Estimating the prices of the used cars

Khin Mon

Table of Contents

# Loading the packages

# load dataset  
CarPrice <- read.csv("../dataset/used\_cars.csv")  
head(CarPrice)

brand model year price transmission mileage fuelType tax mpg engineSize  
1 Audi A1 2017 12500 Manual 15735 Petrol 150 55.4 1.4  
2 Audi A6 2016 16500 Automatic 36203 Diesel 20 64.2 2.0  
3 Audi A1 2016 11000 Manual 29946 Petrol 30 55.4 1.4  
4 Audi A4 2017 16800 Automatic 25952 Diesel 145 67.3 2.0  
5 Audi A3 2019 17300 Manual 1998 Petrol 145 49.6 1.0  
6 Audi A1 2016 13900 Automatic 32260 Petrol 30 58.9 1.4

# Data description

Dataset contains the following variables:

* brand (manufacturer)
* model (of car)
* year (of registration of the car)
* price (in GB pounds)
* transmission (type of gearbox)
* mileage (total distance covered by the car)
* fuelType (type of fuel used by the car)
* tax (annual cost of vehicle tax)
* mpg (miles per gallon - a measure of fuel efficiency)
* engineSize (size of the engine in litres)

# 1) Data Preparation

Filtering the dataset with the following properties:

* mileage less than 60000
* Manual transmission
* Petrol engine (fuelType)
* Costing less than £200 in annual Vehicle Tax.

# filtering these properties  
filtered\_CarPrice <- filter(CarPrice, mileage < 60000 & transmission == "Manual" & fuelType == "Petrol" & tax < 200)  
head(filtered\_CarPrice)

brand model year price transmission mileage fuelType tax mpg engineSize  
1 Audi A1 2017 12500 Manual 15735 Petrol 150 55.4 1.4  
2 Audi A1 2016 11000 Manual 29946 Petrol 30 55.4 1.4  
3 Audi A3 2019 17300 Manual 1998 Petrol 145 49.6 1.0  
4 Audi A3 2015 10200 Manual 46112 Petrol 20 60.1 1.4  
5 Audi A1 2016 12000 Manual 22451 Petrol 30 55.4 1.4  
6 Audi A3 2017 16100 Manual 28955 Petrol 145 58.9 1.4

# select a random sample of 2000 rows   
set.seed(29764) # for reproducibility! 29764  
n\_subsample <- 2000   
bootstrap\_CarPrice <-rep(NA, n\_subsample) # create an empty vector to store random samples   
  
# index <- sample(seq\_along(filtered\_CarPrice$price), n\_subsample, replace=FALSE)   
# index  
# bootstrap\_CarPrice <- filtered\_CarPrice[index, ]  
  
for(i in seq\_len(n\_subsample)){  
 index <- sample(seq\_along(filtered\_CarPrice$price),size=n\_subsample,replace=FALSE)   
 bootstrap\_CarPrice <- filtered\_CarPrice[index,]  
}  
head(bootstrap\_CarPrice) # df with 2000 rows and 10 cols

brand model year price transmission mileage fuelType tax mpg  
11550 Ford Fiesta 2016 9999 Manual 14063 Petrol 0 65.7  
4732 Ford Focus 2018 18891 Manual 12000 Petrol 145 41.5  
5725 Ford EcoSport 2018 13995 Manual 7016 Petrol 150 53.3  
10863 Ford Focus 2018 13500 Manual 22243 Petrol 150 57.7  
7686 Ford Fiesta 2018 12500 Manual 9102 Petrol 145 65.7  
3394 Ford Fiesta 2018 9000 Manual 10945 Petrol 145 64.2  
 engineSize  
11550 1.0  
4732 2.0  
5725 1.0  
10863 1.0  
7686 1.0  
3394 1.1

# removing the redundant variables  
UsedCarPrice = subset(bootstrap\_CarPrice, select = -c(transmission, fuelType) )  
head(UsedCarPrice) # df with 2000 rows and 8 cols

brand model year price mileage tax mpg engineSize  
11550 Ford Fiesta 2016 9999 14063 0 65.7 1.0  
4732 Ford Focus 2018 18891 12000 145 41.5 2.0  
5725 Ford EcoSport 2018 13995 7016 150 53.3 1.0  
10863 Ford Focus 2018 13500 22243 150 57.7 1.0  
7686 Ford Fiesta 2018 12500 9102 145 65.7 1.0  
3394 Ford Fiesta 2018 9000 10945 145 64.2 1.1

# 2) Exploratory Data Analysis

## 2.1) Descriptive Statistics

For continuous variables, min, median, mean, max and 1st Qu to 3rd Qu statistics are appropriate to check.

