

Capstone Project : Used car price

HarvardX : Data Science Professional Certificate

Youssef Bougarrani

7/23/2021

Contents

1	Introduction	2
1.1	Data description	2
1.2	Session information	2
2	Analysis	3
2.1	Initial data exploration and visualization	3
2.2	Exploratory Data Analysis	4
3	Results	28
3.1	Data preparation for a step-by-step approach	28
3.2	First model	28
3.3	Data processing for next models	30
3.4	We define train and test test using the same seed as before	30
3.5	Linear model with caret	31
3.6	rpart model	31
3.7	random forest	32
3.8	xgboost	33
4	Conclusion	34

N.B : All plots are built with theme_minimal to reduce ink when printing.

1 Introduction

This report is the second part of capstone for HarvardX Data Science Professional Certificate. Data set is from the British used car listing about 100,000 vehicles.

The goal of this analysis is to predict the price knowing other features of the car. We start by an initial exploration of data set and some visualizations. Then we perform machine learning models and display results.

1.1 Data description

Data set is about used cars listings 100,000 cars, which have been separated into files corresponding to each car manufacturer.

The cleaned data set contains information of price, transmission, mileage, fuel type, road tax, miles per gallon (mpg), and engine size.

1.2 Session information

```
## R version 4.1.0 (2021-05-18)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 20.04.2 LTS
##
## Matrix products: default
## BLAS:    /usr/lib/x86_64-linux-gnu/blas/libblas.so.3.9.0
## LAPACK: /usr/lib/x86_64-linux-gnu/lapack/liblapack.so.3.9.0
##
## locale:
## [1] LC_CTYPE=en_US.UTF-8          LC_NUMERIC=C
## [3] LC_TIME=en_CA.UTF-8          LC_COLLATE=en_US.UTF-8
## [5] LC_MONETARY=en_CA.UTF-8      LC_MESSAGES=en_US.UTF-8
## [7] LC_PAPER=en_CA.UTF-8         LC_NAME=C
## [9] LC_ADDRESS=C                  LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_CA.UTF-8   LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats      graphics   grDevices utils      datasets  methods   base
##
## loaded via a namespace (and not attached):
## [1] compiler_4.1.0    magrittr_2.0.1    tools_4.1.0     htmltools_0.5.1.1
## [5] yaml_2.2.1       stringi_1.6.2    rmarkdown_2.8   knitr_1.33
## [9] stringr_1.4.0    xfun_0.23       digest_0.6.27   rlang_0.4.11
## [13] evaluate_0.14
```

Load libraries and install them if require

Load data

From kaggle “100,000 UK Used Car Data set”.

2 Analysis

2.1 Initial data exploration and visualization

Dimension of the data set

rows	columns
98926	10

Search for duplicated rows

```
## [1] 1475
```

1475 rows are duplicated.

We remove them.

Look for NAs

	Na
brand	0
model	0
year	0
price	0
transmission	0
mileage	0
fuelType	0
tax	0
mpg	0
engineSize	0

There are no NAs.

The structure

```
## Classes 'data.table' and 'data.frame':  97451 obs. of  10 variables:
## $ brand      : chr  "audi" "audi" "audi" "audi" ...
## $ model      : chr  "A1"  "A6"  "A1"  "A4"  ...
## $ year       : int  2017 2016 2016 2017 2019 2016 2016 2016 2016 2015 ...
## $ 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 ...
```

First six rows

brand	model	year	price	transmission	mileage	fuelType	tax	mpg	engineSize
audi	A1	2017	12500	Manual	15735	Petrol	150	55.4	1.4
audi	A6	2016	16500	Automatic	36203	Diesel	20	64.2	2.0

brand	model	year	price	transmission	mileage	fuelType	tax	mpg	engineSize
audi	A1	2016	11000	Manual	29946	Petrol	30	55.4	1.4
audi	A4	2017	16800	Automatic	25952	Diesel	145	67.3	2.0
audi	A3	2019	17300	Manual	1998	Petrol	145	49.6	1.0
audi	A1	2016	13900	Automatic	32260	Petrol	30	58.9	1.4

Units used

The price is in British Pound £.

Mileage is in mile. And mile = 1.609 km.

Tax is in British Pound £.

Mpg is the “miles per gallon”. And UK gallon ~ 4.5 liter.

EngineSize is in liter.

2.2 Exploratory Data Analysis

2.2.1 Brand

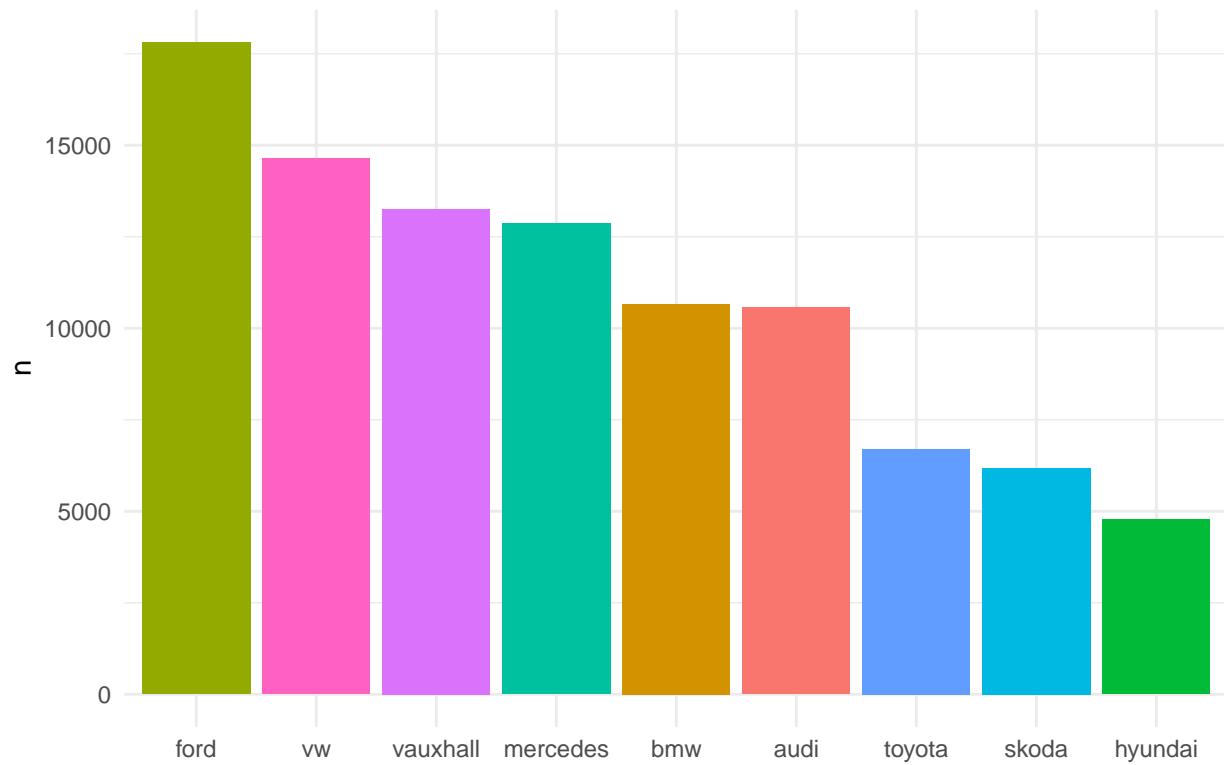
unique(brand)
audi
bmw
mercedes
ford
hyundai
skoda
toyota
vauxhall
vw

There are 9 brands.

