STATISTICS PROJECT

Research question : How much do cars cost in Istanbul based on certain features ?

Done by :

Ali alhaj - 2018456

Omnia Elmenshawy - 2000007

Rayyan al-haj – 2017741

Mohammed adnan – 2018383

Abstract

In this project, we study a linear regression model to predict the car prices. this dataset consists of information cars listed on Sahibinden. We try to predict car prices between used and new and putting in some factors such as the brand, fuel it takes, how many kilometers it has passed, the year it was created, its case type (weather its an suv,sedan,etc..),the gear, the color and obviously the model. In this report we used a number of statistical ways to analyze the dataset that we collected. Some of which include diagrams, test of normality, sampling, point estimation, confidence intervals, correlation, data visualization, hypotheses tests, anova test, goodness of fit test and last but not least we pre-processed the dataset in order to be able to use linear regression model to predict car prices in Istanbul.

Table of content

- Data collection
- Exploratory data analysis
- Normality test
- Checking outliers
- Confidence interval and point estimation
- Anova test
- Goodness of fit test
- Hypothesis test
- Correlation
- Linear regression model and evaluation
- Conclusion
- Source Code

Data collection:

For starters, we need to have a look at the dataset and understand its size, attribute names and the dataType. The following features are shown in the picture attached below. By collecting the data manually from sahibinden, we have collected 197 rows of data with 10 columns of features.

```
> head(df)
# A tibble: 6 x 10
  brand      model                  year status fuel   gear           km case_type color   price
  <chr>      <chr>                 <dbl>  <dbl> <chr>  <chr>       <dbl> <chr>     <chr>   <dbl>
1 ford       ranger 2.2 TDCi XLT    2018      1 diesel Automatic       0 pickup    smoked 859900
2 audi       q 7 3.0 TDI Quattro    2011      2 diesel Automatic  259000 suv       black  810000
3 ford       ranger 2.2 TDCi XLT    2017      1 diesel Automatic       0 pickup    blue   808000
4 ford       ranger 2.2 TDCi XLT    2017      1 diesel Automatic       0 pickup    black  790000
5 volkswagen cherokee 2.0 TD Limited 2016     1 diesel Automatic       0 suv       white  789950
6 ford       ranger 2.2 TDCi XLT    2017      1 diesel Automatic       0 pickup    white  786000
>
```

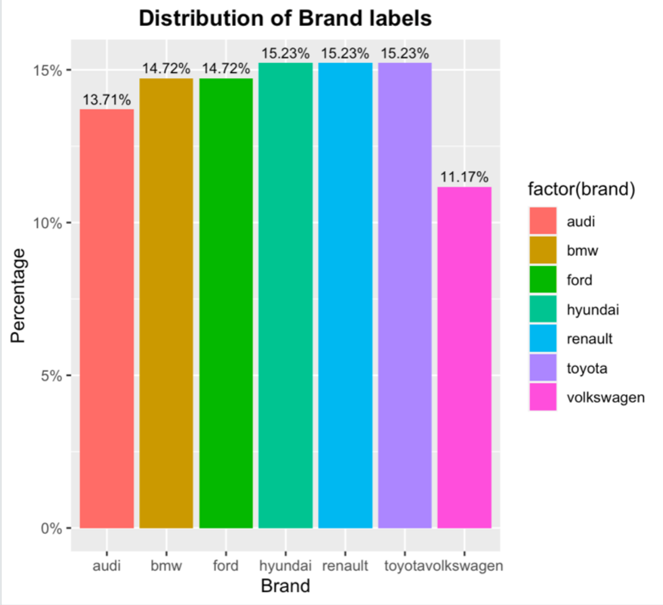
Exploratory data analysis:

The data has 0 null values and 0 duplicated rows and the Balance of the categorical columns is as follows:

```
#2- Brand

common_theme <- theme(plot.title = element_text(hjust = 0.5, face = "bold"))

ggplot(data = df, aes(x = factor(brand),
                      y = prop.table(stat(count)), fill = factor(brand),
                      label = scales::percent(prop.table(stat(count))))) +
  geom_bar(position = "dodge") +
  geom_text(stat = 'count',
            position = position_dodge(.9),
            vjust = -0.5,
            size = 3) +
  scale_y_continuous(labels = scales::percent)+
  labs(x = 'Brand', y = 'Percentage') +
  ggtitle("Distribution of Brand labels") +
  common_theme
```

1. **Brand Column:**

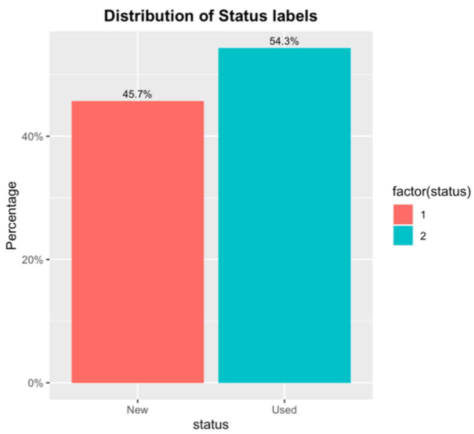
```
audi        bmw      ford    hyundai    renault     toyota volkswagen
  27         29        29         30         30         30         22
```



The column was randomly selected, and some brand names appeared more than others such as hyundai and toyota.

2. **Status column:**

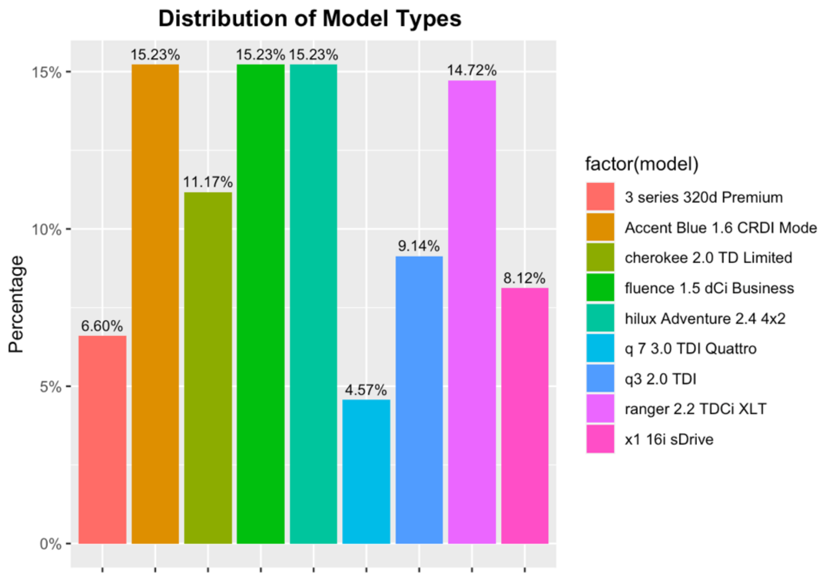
```
 1   2
90 107
```



The value 1 in this column indicates the new cars, and the value 2 indicates the used ones. This column has a small rate of inbalance as 45.7% of the data has the value of 1.

