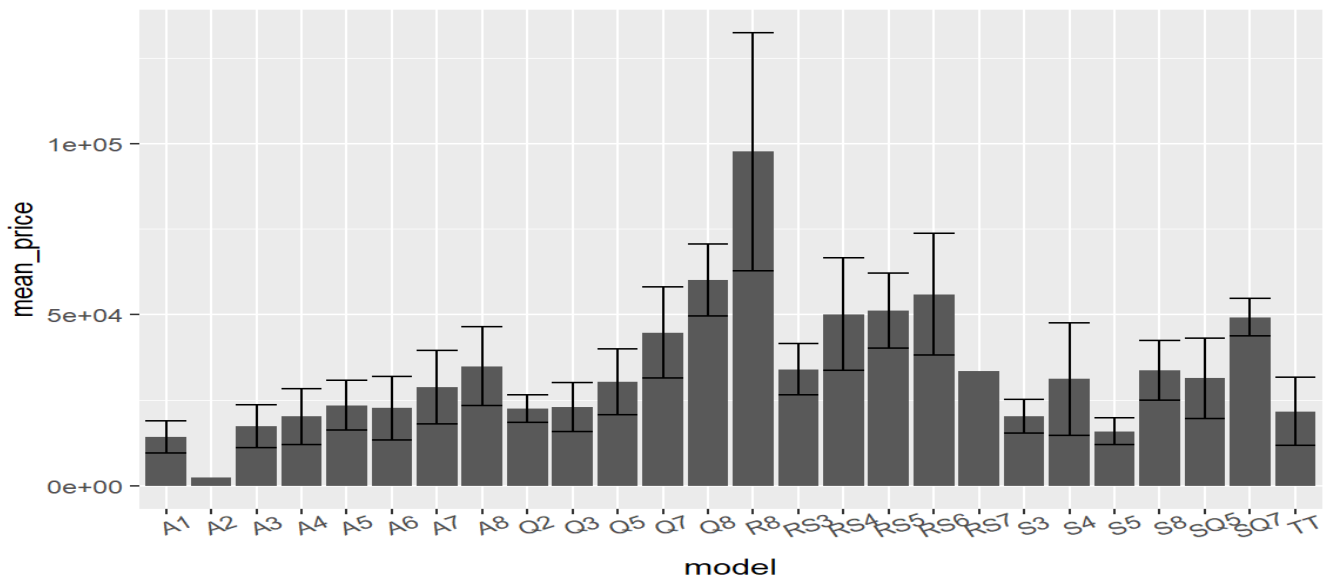
```
3 " A3"    17409.  6231.
```

```
4 " A4"    20255.  8186.
```

```
5 " A5"    23577.  7347.
```

```
6 " A6"    22695.  9228.
```

```
ggplot(mod.pri)+ geom_bar(aes(x= model, y= mean_price), stat = "identity")+ theme(axis.text.x =  
element_text(angle = 25))+ geom_errorbar(aes(x = model, ymin = mean_price - sd_price, ymax =  
mean_price + sd_price))
```



Mean price of model R8 is the highest followed by RS6 in Audi.

R features:

- Plot Type – Bar Median Chart
- Library – ggplot2

Mercedes Model and their prices

```
mod.pri <- MERC[,c("model", "price")]
```

```
head(mod.pri)
```

```
model price
```

```
1 SLK 5200
```

```
2 S Class 34948
```

```
3 SL CLASS 49948
```

```
4 G Class 61948
```

```
5 G Class 73948
```

```
6 SL CLASS 149948
```

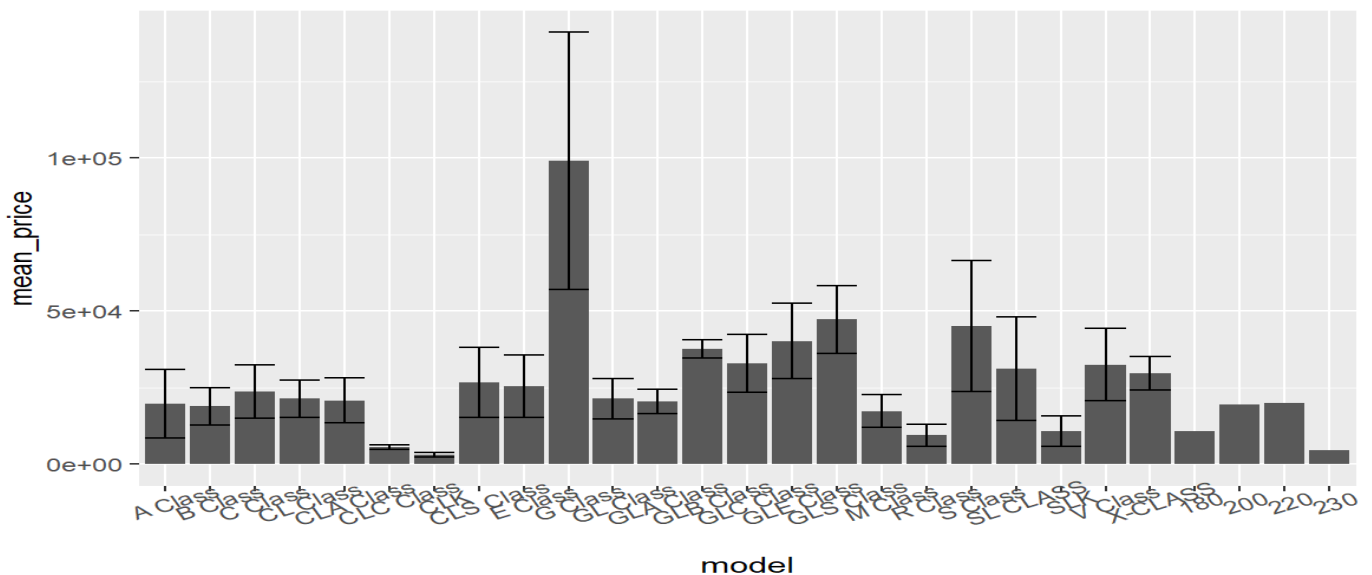
```
mod.pri <- mod.pri %>% group_by(model) %>% summarise(mean_price = mean(price), sd_price =  
sd(price))
```

```
head(mod.pri)
```

```
# A tibble: 6 x 3
```

```
  model    mean_price sd_price
  <chr>      <dbl>    <dbl>
1 " A Class"    19850.  11184.
2 " B Class"    18897.   6003.
3 " C Class"    23696.   8765.
4 " CL Class"   21449.   6083.
5 " CLA Class"  20836.   7229.
6 " CLC Class"   5517.    713.
```

```
ggplot(mod.pri)+ geom_bar(aes(x= model, y= mean_price), stat = "identity")+ theme(axis.text.x =
element_text(angle = 25))+ geom_errorbar(aes(x = model, ymin = mean_price - sd_price, ymax =
mean_price + sd_price))
```



Mean price of model Class E is the highest followed by Class R in Audi.