Distribution of brands

brand	n
ford	17811
vw	14632
vauxhall	13258
mercedes	12860
bmw	10664
audi	10565
toyota	6699
skoda	6188
hyundai	4774

From the most to the least popular brand



As we can see in the UK the most popular car brand is Ford.

2.2.2 Model

Total models

```
##      n
## 1 186
```

Models with few number of observations

model	median(price)	n
180	10799.0	1
200	19495.0	1
220	19995.0	1
230	4500.0	1
A2	2490.0	1
Accent	1295.0	1
Amica	1750.0	1
Escort	3000.0	1
Ranger	14495.0	1
RS7	33490.0	1
Transit Tourneo	12450.0	1
R Class	9474.5	2
Streetka	1924.5	2
Terracan	3092.5	2
Ampera	11400.0	3

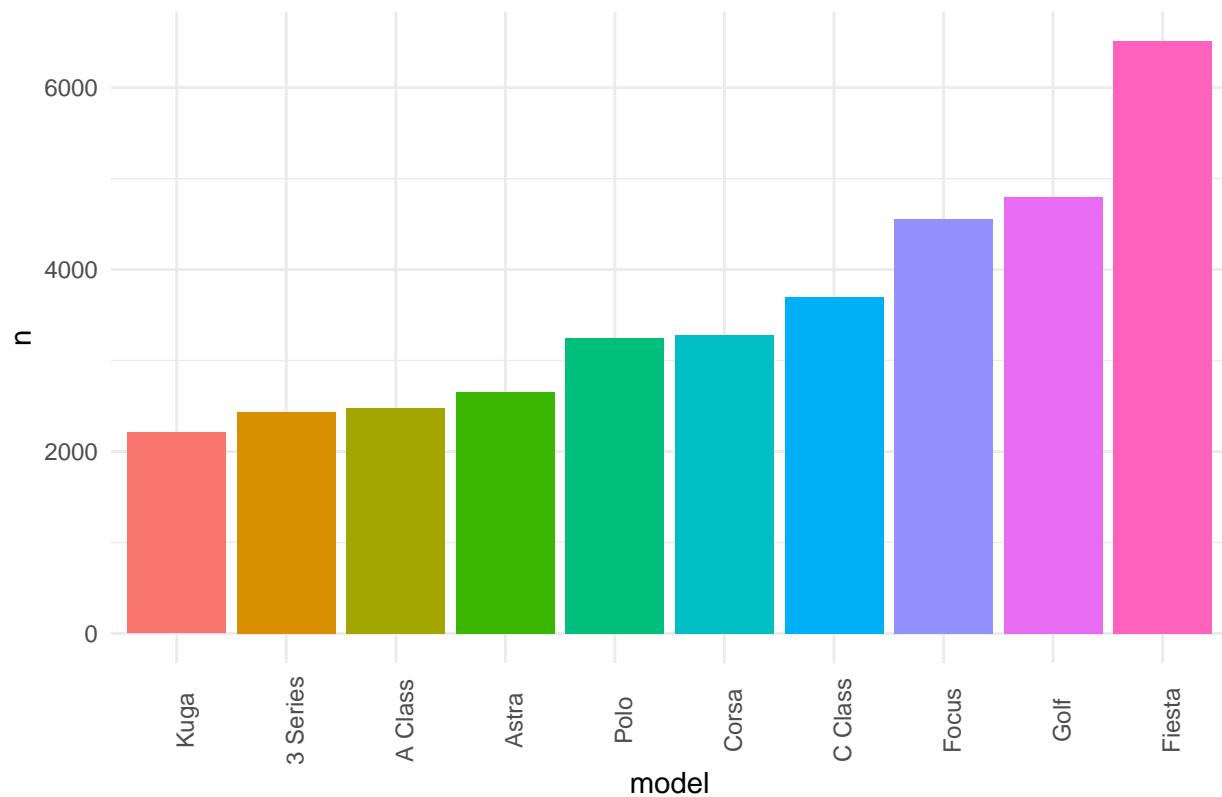
model	median(price)	n
CLC Class	5880.0	3
Kadjar	15100.0	3
S5	17495.0	3
Tigra	2499.0	3
Veloster	6495.0	3
Verso-S	5795.0	3
Cascada	9993.0	4
S8	33120.0	4
Urban Cruiser	4740.0	4
Vectra	1595.0	4
Getz	1992.5	6
CLK	3376.0	7
Z3	3995.0	7
IQ	4422.0	8
M6	33823.5	8
SQ7	47470.0	8
Camry	26491.0	11
S4	40425.0	12
Supra	49990.5	12
G Class	99850.0	15
PROACE VERSO	27990.0	15
Fusion	2443.0	16
SQ5	32419.0	16
i8	57870.0	17
Roomster	5495.0	17
GLB Class	37197.5	18
S3	19965.0	18
M2	44990.0	21
Vivaro	14900.0	21
Agila	4599.0	22
M3	31975.0	26
Antara	6500.0	27
R8	111490.0	28
RS5	54250.0	28
M5	63980.0	29

There are 50 over 186 models with less than 30 observations per model. It will make predictions less accurate.

Ten most popular models

model	n
Fiesta	6509
Golf	4797
Focus	4555
C Class	3694
Corsa	3285
Polo	3244
Astra	2655
A Class	2474
3 Series	2434
Kuga	2208

Most popular models



As we can see in the UK the most popular car models are fiesta, golf and Focus. It is clearly visible that Ford cars are the kings of the roads.

2.2.3 Year

Summary of year

```
##   Min. 1st Qu. Median    Mean 3rd Qu.    Max.
##   1970    2016    2017    2017    2019    2060
```

Search for non-logic value

brand	model	year	price	transmission	mileage	fuelType	tax	mpg	engineSize
ford	Fiesta	2060	6495	Automatic	54807	Petrol	205	42.8	1.4

There is a max value year = 2060 for a Ford Fiesta.