3. Model column:

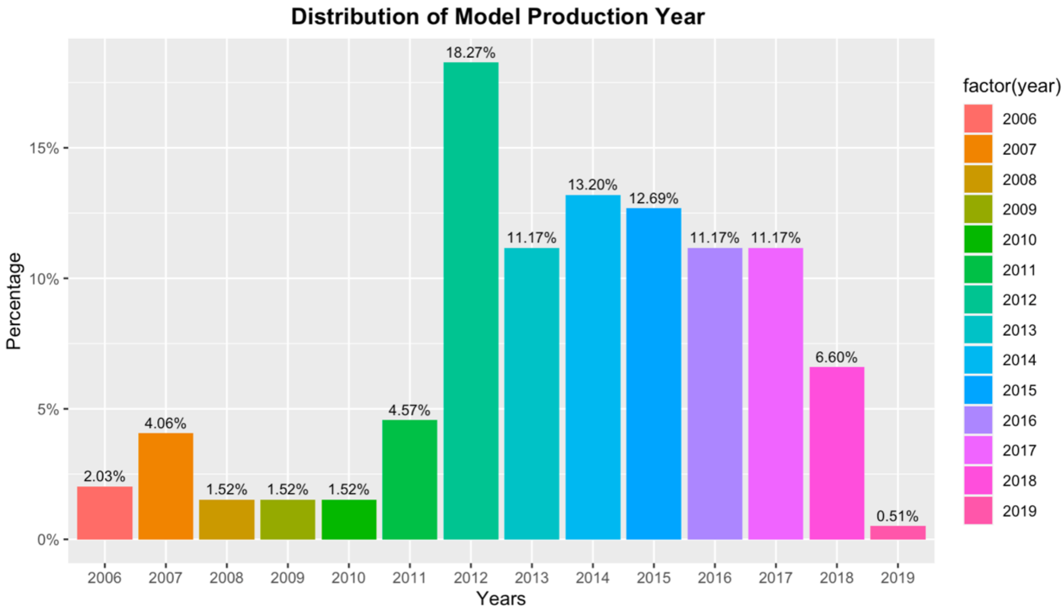
```
> table(df$model)

     3 series 320d Premium Accent Blue 1.6 CRDI Mode     cherokee 2.0 TD Limited 
                        13                        30                        22 
 fluence 1.5 dCi Business     hilux Adventure 2.4 4x2       q 7 3.0 TDI Quattro 
                        30                        30                         9 
               q3 2.0 TDI         ranger 2.2 TDCi XLT             x1 16i sDrive 
                        18                        29                        16 
```



This column is an unbalanced column, some car models doesn't have high frequency in the data such as the q 7 3 TDT Quattro which only exists 9 times in the data, however, some other values such as the Accent Blue 1.6 CRDT Mode exists 30 times.

4. Year Column:

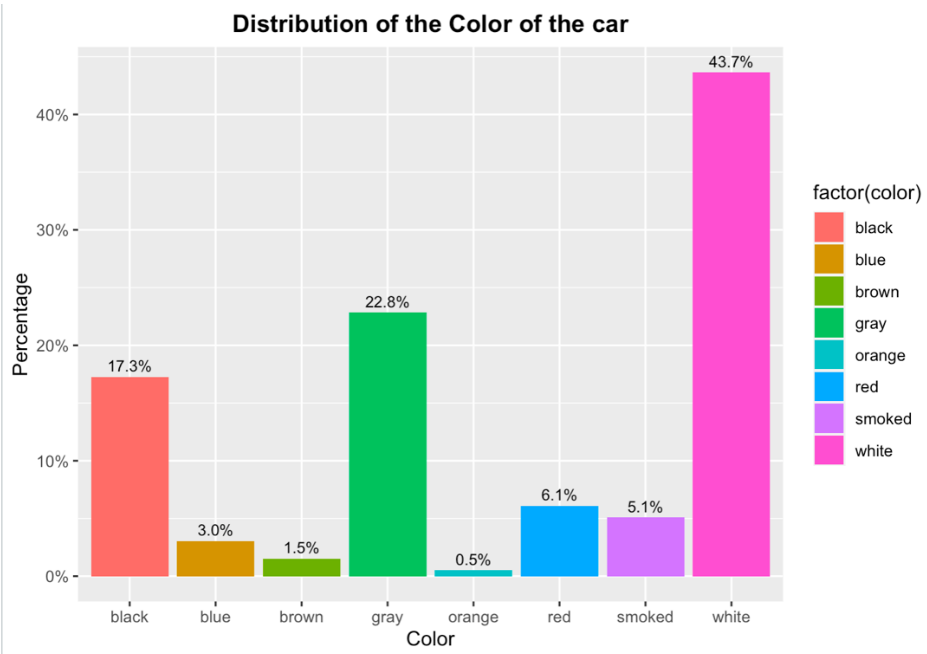
```
2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 
   4    8    3    3    3    9   36   22   26   25   22   22   13    1 
```



This column is highly unbalanced and is highly skewed as some years such as 2008 and 2009 exists only 3 times in the dataset while other years such as 2012 exsits 36 times.

5. Color column:

```
> table(df$color)

 black   blue  brown   gray orange    red smoked  white 
    34      6      3     45      1     12     10     86 
> 
```

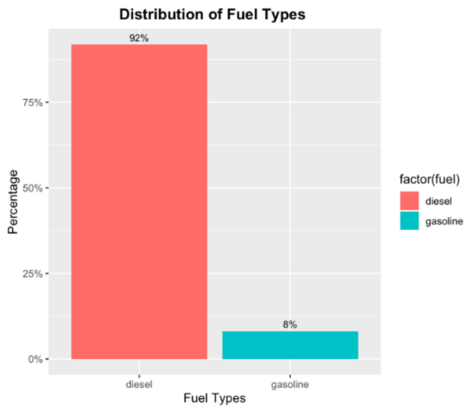


This column is also highly unbalanced as some colors such as the orange one exists only 1 time in the dataset while other colors such as the white color exsits 86 times.