R features:

- Plot Type – Bar Median Chart
- Library – ggplot2

Numeric Variables associations with Sales – BMW, AUDI and MERC

```
library(dplyr)
```

```
library(tidyr)
```

Attaching package: ‘tidyr’

The following object is masked from ‘package:magrittr’:

```
extract
```

```
num <- select_if(BMW, is.numeric)
```

```
head(num)
```

```
year price mileage tax mpg
```

```
1 2014 11200 67068 125 57.6
```

```
2 2018 27000 14827 145 42.8
```

```
3 2016 16000 62794 160 51.4
```

```
4 2017 12750 26676 145 72.4
```

```
5 2014 14500 39554 160 50.4
```

```
6 2016 14900 35309 125 60.1
```

```
engineSize
```

```
1 2.0
```

```
2 2.0
```

```
3 3.0
```

```
4 1.5
```

```
5 3.0
```

```
6 2.0
```

```
num1 <- num %>% gather()
```

```
head(num1)
```

```
key value
```

```
1 year 2014
```

```
2 year 2018
```

```
3 year 2016
```

```
4 year 2017
```

```
5 year 2014
```

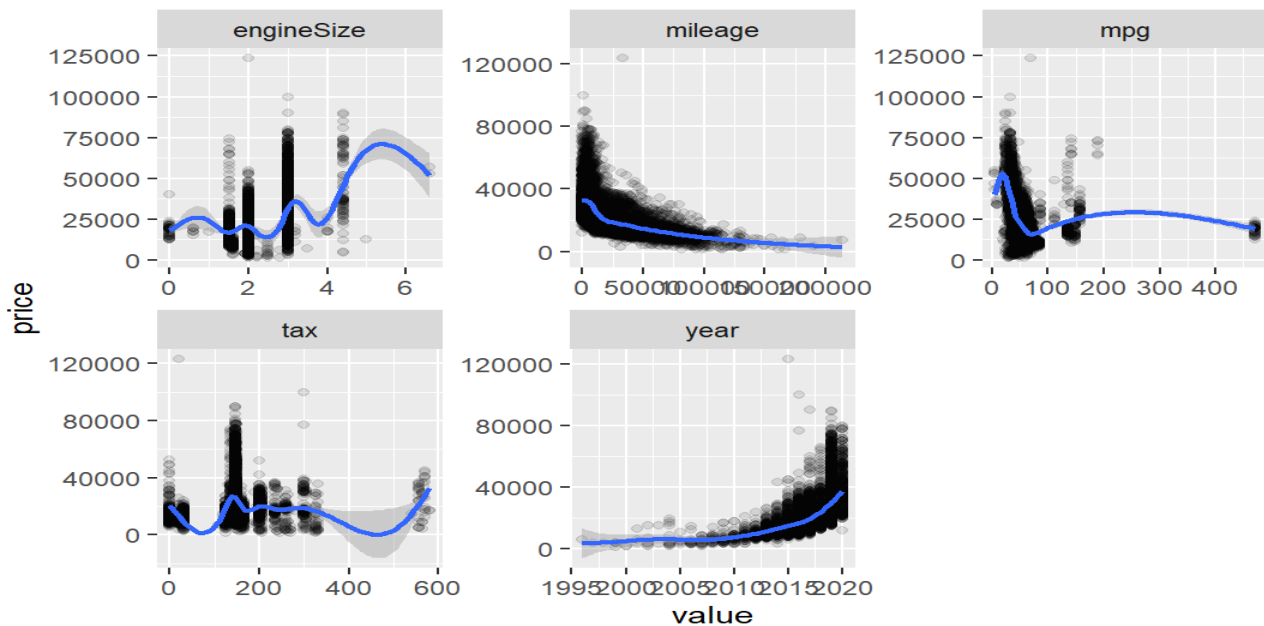
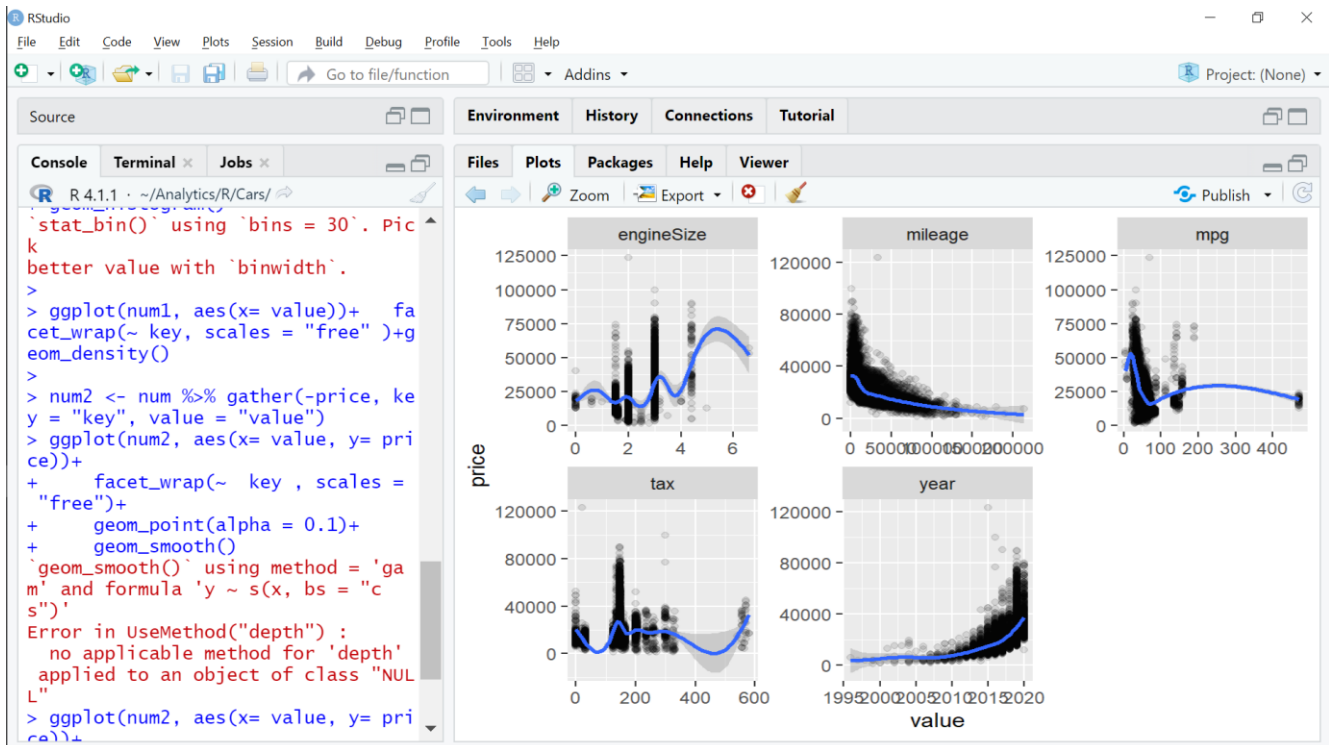
```
6 year 2016
```

```
ggplot(num1, aes(x = value))+ facet_wrap(~ key, scales = "free")+ geom_histogram()
```

```
`stat_bin()` using `bins = 30`.
```