We find identical metrics for this model and change the year.

There are 11 Ford Fiesta with roughly same metrics and median year = 2010.

We replace 2060 by 2010.

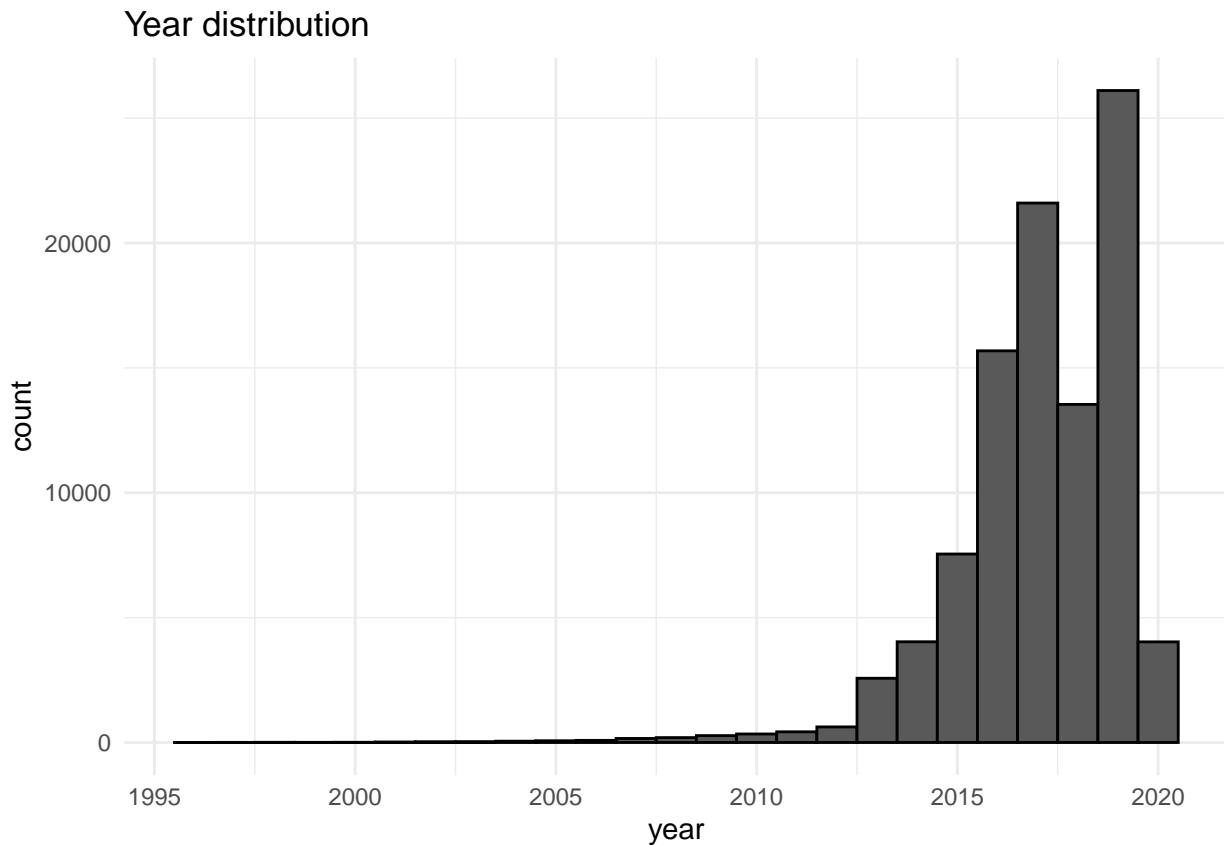
Check the range of year now :

```
## [1] 1970 2020
```

Table of number of cars per year

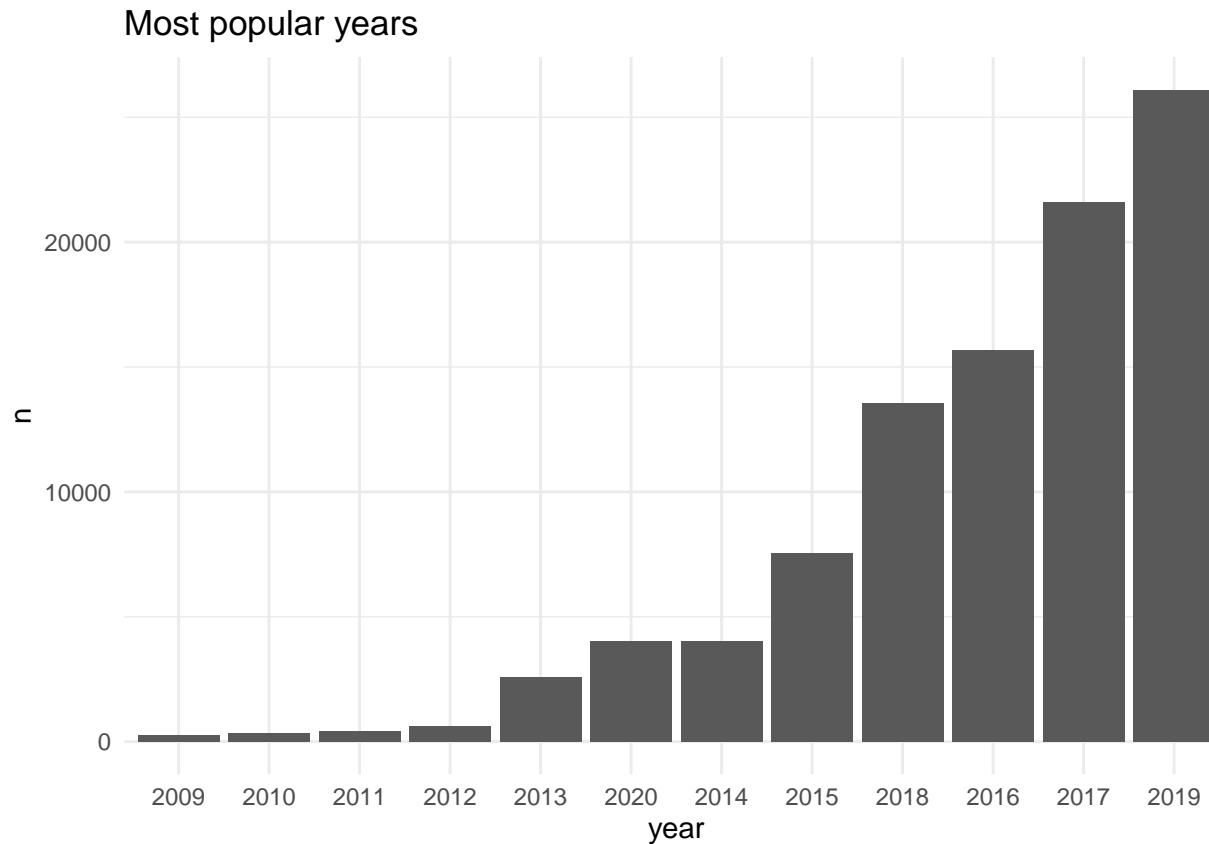
year	n
1970	2
1996	2
1997	4
1998	8
1999	6
2000	9
2001	20
2002	31
2003	34
2004	52
2005	69
2006	84
2007	161
2008	195
2009	276
2010	341
2011	428
2012	624
2013	2572
2014	4033
2015	7546
2016	15683
2017	21599
2018	13537
2019	26105
2020	4030

There are 2 vehicles with year = 1970 and these models are not from this period.
We replace 1970 by the median year for these models with same features.