Thus, the data is randomly collected, and the source of the data is most probably including more white cars than any other color

6. Fuel column :

```
> table(df$fuel)

 diesel gasoline
    181       16
>
```



This column is highly unbalanced, and it could cause misclassification problems and type1 and type2 errors in any classification task, it needs to be preprocessed and fixed.

Statistical Analysis:

1. Statistical Description of the Price (Target) column:

```
> stat.desc(df$price)
     nbr.val     nbr.null       nbr.na          min          max        range          sum       median
1.970000e+02 0.000000e+00 0.000000e+00 5.550000e+04 8.599000e+05 8.044000e+05 9.091082e+07 4.990000e+05
        mean      SE.mean CI.mean.0.95          var      std.dev     coef.var
4.614762e+05 1.405964e+04 2.772760e+04 3.894169e+10 1.973365e+05 4.276200e-01
>
```

Our standard deviation is high, with a value of 197336.5, which means that the data points are far away from each other and has a wide range of values and this is due to the big difference of the car prices from brand to brand and from model to model.

The mean and median of the data are quite close to each other which means that the data might be normally distributed or might have multimodal shape, and the distributon is also negatively skewed as the median is higher than the mean.

```
     price
 Min.   : 55500
 1st Qu.:267900
 Median :499000
 Mean   :461476
 3rd Qu.:620000
 Max.   :859900
```

from the summary, we conclude that our data has 197 non null values, 4 numerical double type values, and 6 categorical string values.Our target column - Price - has some outliers as the max and min values are far away from the median and mean points.

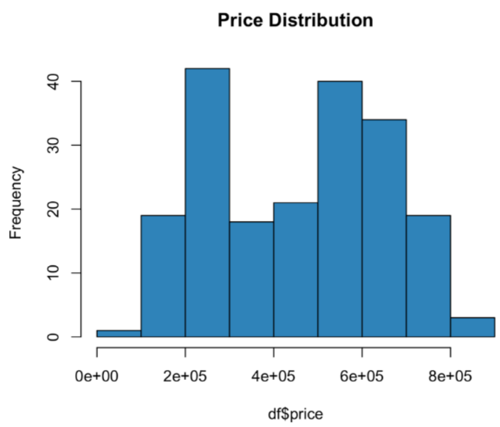
Coefficient of variation:

```
> print(" Price Coefficient of variation is: ")
[1] " Price Coefficient of variation is: "
> sd(df$price) / mean(df$price)
[1] 0.42762
```

Our CV = 0.42762 which means that our price distribution is not high and not small it is in the middle.

Normality Tests:

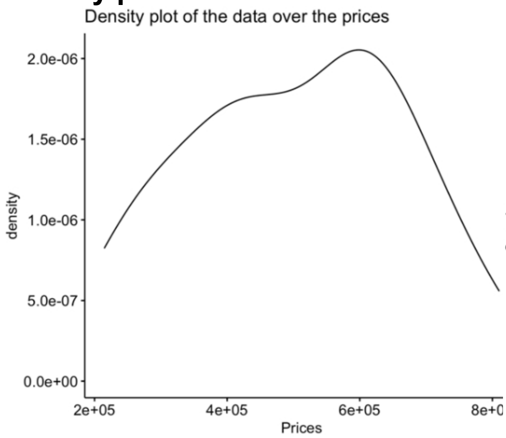
1. **Histogram of price:**

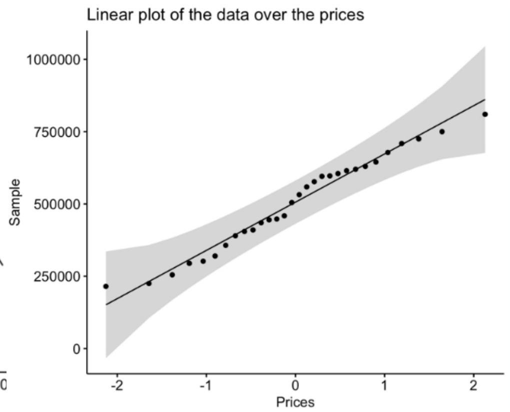


The histogram of the price (target) column states that our data is almost following a biomodal distribution as two price ranges were frequently repeated and used near to the 200k and the 600k Turkish liras.

Let us visualize it more with the following plots.

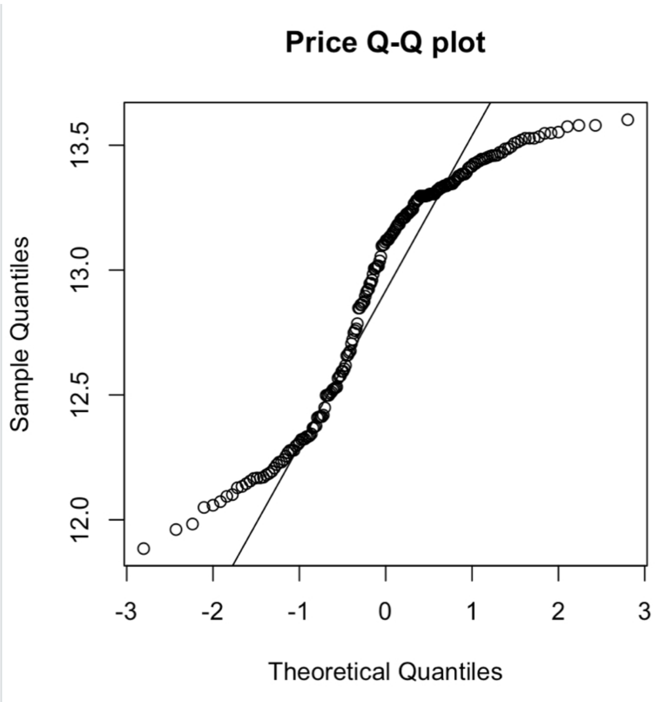
2. **density plot:**





The density plot and linear plot shows that the data is close to the biomodal shape as there are two independenent separate peaks in the population distribution of the price column.

3. <u>**Q-Q plot of the price column:**</u>



This plot clearly show us that the data is not normally distributed because the plot points deviate significantly from a spread diagonal line.