Pick better value with `binwidth`.

```
ggplot(num1, aes(x= value))+ facet_wrap(~ key, scales = "free" )+ geom_density()
ggplot(num2, aes(x= value, y= price))+ facet_wrap(~ key , scales = "free")+geom_point(alpha =
0.1)+geom_smooth()
```



R features:

- Plot Type – Line Chart + Scatter Plot
- Library – ggplot2, tidyr, dplyr

AUDI:

```
num <- select_if(AUDI, is.numeric)
```

```
head(num)
```

```
year price mileage tax mpg
```

```
1 2017 12500 15735 150 55.4
```

```
2 2016 16500 36203 20 64.2
```

```
3 2016 11000 29946 30 55.4
```

```
4 2017 16800 25952 145 67.3
```

```
5 2019 17300 1998 145 49.6
```

```
6 2016 13900 32260 30 58.9
```

```
engineSize
```

```
1 1.4
```

```
2 2.0
```

```
3 1.4
```

```
4 2.0
```

```
5 1.0
```

```
6 1.4
```

```
num1 <- num %>% gather()
```

```
head(num1)
```

```
key value
```

```
1 year 2017
```

```
2 year 2016
```

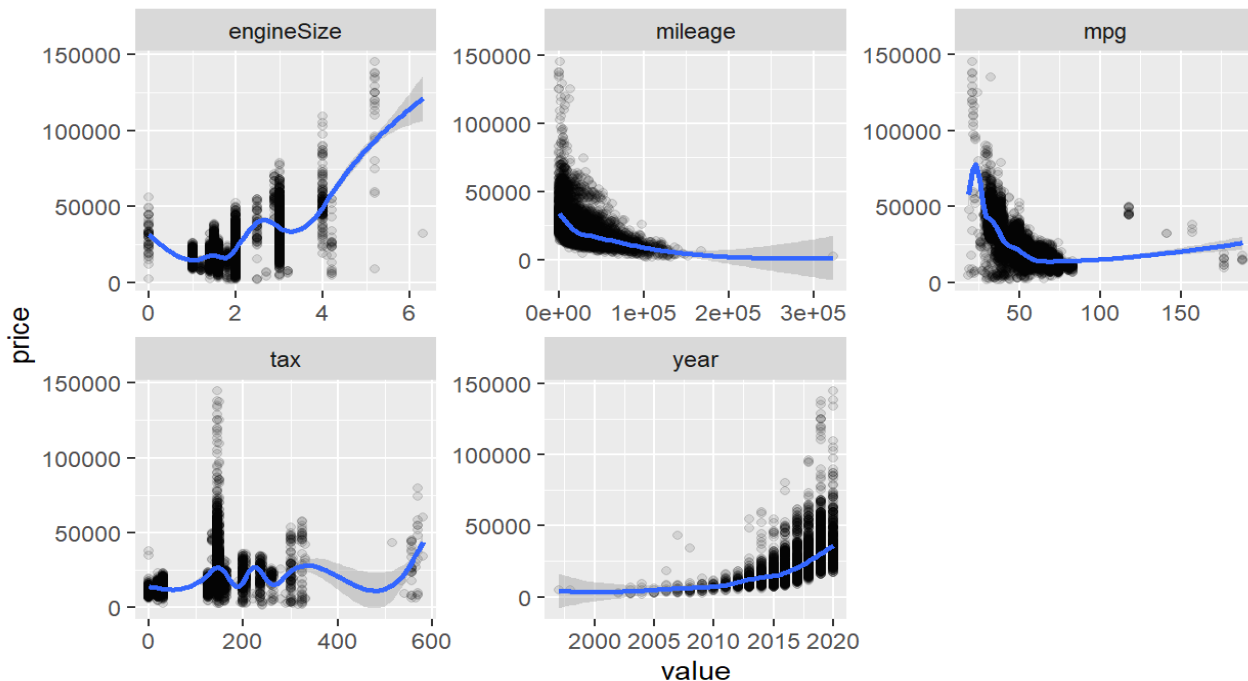
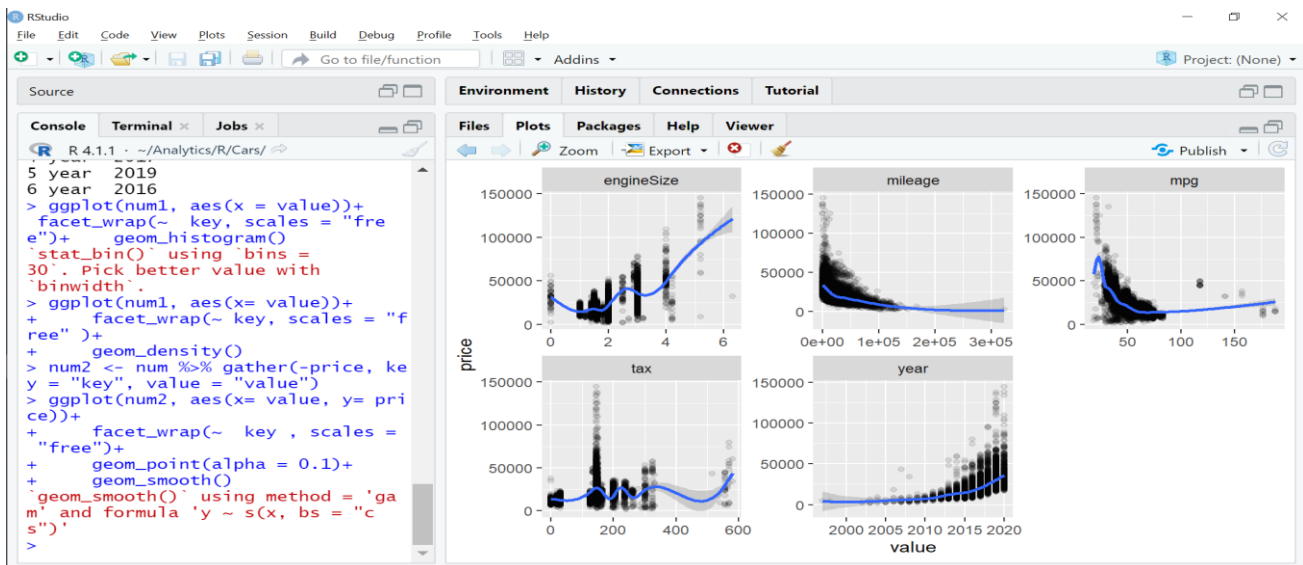
```
3 year 2016
```

```
4 year 2017
```

```
5 year 2019
```