For categorical variables(brand, model and year),labeling correctly is needed.

# checking the dataset  
#view(dfSummary(UsedCarPrice))

# Code factor levels for categorial variables  
UsedCarPrice$brand <- factor(UsedCarPrice$brand, labels=c("Audi", "BMW", "Ford", "Mercedes"))  
  
# levels(UsedCarPrice$model)  
UsedCarPrice$model <- factor(UsedCarPrice$model, labels=c(  
 "1 Series", "2 Series", "3 Series", "4 Series", "A Class",  
 "A1", "A3", "A4", "A5", "B-MAX",  
 "B Class", "C-MAX", "C Class", "CL Class", "CLA Class",  
 "EcoSport", "Fiesta", "Focus", "Galaxy", "GLA Class",  
 "Grand C-MAX", "KA", "Ka+", "Kuga", "Mondeo",  
 "Mustang", "Puma", "Q2", "Q3", "S-MAX",  
 "SL CLASS", "TT", "X1"))  
  
table(UsedCarPrice$year)

2008 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020   
 1 2 1 5 51 87 118 246 507 489 445 48

# levels(UsedCarPrice$year)  
UsedCarPrice$year <- factor(UsedCarPrice$year, labels=c(  
 "2008", "2009", "2011", "2012", "2013", "2014",   
 "2015", "2016", "2017", "2018", "2019", "2020"))  
  
# summary statistics of continuous columns  
#flextable\_describe(X=UsedCarPrice[,c(4:8)])  
  
# Summary statistics of price of cars by each brand  
t1 <- psych::describeBy(UsedCarPrice$price, group=UsedCarPrice$brand, quant=c(.25,.75))  
TB1 <- matrix(NA,nrow=max(seq\_along(t1)), ncol=ncol(t1[[1]]))  
rownames(TB1) <- levels(UsedCarPrice$brand)  
colnames(TB1) <- colnames(t1[[1]])  
for(i in seq\_along(t1)){  
 TB1[i,] <- unlist(t1[[i]])  
}  
TB1 <- TB1[,-1]  
  
# format = "html", this did not work when knitting as word file, so commented out   
# TB1%>% kbl(format = "html", digits = 2) %>% kable\_classic(html\_font="Cambria",full\_width=FALSE)  
  
TB1

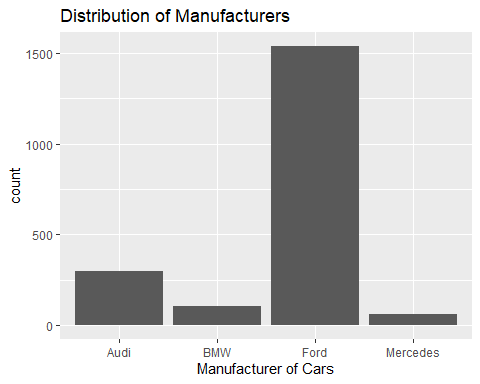
n mean sd median trimmed mad min max range  
Audi 298 16750.50 5284.053 16347.5 16278.89 5412.231 7800 34259 26459  
BMW 104 17335.42 5321.575 16414.5 17160.21 6690.233 8390 28655 20265  
Ford 1537 11616.68 4026.225 10750.0 11255.26 3338.815 2999 48000 45001  
Mercedes 61 18382.77 4335.524 17490.0 18172.29 4435.939 10990 28989 17999  
 skew kurtosis se Q0.25 Q0.75  
Audi 0.8052087 0.1843567 306.0970 12491.25 19734.50  
BMW 0.2051431 -1.1212061 521.8234 12942.50 21976.25  
Ford 1.4274958 5.6171703 102.6978 8977.00 13900.00  
Mercedes 0.4115788 -0.9025974 555.1070 14990.00 21750.00

## 2.2) Exploratory Graphs

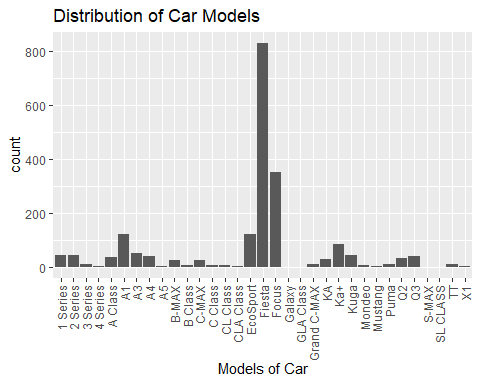
To explore the categorical variables, bar charts are appropriate because we can easily check the distributions of categorical data with bar charts.