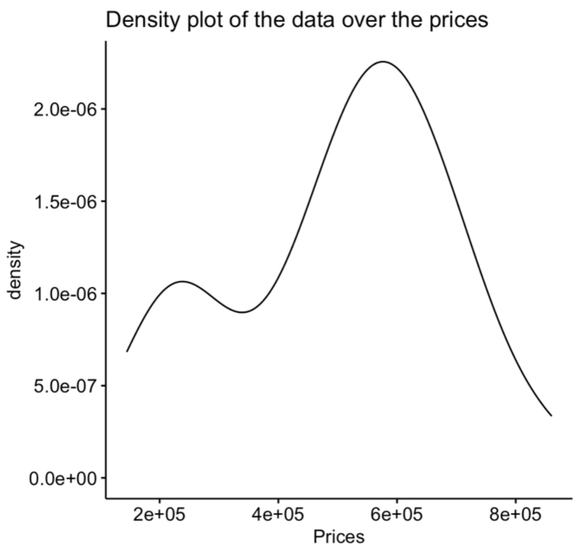
4. Shapiro-Wilk Test and Kolmogorov-Smirnov test

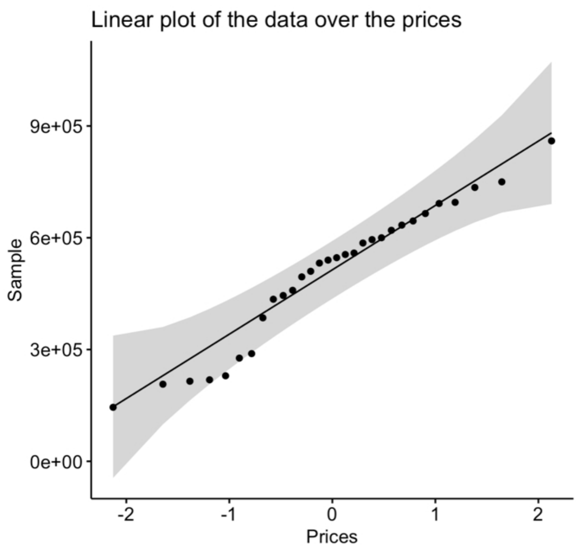
```
> shapiro.test(df2$price)

	Shapiro-Wilk normality test

data:  df2$price
W = 0.96959, p-value = 0.5281
```

```
> ks.test(df2$price, 'pnorm')

	One-sample Kolmogorov-Smirnov test

data:  df2$price
D = 1, p-value = 1.221e-15
alternative hypothesis: two-sided
```

From the P-value of this test we reject the null hypothesis and we conclude that the data column is not normally distributed.

5. **<u>Random sampeling with number of samples = 30 out of 198:</u>**

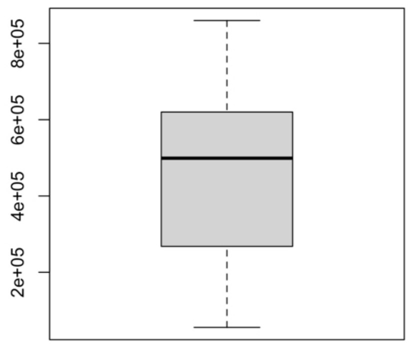




With n = 30, the density and linear plot shows that the data is again not normally distributed and thet it follow as biomodal shape distribution.

Outliers and boxplots:

1. **Price column boxplot:**



Median and mean are close, and some outliers might occur.

Discovering outliers by the the upper and lower bound method from the first and third quartile:

Lower bound = 1% is lower than 156040 , Upper Bound = 99% is upper than 808080,

Thus, any value lower than or upper than this bound is considered as an outlier as follows:

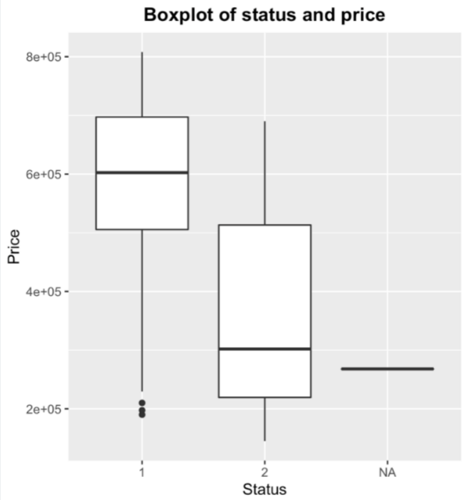
```
> print("Rows with Outliers are: ")
[1] "Rows with Outliers are: "
> df[outlier_ind, ]
# A tibble: 4 × 10
  brand   model                    year status fuel   gear          km case_type color   price
  <chr>   <chr>                   <dbl>  <dbl> <chr>  <chr>      <dbl> <chr>     <chr>   <dbl>
1 ford    ranger 2.2 TDCi XLT      2018      1 diesel Automatic      0 pickup    smoked 859900
2 audi    q 7 3.0 TDI Quattro      2011      2 diesel Automatic 259000 suv       black  810000
3 renault fluence 1.5 dCi Business 2012      2 diesel Manual    245000 sedan     gray   145000
4 toyota  hilux Adventure 2.4 4x2  2017      1 diesel Automatic      0 pickup    white   55500
>
```

So, we substituted the values below the lower bound with the first quartile value, and also substituted the values above the upper bound with the third quartile in order to avoid dropping any of the data raws and use the full dataset.

And here is the new min and max values of the price column after handelling the outliers:

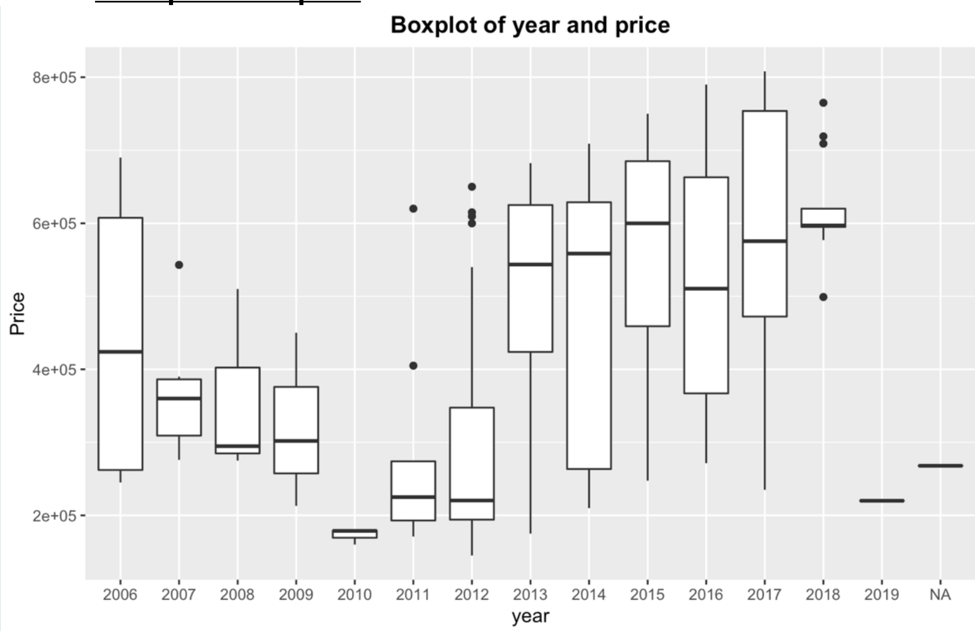
```
> summary(df$price)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 145000  267900  497000  459400  619250  808000
```

2. **<u>Status and price boxplot:</u>**



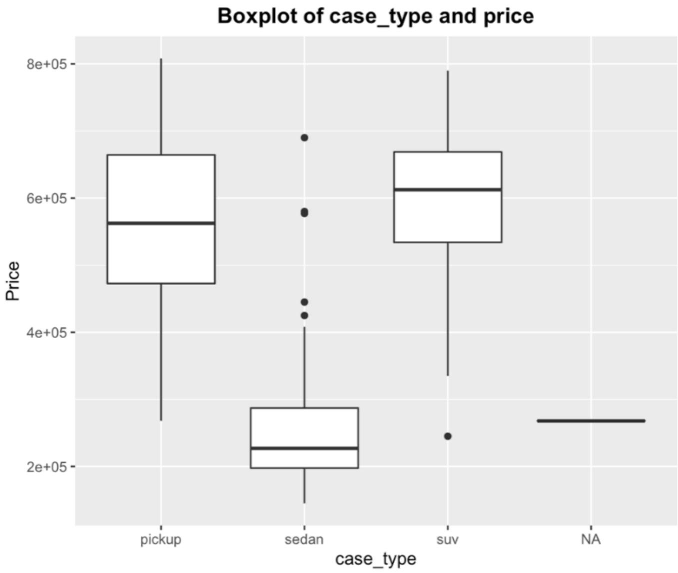
Some low value outliers exist in the cars with a stuatus of value 1, which means that some new cars prices are lower than expected.

3. **<u>Year price boxplot:</u>**



Some car prices are reletevaly higher than expected when compared to their model production year, such as some prices in the year of 2021.

4. **<u>Case_type and price boxplot:</u>**



The sedan type has some high value prices than expected which could be considrerd as outliers, also the suv has a low value price outlier.

Point estimations and confidence intervals:

Goodness of fit using Chi Square Test on the price column:

```
> chisq <- chisq.test(df$price)
> chisq

        Chi-squared test for given probabilities

data:  df$price
X-squared = 16539466, df = 196, p-value < 2.2e-16
```

We got a P-value less than the significant value of 0.05, and we got a very large chi square value which means that our sample data does not fit the population data in na apropriate way, thus they are dependent.

```
# Point estimations and confidence intervals:
  # sample mean as point estimation:
mean(df$price, na.rm = TRUE)
  # Sample size, std, margin
n <- 197
xbar <- mean(df$price, na.rm = TRUE)
s <- sd(df$price)
margin <- qt(0.975, df=n-1)*s/sqrt(n)
low <- xbar - margin
print(low)
high <- xbar + margin
print(high)
k <- sum(df$price < 240000)
p <- k/n
print("95% confidence interval, and estimate th
print(p)
```

```
> print(p)
[1] 0.213198
```

Anova test

first we take a random sample, here we take a random sample of 30 rows.

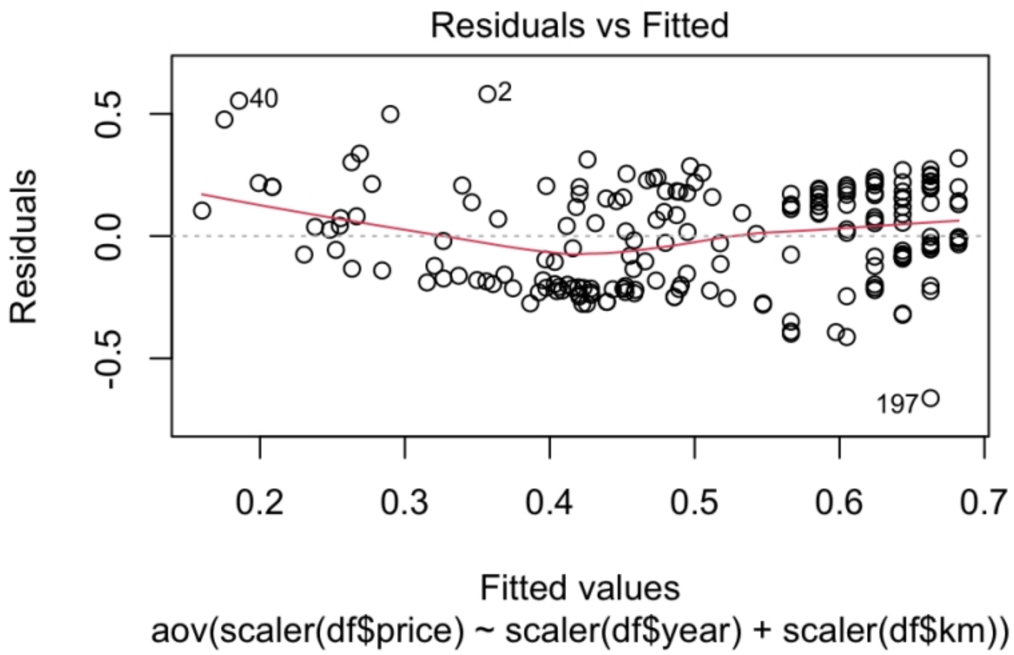
```
> set.seed(1234)
> sample_data <- sample_n(df, 30)
```

then we wrote a scaling function to scale the data (normalize it)

```
scaler <- function(x, na.rm = TRUE) { return((x- min(x)) /(max(x)-min(x))) }
```

Now we computer one way analysis of variance test

```
> aov_test <- aov(scaler(df$price) ~ scaler(df$year) + scaler(df$km))
> print(aov_test)
Call:
   aov(formula = scaler(df$price) ~ scaler(df$year) + scaler(df$km))

Terms:
                scaler(df$year) scaler(df$km) Residuals
Sum of Squares         2.322365      1.248049  8.225368
Deg. of Freedom               1             1       194

Residual standard error: 0.2059097
Estimated effects may be unbalanced
```



Hypothesis test

Ali claims that the age of the prices of the cars have a x > 0.40 correlation with the year its produced

```
> cor(sample_df$price, sample_df$year)
[1] 0.6828854
```

We accept his claim since the correlation is 0.6828854, however, it can be lower with a different sample.