6 year 2016

```
ggplot(num1, aes(x = value))+ facet_wrap(~ key, scales = "free")+ geom_histogram()
ggplot(num1, aes(x= value))+ facet_wrap(~ key, scales = "free" )+ geom_density()
num2 <- num %>% gather(-price, key = "key", value = "value")
ggplot(num2, aes(x= value, y= price))+ facet_wrap(~ key , scales = "free")+ geom_point(alpha =
0.1)+geom_smooth()
```



MERC:

```
num <- select_if(MERC, is.numeric)
```

```
head(num)
```

```

year price mileage tax mpg
1 2005  5200  63000 325 32.1
2 2017 34948  27000  20 61.4
3 2016 49948  6200 555 28.0
4 2016 61948 16000 325 30.4
5 2016 73948  4000 325 30.1
6 2011 149948  3000 570 21.4

```

```
engineSize
```

```

1      1.8
2      2.1
3      5.5
4      4.0
5      4.0
6      6.2

```

```
num1 <- num % % gather()
```

```
head(num)
```

```
ggplot(num1, aes(x = value))+ facet_wrap(~ key, scales = "free")+ geom_histogram()
```

```
`stat_bin()` using `bins =
```

```
30`. Pick better value with
```

```
`binwidth`.
```

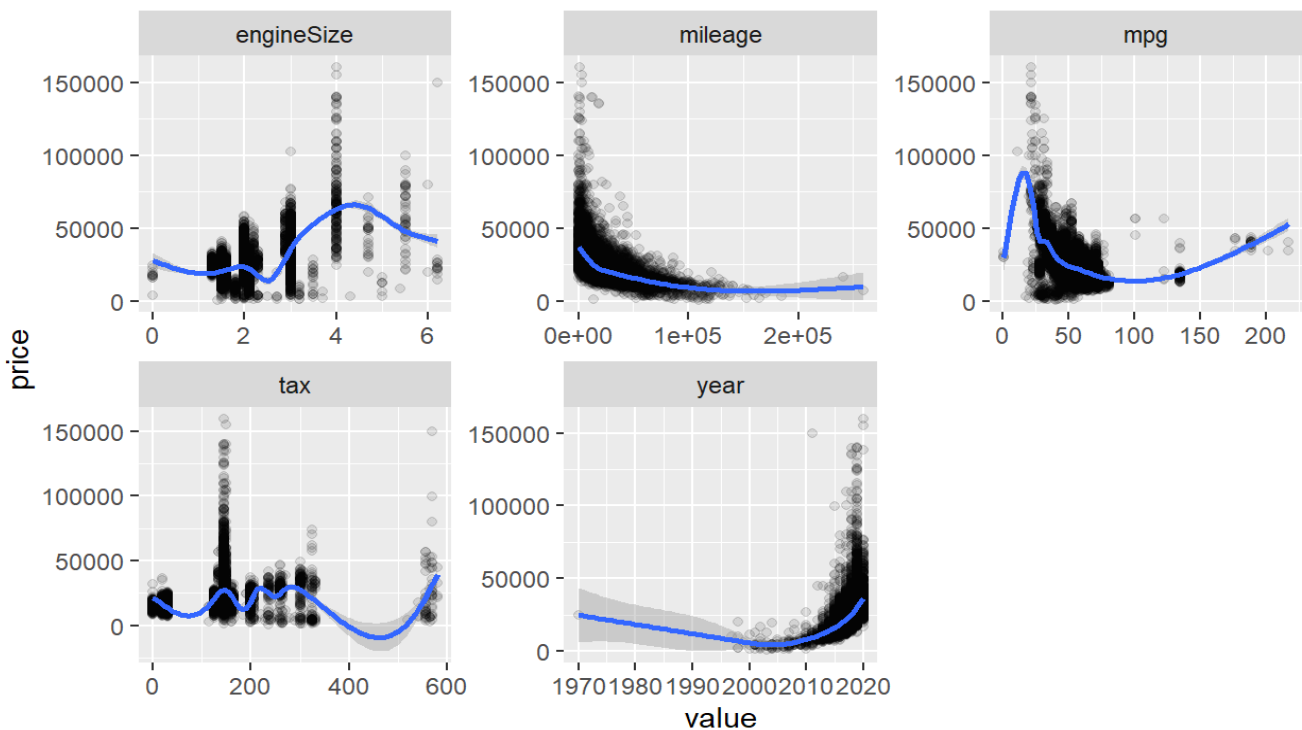
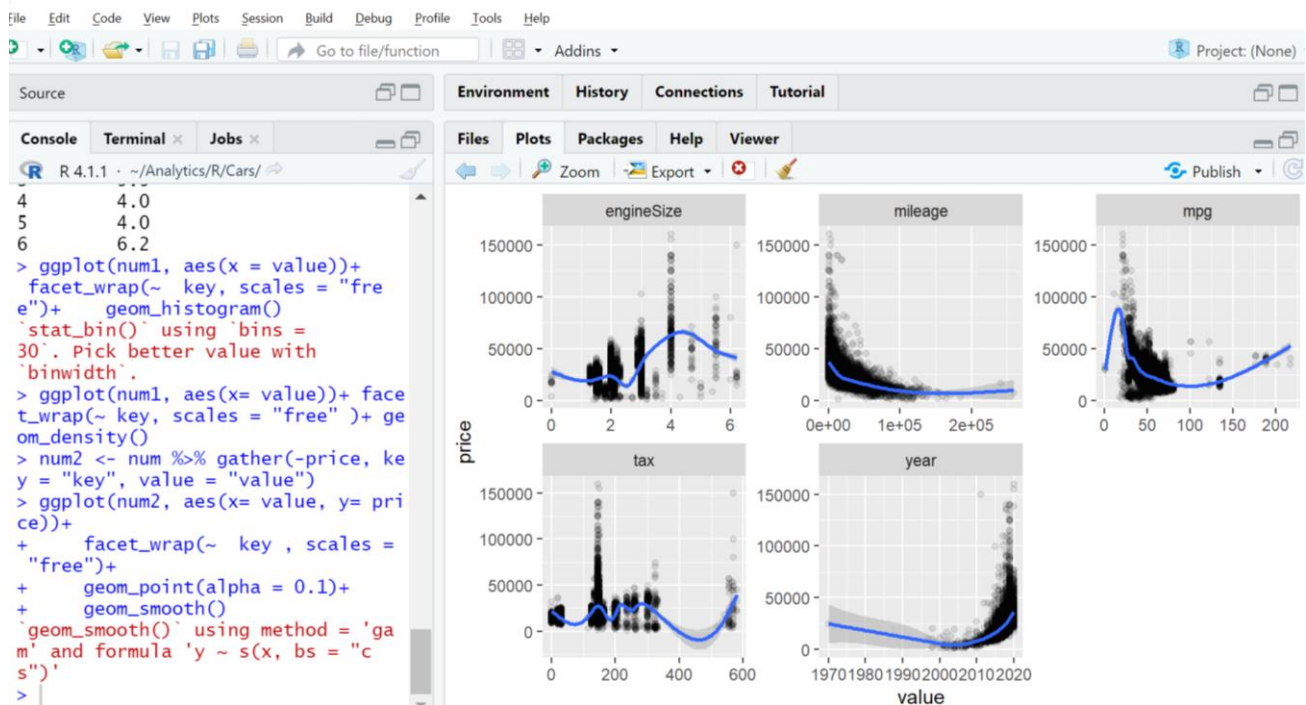
```
ggplot(num1, aes(x= value))+ facet_wrap(~ key, scales = "free" )+ geom_density()
```

```
num2 <- num % % gather(-price, key = "key", value = "value")
```

```

ggplot(num2, aes(x= value, y= price))+ facet_wrap(~ key , scales = "free")+geom_point(alpha =
0.1)+ geom_smooth()