Only 953 cars have year < 2010.

The year from which cars are the most popular



As we can see in the UK the most popular car year is 2019. Perhaps people prefer to buy cars that are 4 years old rather than new vehicles.

2.2.4 Transmission :

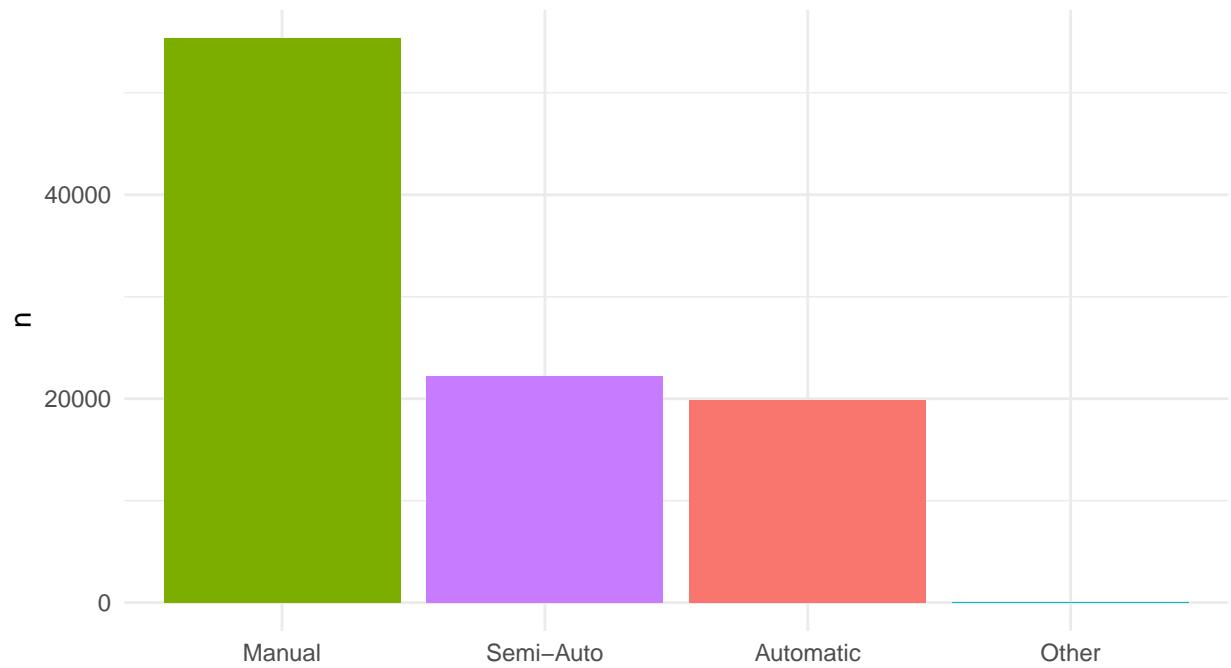
Number of cars by transmission

```
## 
##   Automatic      Manual      Other  Semi-Auto
##     19848       55378        9     22216
```

brand	model	year	price	transmission	mileage	fuelType	tax	mpg	engineSize
mercedes	GLA Class	2016	18700	Other	30895	Other	125	56.5	0.0
mercedes	SLK	2015	12995	Other	39000	Diesel	150	56.5	2.1
hyundai	Ioniq	2017	12495	Other	27779	Hybrid	0	78.5	1.6
hyundai	Tucson	2017	16995	Other	25915	Petrol	145	39.8	1.6
skoda	Scala	2019	15999	Other	3500	Petrol	145	47.1	1.0
toyota	Yaris	2015	12795	Other	16733	Hybrid	0	78.0	1.5
vauxhall	Mokka	2019	19995	Other	1450	Diesel	145	57.7	1.5
vauxhall	Mokka	2019	13499	Other	3000	Petrol	145	44.8	1.4
vauxhall	Mokka	2019	22499	Other	4500	Petrol	145	42.2	1.5

Only 9 cars have “Other” type of transmission.