Also, the distribution and relationship between two continuous variables such as numbers and datetime can be easily checked with the scatterplots.

# categorical variable - brand  
ggplot(data = UsedCarPrice) +  
 geom\_bar(mapping = aes(x = brand)) +   
 xlab("Manufacturer of Cars") +   
 ggtitle("Distribution of Manufacturers")

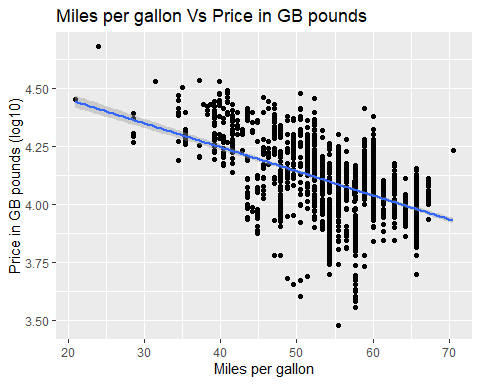


# categorical variable - model  
ggplot(data = UsedCarPrice) +  
 geom\_bar(mapping = aes(x = model)) +   
 theme(axis.text.x = element\_text(angle = 90, vjust = 0.5, hjust=1)) +  
 xlab("Models of Car") + ggtitle("Distribution of Car Models")



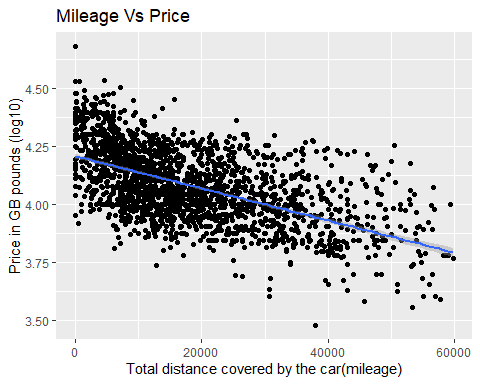
# two continuous variables, the relationship between price and mpg  
ggplot(UsedCarPrice, aes(x = mpg, y = log(price,base = 10))) +   
 geom\_point() +  
 geom\_smooth(method = "lm", se = TRUE) +  
 ylab("Price in GB pounds (log10)") +  
 xlab("Miles per gallon") +  
 ggtitle("Miles per gallon Vs Price in GB pounds")

`geom\_smooth()` using formula 'y ~ x'



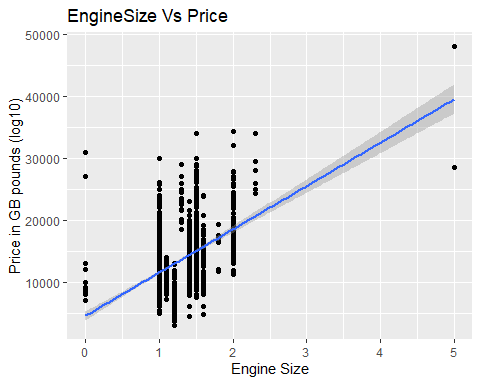
# two continuous variables, the relationship between price and mileage  
ggplot(UsedCarPrice, aes(x = mileage, y = log(price,base = 10))) +  
 geom\_point() +  
 geom\_smooth(method = "lm", se = TRUE) +  
 ylab("Price in GB pounds (log10)") +  
 xlab("Total distance covered by the car(mileage)") +  
 ggtitle("Mileage Vs Price")

`geom\_smooth()` using formula 'y ~ x'



# two continuous variables, the relationship between price and enginesize  
ggplot(UsedCarPrice, aes(x = engineSize, y = price)) +  
 geom\_point() +  
 geom\_smooth(method = "lm", se = TRUE) +  
 ylab("Price in GB pounds (log10)") +  
 xlab("Engine Size") +  
 ggtitle("EngineSize Vs Price")

`geom\_smooth()` using formula 'y ~ x'