```



SUMMARY -

The Scatterplots show that a linear relation between all the predictor variables and the price variable yields a high bias for the upcoming models. Following is the general briefing of each graph

individually: firstly, the price increases as the size of engine increase. Secondly, there is a decrease in

mileage (mpg) if the engine size is large. For example, the expensive supercars have less mileage as the engine size is huge. So, in short, the smaller the engine size the greater the mileage. The road tax is directly connected with the price. The higher the price of car, the greater road tax needs to be paid (due to expensive auto parts used in it) which leads to the higher value of the car.

The graphs of BMW shows that the engine size increases with the increase in price. The most common engine size used in BMW is 3 which gives the best mileage between 50 to 100mph at the price of approx. 75000\$. The most demanding car with the great value was the 2020 model. Similarly, the graphs of the Audi shows that the price was above 100,000\$ for the engine size greater than 6. But the best mileage was shown between 50 to 100mph when the engine size was between 2 to 4, with the approx. costs between 50,000\$ to 100,000\$, where the road tax was comparatively low for the 2020 model. For Mercedes, the best engine size was 4 with the base price of 150,000\$, which had the exceptional mileage between 100 to 150mph, with low road tax for the later 2020 models. Concluding, Mercedes's 2020 model was the best in terms of Engine Size, Mileage (mpg), Tax and Year as compared to the 2020 model of BMW and Audi.

Correlation Plot – BMW, AUDI and MERC

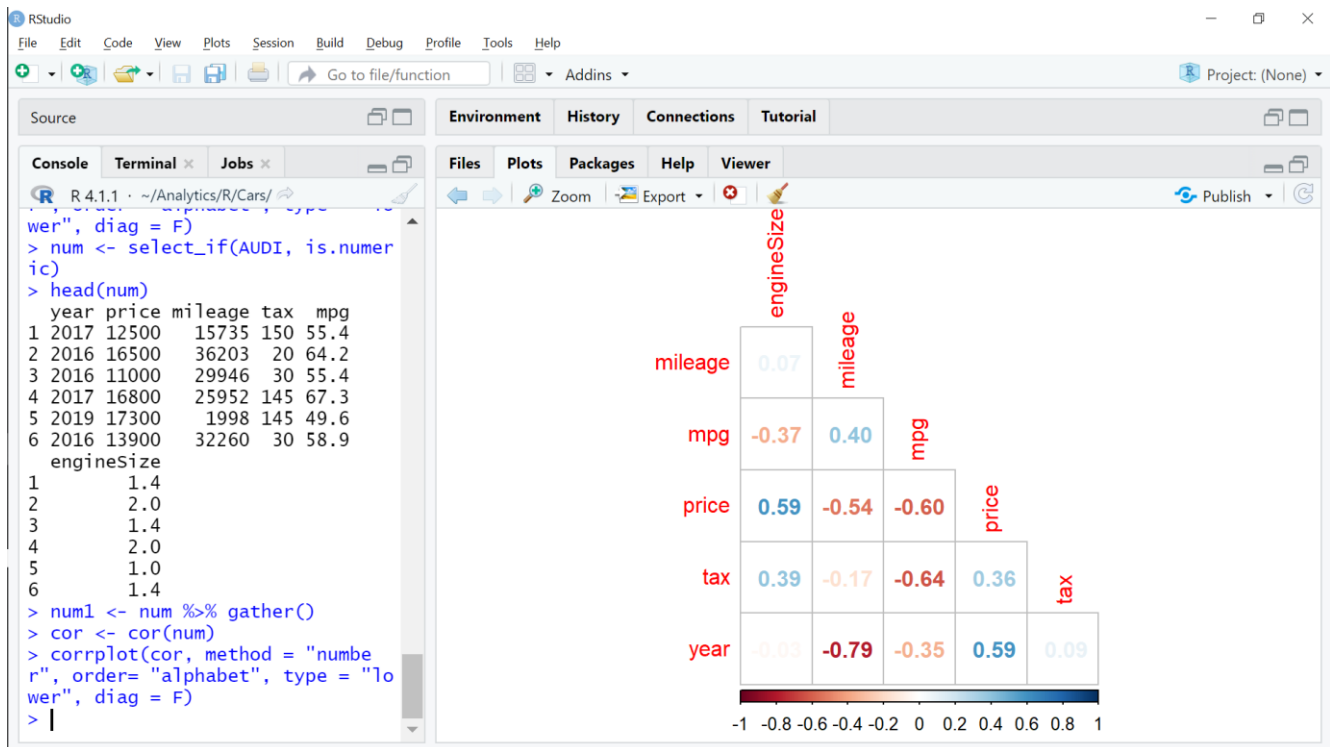
```
cor <- cor(num)

library(corrplot)

corrplot 0.92 loaded

Warning message:
package 'corrplot' was built under R version 4.1.2

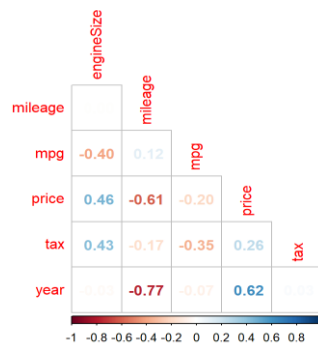
corrplot(cor, method = "number", order= "alphabet", type = "lower", diag = F)
```