Transmission distribution

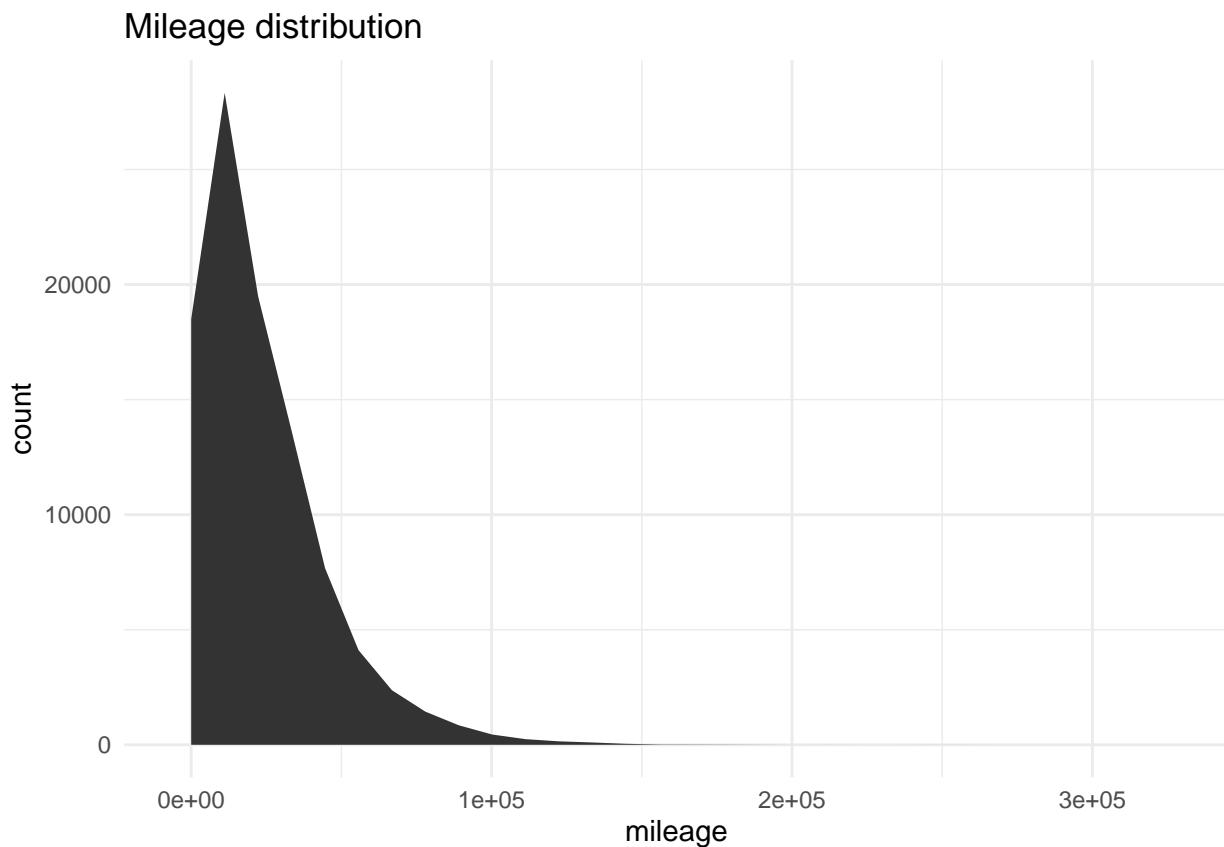


As we can see the most of the bought cars have manual gearbox

2.2.5 Mileage

Summary of mileage

```
##   Min. 1st Qu. Median   Mean 3rd Qu.   Max.  
##      1    7664  17663  23190  32463 323000
```

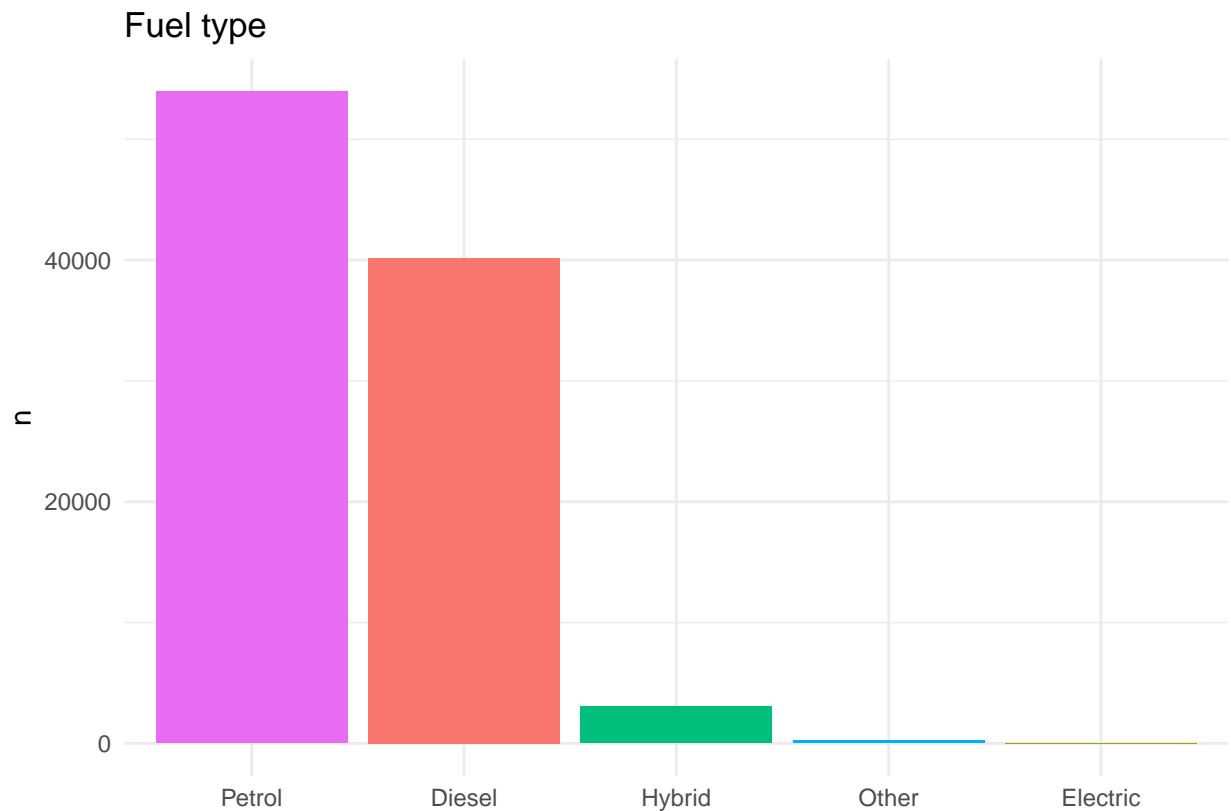


The distribution of mileage is right skewed.

2.2.6 FuelType

Distribution of fuelType

fuelType	count
Diesel	40175
Electric	6
Hybrid	3059
Other	246
Petrol	53965



As we can see most of the cars are Petrol.