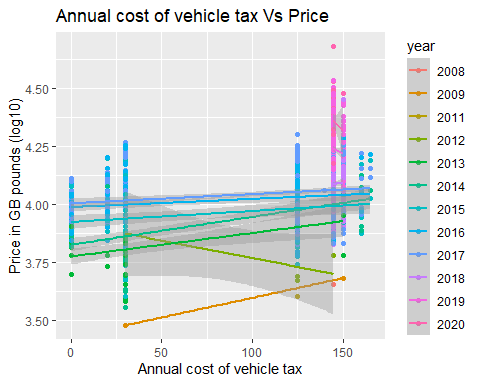


# two continuous variables, the relationship between price and annual vehicle tax  
ggplot(UsedCarPrice, aes(x = tax, y = log(price,base = 10) , colour = year)) +  
 geom\_point() +  
 geom\_smooth(method = "lm", se = TRUE) +  
 ylab("Price in GB pounds (log10)") +  
 xlab("Annual cost of vehicle tax") +  
 ggtitle("Annual cost of vehicle tax Vs Price")

`geom\_smooth()` using formula 'y ~ x'

Warning in qt((1 - level)/2, df): NaNs produced

Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning  
-Inf



The first “brand” graph, most people might choose the Ford brand and so it stands in the top among others. Mercedes looks less popular in the used car brand and it has minimum count.

The second “model” graph, Fiesta model has max records and CLA Class(2), Galaxy(1), Mustang(1), SL CLASS(1) and X2(1) models have a few counts in each.(There is no missing values in this variable. So, the min is 1 car.)

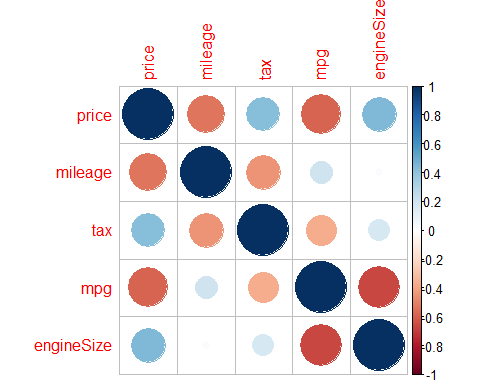
In third graph, I explored the relationship between miles per gallon and the price of car. It looks like there is negative correlation between those two variables. When mgp increases, the the price of car looks decrease. Also, the variation of price is not constant across mpg.

In fourth graph, it also has negative correlation between price and mileage.

The last fifth and sixth graph generally show the positive correlations with price of used cars.

## 2.3) Correlations

# only numeric variables  
X<-dplyr::select\_if(UsedCarPrice, is.numeric)  
M <- cor(X)  
corrplot(M)



# the linear correlation between variables price and the independent variables  
with(UsedCarPrice, cor(price, mpg)) # -0.6126887 (strongest negative correlation)

[1] -0.5888712

with(UsedCarPrice, cor(price, mileage)) # -0.5365772

[1] -0.5330638

with(UsedCarPrice, cor(price, engineSize)) # 0.4690952 (strongest positive correlation)

[1] 0.447848

with(UsedCarPrice, cor(price, tax)) # 0.408758

[1] 0.4255317

# 3) Bivariate relationship

1. Which of the potential explanatory variables has the strongest linear relationship with the dependent variable?

According to the corrplot and calculation of linear correlation between variables, mpg(miles per gallon) has strongest negative linear relationship with the price.

1. Creating a linear model to model this relationship.

model1<-lm(price ~ mpg, data=X)  
#as\_flextable(model1)

1. Explain and interpret the model:

The regression equation is:

An increase of 1 unit in mpg(independent, X) leads to an average decrease of 400 unit in price(dependent, Y).

The p-value for the whole model and mpg is highly significant (0.0000).

1. Comment on the performance of the model, including comments on overall model fit and the validity of model assumptions.

R-squared of the model = 0.3468. It means this model explains 34.68% of variation in price.

In residual linearity plots, those are analyzing the linearity of the fitted values and residuals. There are some exceptional errors when the model predictions(on x-axis) are around 20K and 25K compared to below those.

In influential Observations plot, it looks like all of the leverage points are inside of the contour lines. So, model adjustments might not have been needed.

In normality of residuals plots, dots are along with the lines except some of those(where model is not predicting well) near the end. The last one looks quite ok.

# check\_model(model1) # to do: check the required package to be installed

## 3.1) Bootstrap

1. Using bootstrapping on the model to obtain a 95% confidence interval of the estimate of the slope parameter.

# TBC