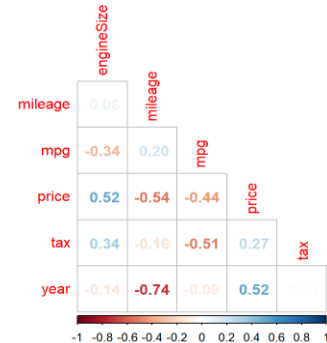
AUDI



BMW



MERC



R features:

- Plot Type – Correlation Plot
- Library – corrplot

SUMMARY

The above correlation plot shows a moderate positive correlation between price and engine size, as well as a high positive correlation between price and registration year. There is a small correlation between the price and tax. In addition to that mileage negatively correlates with engine size. A small negative correlation between price and miles per gallon can be observed to. The high correlation

between mileage and year that are both related to the price indicates a possible multicollinearity for the three cars Audi, BMW and Merc.

Fuel Type – BMW, AUDI and MERC



```
FuelType<- BMW % % group_by(fuelType)
```

```
ft1 <- count(FuelType)
```

```
ft1 <- ft1[, "n"]
```

```
ft.s <- FuelType % % summarise(mean_price = mean(price), sd_price = sd(price))
```

```
ft.s <- ft.s % % cbind(ft1)
```

```
ft.s
```

```
fuelType mean_price sd_price    n
```

```
1 Diesel    21779.26 11194.0751 7027
```

```
2 Electric 18466.00   923.1831     3
```

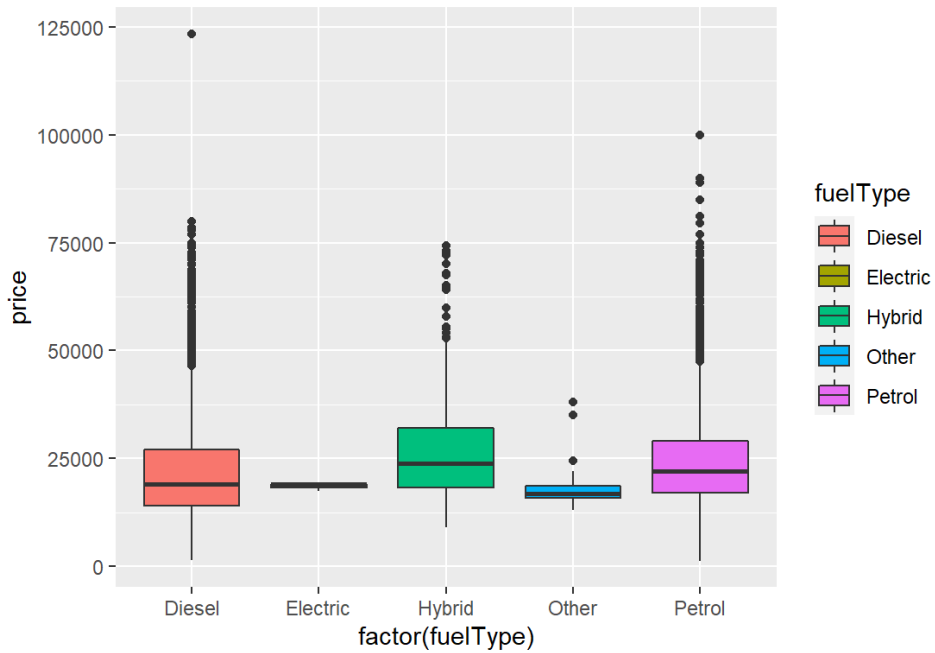
```
3 Hybrid   27169.71 12642.3795   298
```

```
4 Other    18193.86  5054.7885    36
```

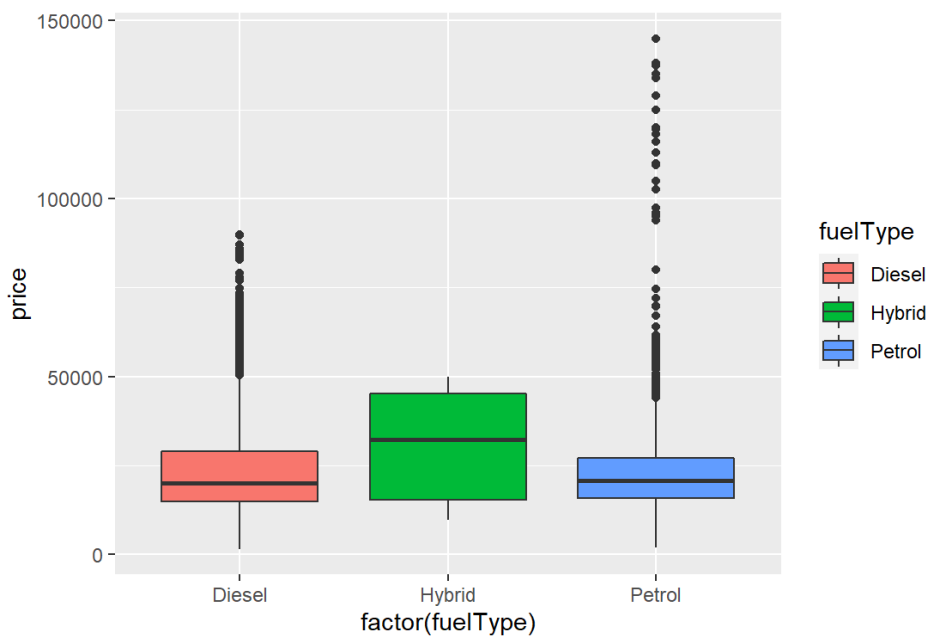
5 Petrol 24360.27 11527.2676 3417

```
ggplot(df, aes(x=factor(fuelType), y= price))+ geom_boxplot(aes(fill = fuelType))
```