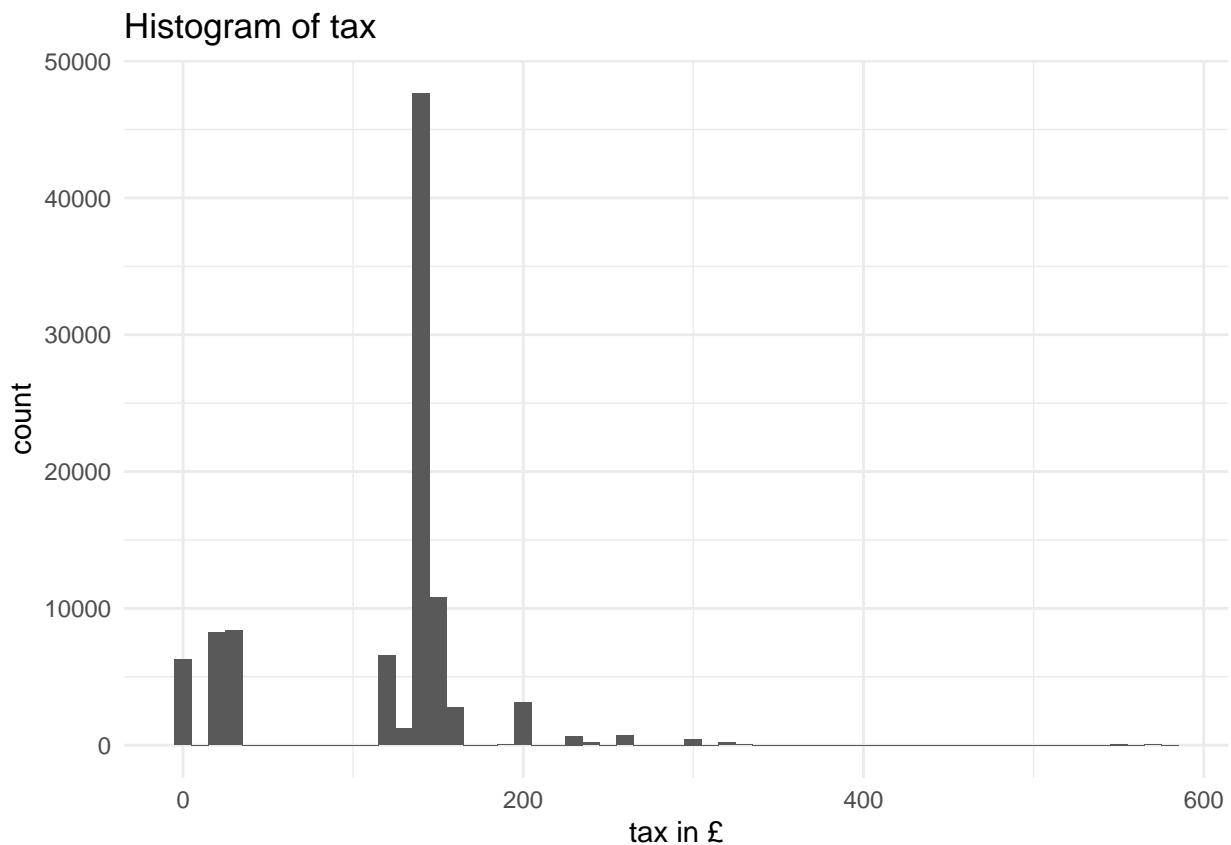
2.2.7 Tax

Summary of tax and total number of cars with small tax category

```
##   Min. 1st Qu. Median   Mean 3rd Qu.   Max.  
##   0.0  125.0 145.0 120.1 145.0 580.0
```

amount of tax	count
0	6259
10	25
20	8228
30	8381
110	2
115	12

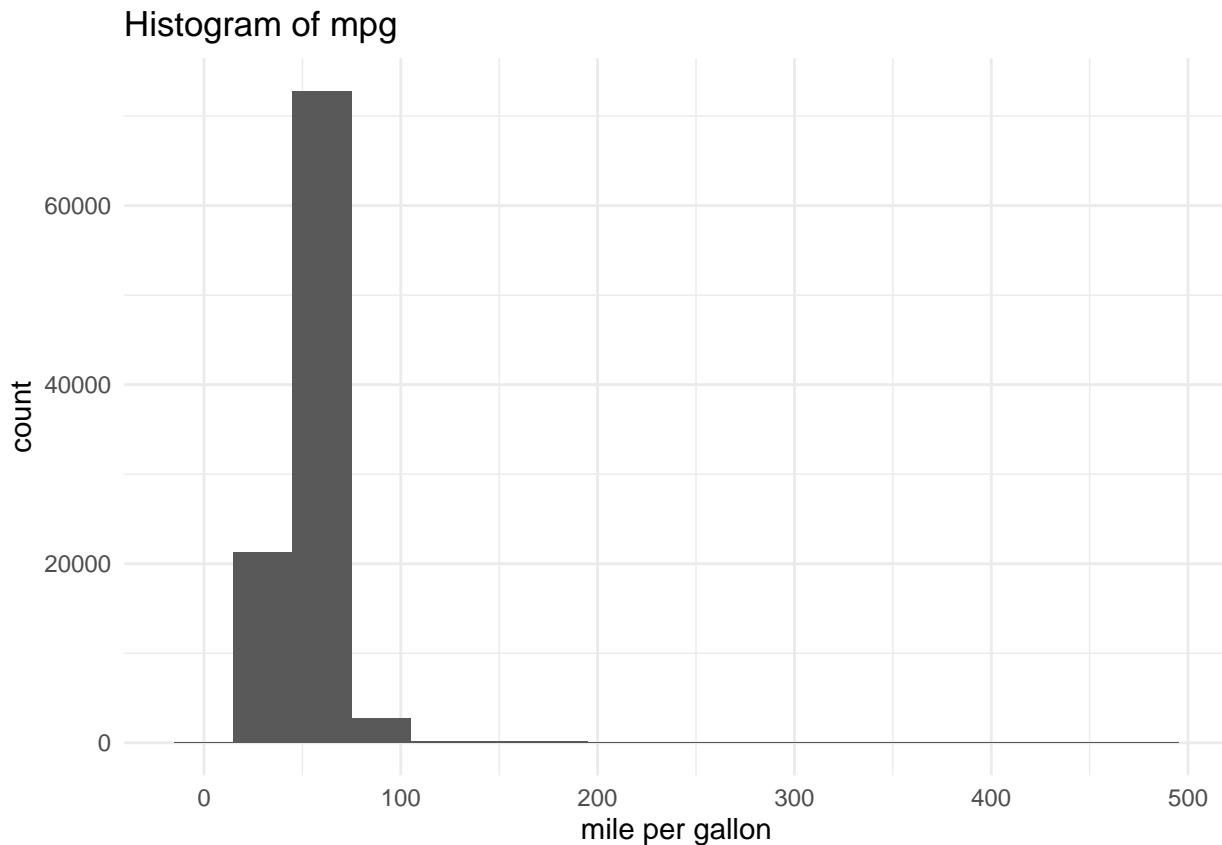
6259 cars have 0£ on tax, and only 3065 cars are electric or hybrid. This can be a mistake because cars in UK payed tax for CO2 rejections.



2.2.8 MPG

Summary of mpg

```
##   Min. 1st Qu. Median   Mean 3rd Qu.   Max.  
## 0.30 47.10 54.30 55.22 62.80 470.80
```



90% of cars have a rate of fuel consumption between 30mpg and 70 mpg.

2.2.9 EngineSize

Summary of engineSize and total number of cars with small engineSize

```
##   Min. 1st Qu. Median   Mean 3rd Qu.   Max.
##   0.000 1.200 1.600 1.664 2.000 6.600
```

engineSize	count
0	268
0.6	7
1	17083
1.1	558
1.2	6715
1.3	1307

273 vehicles have engine size = 0.