Omnia thinks that the fuel type have a significant effect on the price of the car, lets see check the mean prices of each type of fuel

```
> diesel <- df[df$fuel == 'diesel',]
> mean(diesel$price)
[1] 447720
```

```
benzin <- df[df$fuel == "gasoline",]
> mean(benzin$price)
[1] 617093.8
```

However, the data might not be balanced, so we will try to check the prices of the brand new cars only

```
> new_gas <- df[df$fuel == 'gasoline' & df$km == 0,]
> mean(new_gas$price)
[1] 637285.7
> new_diesel <- df[df$fuel == 'diesel' & df$km == 0,]
> mean(new_diesel$price)
[1] 565974.3
```

now we can see that the price difference got lower, however, we cant be sure since the information about the fuel type is not balanced.

Rayyan claims that cars with higher km have low difference in terms of prices in comparison to brand new cars.

However, we can see that the cars price to km correlation is relatively high.

we reject rayyans claim

```
> cor(df$price, df$km)
[1] -0.5163152
```

Non-Parametic tests:

when we run shapiro test to see the normality of the data, we can see that the data is far from normal

```
> shapiro.test(heyo$price) Shapiro-Wilk normality test

data: heyo$price
W = 0.93811, p-value = 1.884e-07

> shapiro.test(heyo$km) Shapiro-Wilk normality test

data: heyo$km
W = 0.82932, p-value = 6.2e-14

> shapiro.test(heyo$year) Shapiro-Wilk normality test

data: heyo$year
W = 0.94072, p-value = 3.162e-07
```

! 3.162e-7 in decimal form = 0.0000003162

Since the data is not normaliy distributed, and also dependent on eachother, we will use kendalls test to see the relation between two numerical parameters.

Executing kendalls test, we can see that the lowe the kilometers of a car the higher its price, the estimated effect is 41%.

```
> cor.test(df$price, df$km, method="kendall") Kendall's rank
correlation tau

data: df$price and df$km
z = -7.9705, p-value = 1.58e-15
alternative hypothesis: true tau is not equal to 0 sample estimates:

tau -0.4086659
```

Pre-Processing:

In order to make our data readable by a linear regression model, we need no encode each categorical string variable into an int or float type variables, and this process has been done on our data as follows for 6 columns:

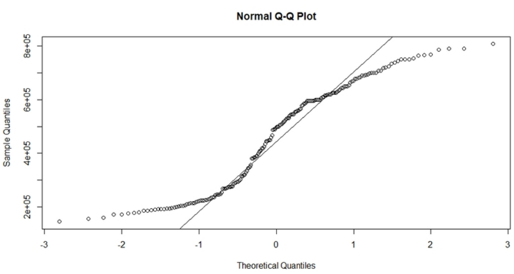
```
# Data Encoding:
  #1. brand:
label <- LabelEncoder$new()
print(label$fit(df$brand))
df$brand <- label$fit_transform(df$brand)
print(df$brand)
```

Then the data scale needed to be better as the varianve of the data is big, so we rescalled the dataset using the scale function as follows:

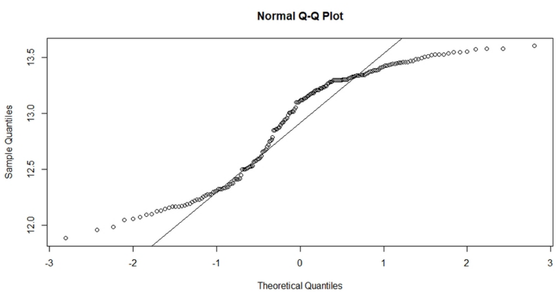
```
# standarization:
scale(df)
```

And lastly, we tried to fix the skweness in the price column (the target column for linear regression) using the logarithm of the column, and we noticed a change as follows:

```
> df$price <- log(df$price)
> skewness(df$price)
```



Q-Q Plot Before taking the log of the column



Q-Q Plot After taking the log of the column

Regression :

In order to apply the linear regression model we splitted the data into 70% training sample and 30% testing sample randomly after removing the color column as it has no correlation with the price and it does not effect the car price by any means, then we defined the target column the data set as follows for the model:

```
# Linear Regression:
  #Selection:
df = df[!names(df) %in% c("color")]


#install.packages('caTools')
library(caTools)

split = sample.split(df$price, SplitRatio = 0.7)
trainingset = subset(df, split == TRUE)
testset = subset(df, split == FALSE)


price.lm<-lm(price ~  + brand  + model  +  year +  status  +  fuel  +  gear +
               + km  + case_type ,data = df)
```

And here is the summary of the model results:

- The Residual Standard Error is 0.2127 which means that the model is not giving the best acurate results after being fitted to the dataset.

- The Multiple R-Squared Error is 0.8

```
> summary(price.lm)

Call:
lm(formula = price ~ +brand + model + year + status + fuel + 
    gear + +km + case_type, data = df)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.70197 -0.11412  0.00203  0.11195  0.79744 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) -4.920e+01  1.552e+01  -3.169  0.00178 ** 
brand       -1.721e-01  2.606e-02  -6.603 4.00e-10 ***
model        3.956e-02  1.928e-02   2.052  0.04156 *  
year         3.122e-02  7.707e-03   4.051 7.46e-05 ***
status      -1.208e-01  7.410e-02  -1.630  0.10483    
fuel         2.838e-01  6.482e-02   4.378 1.99e-05 ***
gear        -1.162e-01  2.110e-02  -5.504 1.20e-07 ***
km          -4.332e-07  2.970e-07  -1.458  0.14642    
case_type   -6.341e-02  3.171e-02  -2.000  0.04695 *  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2127 on 188 degrees of freedom
  (1 observation deleted due to missingness)
Multiple R-squared:  0.8129,	Adjusted R-squared:  0.805 
F-statistic: 102.1 on 8 and 188 DF,  p-value: < 2.2e-16
```

Conclusion:

Car prices in Istanbul varries significantly based on some features such as the brand name and the number of kilometers driven by the car, however, some features do not effect the price at all such as the color of the car. In Addition, the collected data is not only random, but it is also highly unbalanced such as the fuel column which has only two variables and one of them repeated 181 times while the other repeated only 16, which could be considered as biased in the dataset and any Machine learning model applied on the dataset will give biase predictions for the benifit of the diesl type fuel.