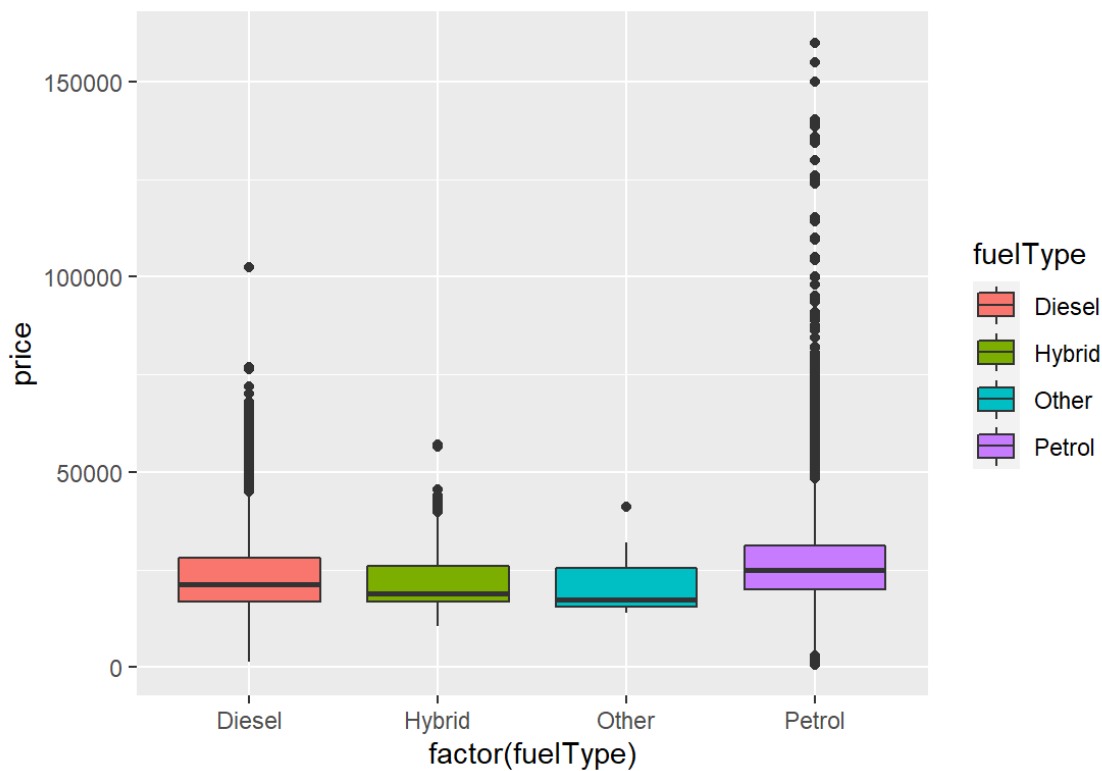
BMW



AUDI



MERC



SUMMARY -

Even though it's hard to see in the boxplot, it seems that Hybrid and Petrol Cars are the most expensive. Followed by Diesel, Electric and Other fuel types.

Note: Just a very small number of cars have the “other” or “electric” fuel type, so our sample for that group is small, which enlarges the standard-error for these estimated variables.

BMW – stands out for Diesel and Hybrid.

Audi – Hybrid

Mercedes – Has high price rise for petrol and the consumption of fuel is similar for all the four categories.

STATISTICAL SUMMARY AND FUNCTIONS

Statistical Summary:

- SCRIPT-

```
- BMW <- read.csv ("bmw.csv")
- AUDI <- read.csv ("audi.csv")
- MERC <- read.csv ("merc.csv")
- any(is.na(BMW))
- any(is.na(AUDI))
- any(is.na(MERC))
- BMW$model % % unique()
- AUDI$model % % unique()
- MERC$model % % unique()
- library(tidyverse)
- library(corrplot)
- install.packages("ggplot")
- library(ggplot2)
- ggplot(BMW, aes(x= model))+ geom_bar()+ theme(axis.text.x =
  element_text(angle = 25))
- ggplot(AUDI, aes(x= model))+ geom_bar()+ theme(axis.text.x =
  element_text(angle = 25))
- ggplot(MERC, aes(x= model))+ geom_bar()+ theme(axis.text.x =
  element_text(angle = 25))
- mod.pri <- BMW[,c("model", "price")]
- head(mod.pri)
- mod.pri <- mod.pri % % group_by(model) % % summarise(mean_price =
  mean(price), sd_price = sd(price))
- head(mod.pri)
- ggplot(mod.pri)+ geom_bar(aes(x= model, y= mean_price), stat = "identity")+
  theme(axis.text.x = element_text(angle = 25))+ geom_errorbar(aes(x = model,
  ymin = mean_price - sd_price, ymax = mean_price + sd_price))
- mod.pri <- AUDI[,c("model", "price")]
- head(mod.pri)
- ggplot(mod.pri)+ geom_bar(aes(x= model, y= mean_price), stat = "identity")+
  theme(axis.text.x = element_text(angle = 25))+ geom_errorbar(aes(x = model,
  ymin = mean_price - sd_price, ymax = mean_price + sd_price))
- mod.pri <- MERC[,c("model", "price")]
- head(mod.pri)
- mod.pri <- mod.pri % % group_by(model) % % summarise(mean_price =
  mean(price), sd_price = sd(price))
- head(mod.pri)
- ggplot(mod.pri)+ geom_bar(aes(x= model, y= mean_price), stat = "identity")+
  theme(axis.text.x = element_text(angle = 25))+ geom_errorbar(aes(x = model,
  ymin = mean_price - sd_price, ymax = mean_price + sd_price))
- library(dplyr)
```

```

- library(tidyr)
- num <- select_if(BMW, is.numeric)
- head(num)
- num1 <- num % % gather()
- head(num1)
- ggplot(num1, aes(x = value))+ facet_wrap(~ key, scales = "free")+
geom_histogram()
- ggplot(num1, aes(x= value))+ facet_wrap(~ key, scales = "free" )+
geom_density()
- ggplot(num2, aes(x= value, y= price))+ facet_wrap(~ key , scales =
"free")+geom_point(alpha = 0.1)+geom_smooth()
- num <- select_if(AUDI, is.numeric)
- head(num)
- num1 <- num % % gather()
- head(num1)
- ggplot(num1, aes(x = value))+ facet_wrap(~ key, scales = "free")+
geom_histogram()
- ggplot(num1, aes(x= value))+ facet_wrap(~ key, scales = "free" )+
geom_density()
- num2 <- num % % gather(-price, key = "key", value = "value")
- ggplot(num2, aes(x= value, y= price))+ facet_wrap(~ key , scales =
"free")+ geom_point(alpha = 0.1)+geom_smooth()
- num1 <- num % % gather()
- head(num)
- ggplot(num1, aes(x = value))+ facet_wrap(~ key, scales = "free")+
geom_histogram()
- ggplot(num1, aes(x= value))+ facet_wrap(~ key, scales = "free" )+
geom_density()
- num2 <- num % % gather(-price, key = "key", value = "value")
- ggplot(num2, aes(x= value, y= price))+ facet_wrap(~ key , scales =
"free")+geom_point(alpha = 0.1)+ geom_smooth()
- cor <- cor(num)
- library(corrplot)
- corrplot(cor, method = "numberss", order= "alphabet", type = "lower", diag
= F)
- FuelType<- BMW % % group_by(fuelType)
- ft1 <- count(FuelType)
- ft1 <- ft1[, "n"]
- ft.s <- FuelType % % summarise(mean_price = mean(price), sd_price =
sd(price))
- ft.s <- ft.s % % cbind(ft1)
- ft.s
- ggplot(df, aes(x=factor(fuelType), y= price))+ geom_boxplot(aes(fill =
fuelType))