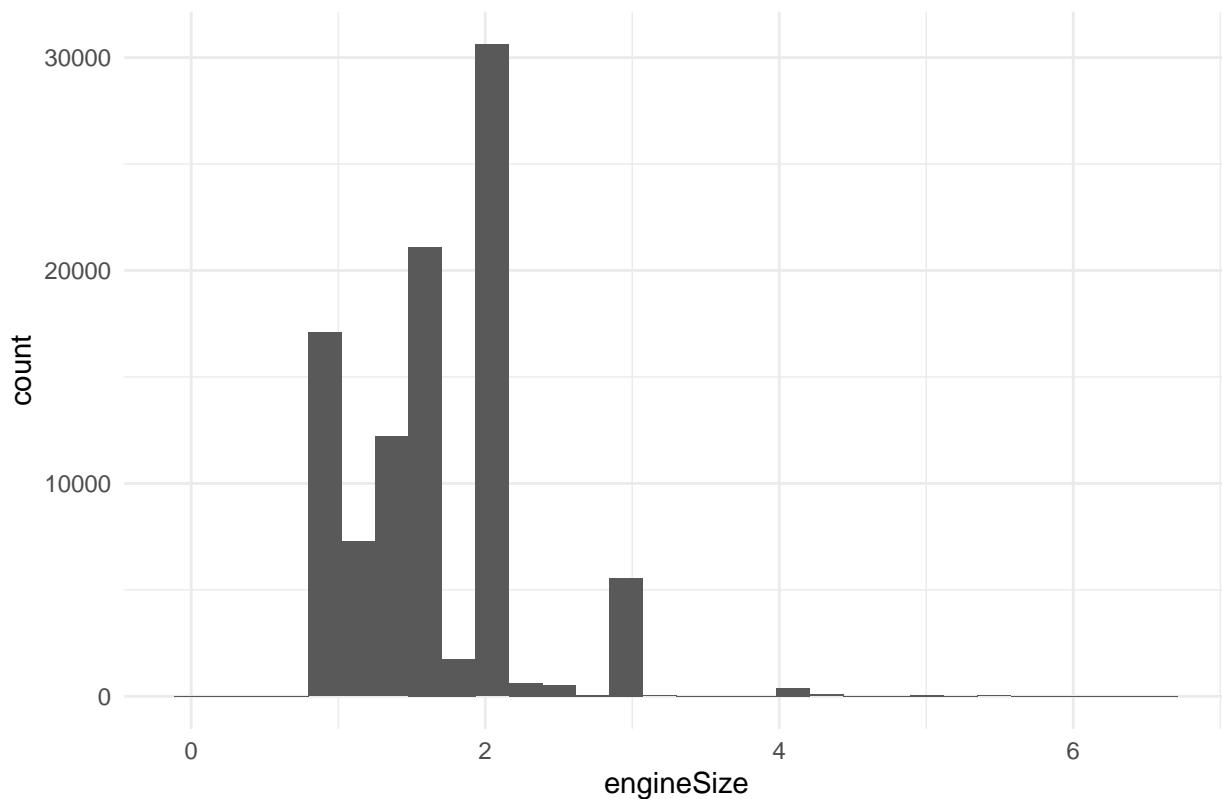
The assumption is electrical vehicles have enginSize = 0

brand	model	year	price	transmission	mileage	fuelType	tax	mpg	engineSize
bmw	i3	2017	18999	Automatic	20321	Electric	135	470.8	0.0
bmw	i3	2016	18999	Automatic	9990	Electric	0	470.8	0.0
bmw	i3	2015	17400	Automatic	29465	Electric	0	470.8	1.0
ford	Mondeo	2016	15975	Automatic	9396	Electric	0	67.3	2.0
ford	Mondeo	2016	15500	Automatic	24531	Electric	0	67.3	2.0
vauxhall	Ampera	2015	12999	Automatic	34461	Electric	0	235.4	1.4

But only 2 of 6 electrical cars have engine size = 0, others are not.

We replace all non-electric cars by the median. The median engineSize is 1.6 liter.

Histogram of engineSize

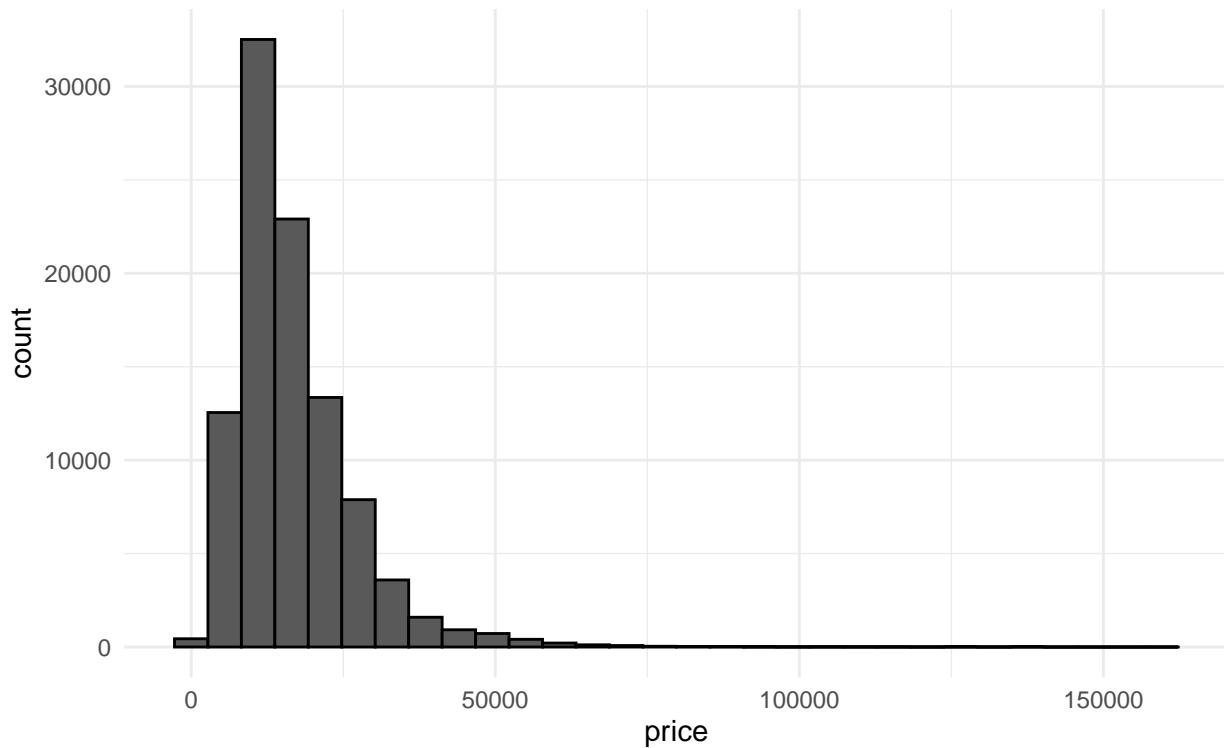


Most vehicles have less than 1; 1.6; 2; or 2.5 liter.