The dataset has a multimodal distribution, and the data visualization shows that some car prices are super high causing that distribution shape, and after the analysis, we find that the year of the car production highly affect the price of the car along with the status of the car (wether it is new or used).

Furthermore, highst and lowest collected car prices were considered as outliers and we can conclude that the BMW brand type has the highst prices in istanbul while the Renult is the lowest in price.

Source Code:

sourse code

```
#install.packages("ggplot2")
#install.packages("pastecs")
#install.packages("corrplot")
#install.packages("GGally")
#install.packages("dplyr")
#install.packages("ggpubr")
#install.packages("scales")
#install.packages("moments")
#install.packages("ggstatsplot")
#install.packages("superml")
install.packages("superml")
library("GGally")
library("readxl")
```

```
library("dplyr")
library(ggplot2)
library(pastecs)
library(corrplot)
library("ggpubr")
library(scales)
library(ggstatsplot)
library(superml)



df = read_excel("/Users/omniaelmenshawy/Desktop/Statistics Project/heyo.xlsx")
head(df)
# Basic data Exploration:
summary(df)
quantile(df$price)
stem(df$price)
stat.desc(df$price)
stat.desc(df$km)
print(" Price Coefficient of variation is: ")
sd(df$price) / mean(df$price)
sd(df$price)
#Null:
colSums(is.na(df))
#Duplicated:
duplicated(df)
# Balance:
table(df$status)
table(df$brand)
table(df$model)
table(df$year)
table(df$color)
table(df$fuel)
# Categorical Columns Distribution Visualization:
  #1- Status
common_theme <- theme(plot.title = element_text(hjust = 0.5, face = "bold"))
```

```
ggplot(data = df, aes(x = factor(status),
                      y = prop.table(stat(count)), fill = factor(status),
                      label = scales::percent(prop.table(stat(count))))) +
  geom_bar(position = "dodge") +
  geom_text(stat = 'count',
            position = position_dodge(.9),
            vjust = -0.5,
            size = 3) +
  scale_x_discrete(labels = c("New", "Used"))+
  scale_y_continuous(labels = scales::percent)+
  labs(x = 'status', y = 'Percentage') +
  ggtitle("Distribution of Status labels") +
  common_theme
  #2- Brand
common_theme <- theme(plot.title = element_text(hjust = 0.5, face = "bold"))
ggplot(data = df, aes(x = factor(brand),
                      y = prop.table(stat(count)), fill = factor(brand),
                      label = scales::percent(prop.table(stat(count))))) +
  geom_bar(position = "dodge") +
  geom_text(stat = 'count',
            position = position_dodge(.9),
            vjust = -0.5,
            size = 3) +
  scale_y_continuous(labels = scales::percent)+
  labs(x = 'Brand', y = 'Percentage') +
  ggtitle("Distribution of Brand labels") +
  common_theme
  #3- Case_Type
common_theme <- theme(plot.title = element_text(hjust = 0.5, face = "bold"))
ggplot(data = df, aes(x = factor(case_type),
                      y = prop.table(stat(count)), fill = factor(case_type),
                      label = scales::percent(prop.table(stat(count))))) +
  geom_bar(position = "dodge") +
  geom_text(stat = 'count',
            position = position_dodge(.9),
            vjust = -0.5,
```

```
          size = 3) +
  scale_y_continuous(labels = scales::percent)+
  labs(x = 'Case Type', y = 'Percentage') +
  ggtitle("Distribution of Case Types ") +
  common_theme
  #4- fuel:
common_theme <- theme(plot.title = element_text(hjust = 0.5, face = "bold"))
ggplot(data = df, aes(x = factor(fuel),
                      y = prop.table(stat(count)), fill = factor(fuel),
                      label = scales::percent(prop.table(stat(count))))) +
  geom_bar(position = "dodge") +
  geom_text(stat = 'count',
          position = position_dodge(.9),
          vjust = -0.5,
          size = 3) +
  scale_y_continuous(labels = scales::percent)+
  labs(x = 'Fuel Types', y = 'Percentage') +
  ggtitle("Distribution of Fuel Types ") +
  common_theme
  #5- model:
common_theme <- theme(plot.title = element_text(hjust = 0.5, face = "bold"))
ggplot(data = df, aes(x = factor(model),
                      y = prop.table(stat(count)), fill = factor(model),
                      label = scales::percent(prop.table(stat(count))))) +
  geom_bar(position = "dodge") +
  geom_text(stat = 'count',
          position = position_dodge(.9),
          vjust = -0.5,
          size = 3) +
  scale_y_continuous(labels = scales::percent)+
  labs(x = 'Model Types', y = 'Percentage') +
  ggtitle("Distribution of Model Types ") +
  common_theme
#5- year:
common_theme <- theme(plot.title = element_text(hjust = 0.5, face = "bold"))
```

```
ggplot(data = df, aes(x = factor(year),
                      y = prop.table(stat(count)), fill = factor(year),
                      label = scales::percent(prop.table(stat(count))))) +
  geom_bar(position = "dodge") +
  geom_text(stat = 'count',
            position = position_dodge(.9),
            vjust = -0.5,
            size = 3) +
  scale_y_continuous(labels = scales::percent)+
  labs(x = 'Years', y = 'Percentage') +
  ggtitle("Distribution of Model Production Year ") +
  common_theme
  #5- color:
common_theme <- theme(plot.title = element_text(hjust = 0.5, face = "bold"))
ggplot(data = df, aes(x = factor(color),
                      y = prop.table(stat(count)), fill = factor(color),
                      label = scales::percent(prop.table(stat(count))))) +
  geom_bar(position = "dodge") +