```

SUMMARY-

1. BMW -

- Most common model 3 series.
- M5 model with the highest mean price.
- Through scatter plot we were able to understand aspects like:
engine size, mileage, mph and price .

Engine size increases with the increase in price. The most common engine size used in BMW is 3 which gives the best mileage between 50 to 100mph at the price of approx. 75000\$. The most demanding car with the great value was the 2020 model.

BMW – stands out for Diesel and Hybrid fuel type.

2. AUDI -

- Most common model A3.
- Model R8 with the highest mean price.
- Through scatter plot we were able to understand aspects like:
engine size, mileage, mph and price .

Audi shows that the price was above 100,000\$ for the engine size greater than 6. But the best mileage was shown between 50 to 100mph when the engine size was between 2 to 4, with the approx. costs between 50,000\$ to 100,000\$, where the road tax was comparatively low for the 2020 model.

Audi- Hybrid fuel type

3. MERC -

- Most common model C Class.
- Class E model with the highest mean price.
- Through scatter plot we were able to understand aspects like:
engine size, mileage, mph and price .

For Mercedes, the best engine size was 4 with the base price of 150,000\$, which had the exceptional mileage between 100 to 150mph, with low road tax for the later 2020 models.

Mercedes – Has high price rise for petrol and the consumption of fuel is similar for all the four categories.

Concluding, Mercedes's 2020 model was the best in terms of Engine Size, Mileage (mpg), Tax and Year as compared to the 2020 model of BMW and Audi.

Functions Used:

1. During data cleaning:

- a. `Str()` –used for compactly exhibiting the inner structure of a R object. It can show even the core structure of huge lists which are nested. It delivers one liner output for the elementary R objects letting the user know about the thing and its elements.⁵
- b. `Any(is.na())` – will tell if you if there are ANY of the given search terms in your vector. It returns either TRUE or FALSE.⁶
- c. `Unique ()` – returns a vector, data frame or array with duplicate elements/rows removed.

2. Most Common Model:

- a. `Ggplot()` – initializes a ggplot object. It can be used to state the input data frame for a graphic and to require the set of plot aesthetics proposed to be common during all following layers unless overridden.⁷
- b. `aes()` – creates a list of unevaluated terms. This function also achieves partial name identical, converts color to color, and old-style R names to ggplot names.⁸

⁵ [https://www.geeksforgeeks.org/display-the-internal-structure-of-an-object-in-r-programming-str-function/#text=str\(\)%20function%20in%20R,the%20object%20and%20its%20constituents](https://www.geeksforgeeks.org/display-the-internal-structure-of-an-object-in-r-programming-str-function/#text=str()%20function%20in%20R,the%20object%20and%20its%20constituents).

⁶ <https://study.com/academy/lesson/any-and-all-functions-in-r-programming.html>

⁷ <https://www.rdocumentation.org/packages/ggplot2/versions/3.3.5/topics/ggplot>

⁸ <https://www.rdocumentation.org/packages/ggplot2/versions/1.0.0/topics/aes>

- c. `theme()` – allows you to override the theme elements by calling element functions, like `theme(plot.title = element_text(colour = "red"))`.⁹
- d. `geom_bar()` – makes the height of the bar proportional to the number of cases in each group.

3. Model and their Prices:

- a. `Head()` - `head()` function in R Language is used to get the initial parts of a vector, matrix, table, data frame or function.¹⁰
- b. `group_by()` – groups the data frame by several columns with mean, sum and other functions like count, max and min.
- c. `summarize()` – used on grouped data created by `group_by()`. The output will have one row for each group. Reduces multiple values down to a single value¹¹
- d. `mean()` – calculate the mathematics mean of the elements of the vector passed to it as argument.
- e. `sd()` – standard deviation of given values in R. It is the square root of its variance.

4. Numeric Variables:

- a. `select_if()` – a predicate on the columns of data frame for which the predicate returns TRUE will be selected.¹²
- b. `facet_wrap()` –wraps a 1d sequence of panels into 2d, this helps in better use of screen space than `facet_grid()` because most shows are unevenly rectangular.¹³
- c. `geom_density()` – smoothed version of histogram.
- d. `geom_point()` – adds layer of points to your plot, which creates a scatterplot.

⁹ <https://ggplot2-book.org/polishing.html>

¹⁰ <https://www.geeksforgeeks.org/get-the-first-parts-of-a-data-set-in-r-programming-head-function/>

¹¹ <https://www.rdocumentation.org/packages/dplyr/versions/0.7.8/topics/summarise>

¹² https://www.rdocumentation.org/packages/dplyr/versions/0.5.0/topics/select_if

¹³ https://www.rdocumentation.org/packages/ggplot2/versions/3.3.5/topics/facet_wrap

- e. `geom_smooth()` – add smoothed conditional means / regression line.
- f. `gather()` – function may not be clear what accurately is going on, but in this case we essentially have a lot of column names that signify what we would like to have as data values.¹⁴

5. Correlation Plot:

- a. `corrplot()` – has about 50 parameters, mostly common ones are only a few. We get a correlation matrix plot with only one line of code in most cases. The most used parameters include `method`, `type`, `order`, `diag`, and etc.¹⁵

6. Fuel Type :

- a. User Defined Function – `Fueltype()` – Used it to take out standard deviation and mean of the price and summarized it with the fuel type to get the most consumed fuel for the car.
- b. `Count()` – count the unique values of one or more variables.
- c. `cbind()` – combine specified Vector, Matrix or Data Frame by columns.

¹⁴ <http://statseducation.com/Introduction-to-R/modules/tidy%20data/gather/>

¹⁵ <https://cran.r-project.org/web/packages/corrplot/vignettes/corrplot-intro.html>