  geom_text(stat = 'count',
            position = position_dodge(.9),
            vjust = -0.5,
            size = 3) +
  scale_y_continuous(labels = scales::percent)+
  labs(x = 'Color', y = 'Percentage') +
  ggtitle("Distribution of the Color of the car") +
  common_theme

# Box Plots to check Outliers:
  #1. Status:
ggplot(df, aes(x = factor(status), y = price)) + geom_boxplot() +
  labs(x = 'Status', y = 'Price') +
  ggtitle("Boxplot of status and price") + common_theme


  #2. Year:
ggplot(df, aes(x = factor(year), y = price)) + geom_boxplot() +
```

```r
  labs(x = 'year', y = 'Price') +
  ggtitle("Boxplot of year and price") + common_theme

  #3. Model:

ggplot(df, aes(x = factor(model), y = price)) + geom_boxplot() +
  labs(x = 'model', y = 'Price') +
  ggtitle("Boxplot of model and price") + common_theme


  #4. case_type:

ggplot(df, aes(x = factor(case_type), y = price)) + geom_boxplot() +
  labs(x = 'case_type', y = 'Price') +
  ggtitle("Boxplot of case_type and price") + common_theme

  #5. Price:
boxplot(df$price)

  #5. km:
boxplot(df$km)

# Visualization and Numerical Correlation:
  #1. Price vs k  correlation : Negative correlation

ggplot(data=df[!is.na(df$price),], aes(x=km, y=price))+
  geom_point(col='blue') + geom_smooth(method = "lm", se=FALSE, color="black", aes(group=1)) +
  scale_y_continuous(breaks= seq(0, 800000, by=100000), labels = comma)

cor(df$price[!is.na(df$price)], df$km[!is.na(df$km)])
#Normality Test and random sampling:

 #1. Histogram

hist(df$price, col='steelblue', main='Price Distribution')
```

```
 #2. Q-Q plot
qqnorm(df$price, main='Price Q-Q plot')
qqline(df$price)
  #3. Shapiro-Wilk Test
shapiro.test(df2$price)

  #4. Kolmogorov-Smirnov Test

ks.test(df2$price, 'pnorm')

  #5. Random sampling with density and q plot distribution:
df2 = dplyr::sample_n(df,30)
ggdensity(df2$price,
          main = "Density plot of the data over the prices",
          xlab = "Prices")
ggqqplot(df2$price,
         main = "Linear plot of the data over the prices",
         xlab = "Prices")
# Point estimations and confidence intervals:
  # sample mean as point estimation:
mean(df$price, na.rm = TRUE)
  # Sample size, std, margin
n <- 197
xbar <- mean(df$price, na.rm = TRUE)
s <- sd(df$price)
margin <- qt(0.975, df=n-1)*s/sqrt(n)
low <- xbar - margin
print(low)
high <- xbar + margin
print(high)
k <- sum(df$price < 240000)
p <- k/n
print("95% confidence interval, and estimate the percentage of cars price bellow 240,000:")
print(p)

#Hypothesis Testing:
```

```
  #1. are the new car prices higher than the used ones
  #2. Does the fuel affect the price of a car
  #3. does the number of kilometers significantly decreases the car price of the same model
  #4.Are two cars of the same model share the same price with the fuel or engine type
#Goodness of fit:
chisq <- chisq.test(df$price)
chisq
chisq$p.value
chisq$estimate
round(chisq$expected,2)
round(chisq$residuals, 3)
corrplot(chisq$residuals, is.cor = FALSE)

#ANOVA
#Application of non-parametric test

#Handelling the Outliers with:
lower_bound <- quantile(df$price, 0.01)
print(lower_bound)
upper_bound <- quantile(df$price, 0.99)
print(upper_bound)
outlier_ind <- which(df$price < lower_bound | df$price > upper_bound)
outlier_ind
print("Rows with Outliers are: ")
df[outlier_ind, ]
  # replacing outliers with Q1 and Q2:
df[1, 10] = 620000
df[2, 10] = 620000
df[197, 10] = 267900
df[198, 10] = 267900
# Handelling the skewness:
  # 1. Price skweness
skewness(df$price)
qqnorm(df$price)
qqline(df$price)
df$price <- log(df$price)
```

```
skewness(df$price)
qqnorm(df$price)
qqline(df$price)
# Data Encoding:
 #1. brand:
label <- LabelEncoder$new()
print(label$fit(df$brand))
df$brand <- label$fit_transform(df$brand)
print(df$brand)
 #2. model:
label <- LabelEncoder$new()
print(label$model(df$model))
df$model <- label$fit_transform(df$model)
print(df$model)

 #3. fuel:
label <- LabelEncoder$new()
print(label$fuel(df$fuel))
df$fuel <- label$fit_transform(df$fuel)
print(df$fuel)

 #4. gear:
label <- LabelEncoder$new()
print(label$gear(df$gear))
df$gear <- label$fit_transform(df$gear)
print(df$gear)

 #5. case_type:
label <- LabelEncoder$new()
print(label$case_type(df$case_type))
df$case_type <- label$fit_transform(df$case_type)
print(df$case_type)

 #6. color:
label <- LabelEncoder$new()
print(label$color(df$color))
```

```
df$color <- label$fit_transform(df$color)
print(df$color)

# standarization:
scale(df)
# Data Correlation:
M = cor(df)
corrplot(M, method = 'shade')

# Linear Regression:
  #Selection:
df = df[!names(df) %in% c("color")]
#install.packages('caTools')
library(caTools)
split = sample.split(df$price, SplitRatio = 0.7)
trainingset = subset(df, split == TRUE)
testset = subset(df, split == FALSE)
price.lm<-lm(price ~  + brand  + model  +  year +  status  +  fuel  +  gear +
             + km  + case_type ,data = df)
summary(price.lm)
par(mfrow=c(2,2))
plot(price.lm)
par(mfrow=c(1,1))
```

Resources:

[1] *Web Site for data Collection*. (2022). Sahibinden. https://www.sahibinden.com

[2] *ggplot Documentation finder Site*. (2022). Ggplot. https://ggplot2.tidyverse.org/

[3] *Corplot Documentation*. (2022). Correlation

https://cran.rproject.org/web/packages/corrplot/vignettes/corrplot-intro.html

[4] *SuperML Documentation finder Site*. (2022). Linear Reg.

https://www.rdocumentation.org/packages/superml/versions/0.5.5

[3] R code examples

https://www.geeksforgeeks.org/