2.2.10 Price

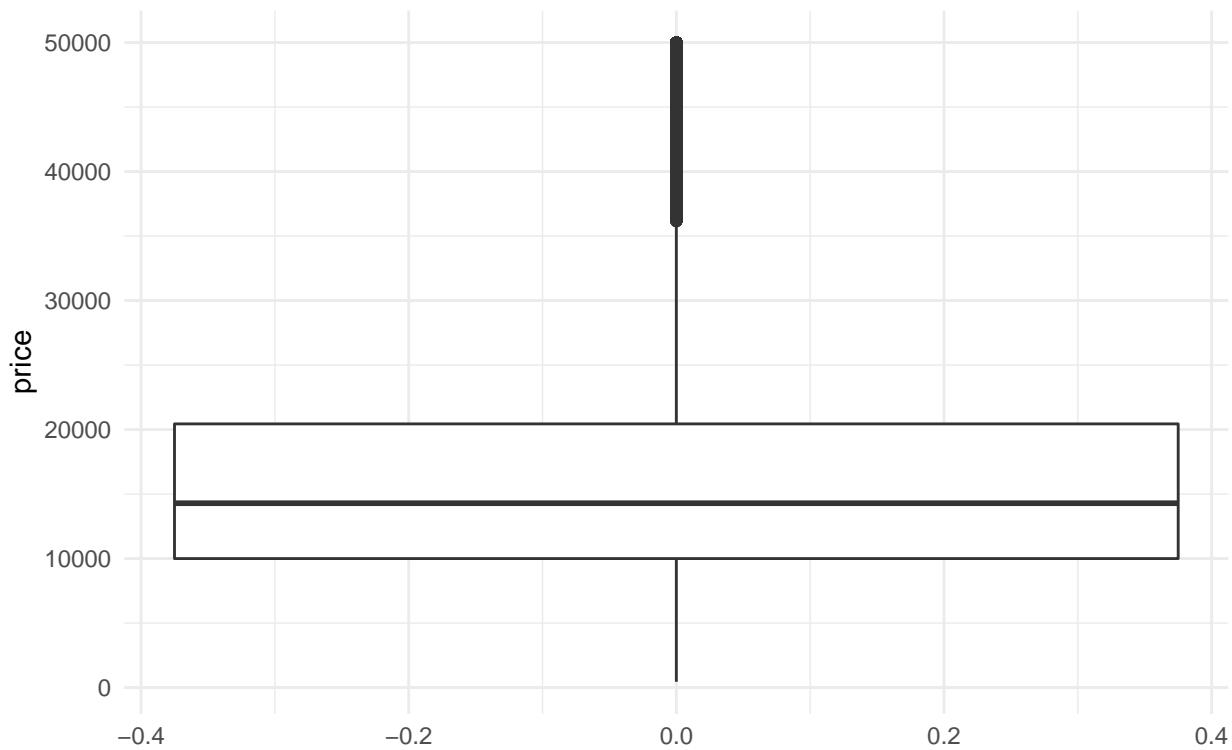
Summary of price

```
##   Min. 1st Qu. Median   Mean 3rd Qu.   Max.  
##     450    9999  14480  16769  20750 159999
```



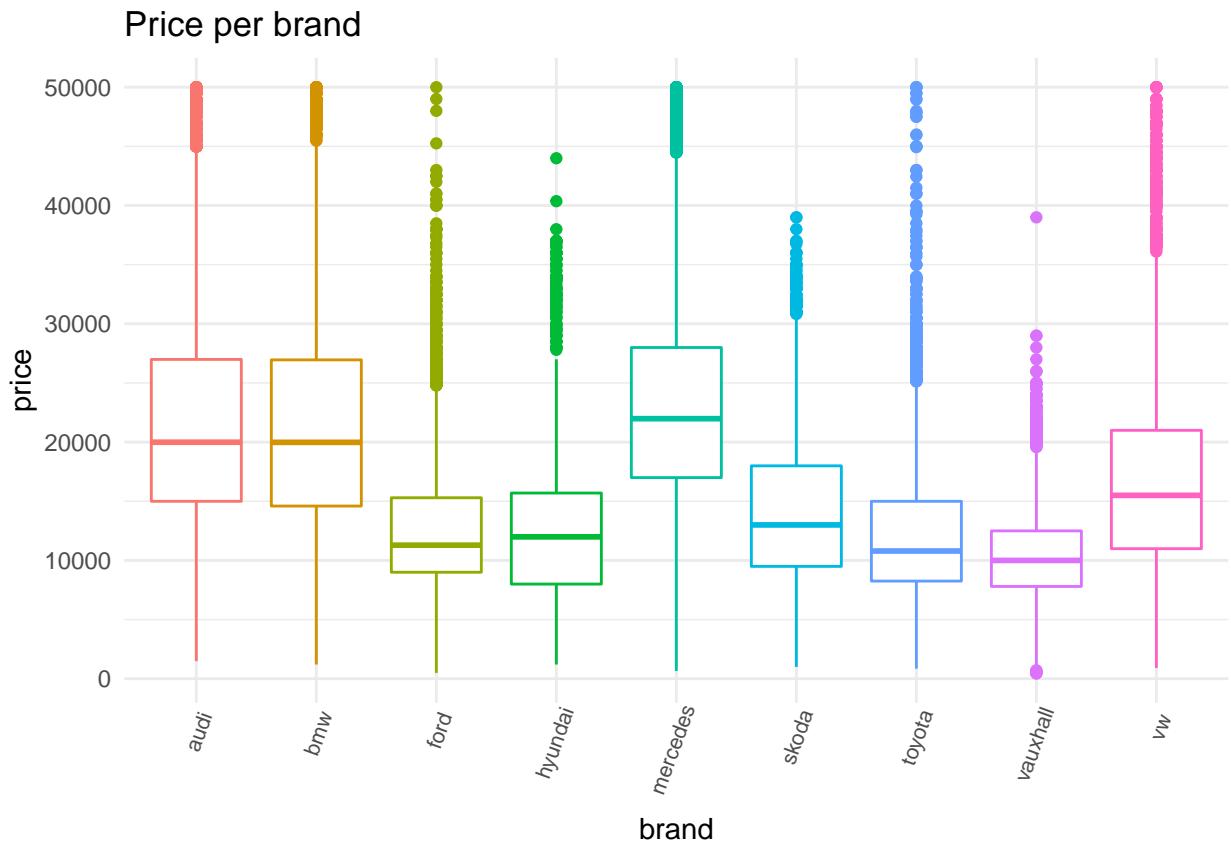
Very few cars have price > 50,000 £

1135 cars are, it's roughly 1% of all vehicles.



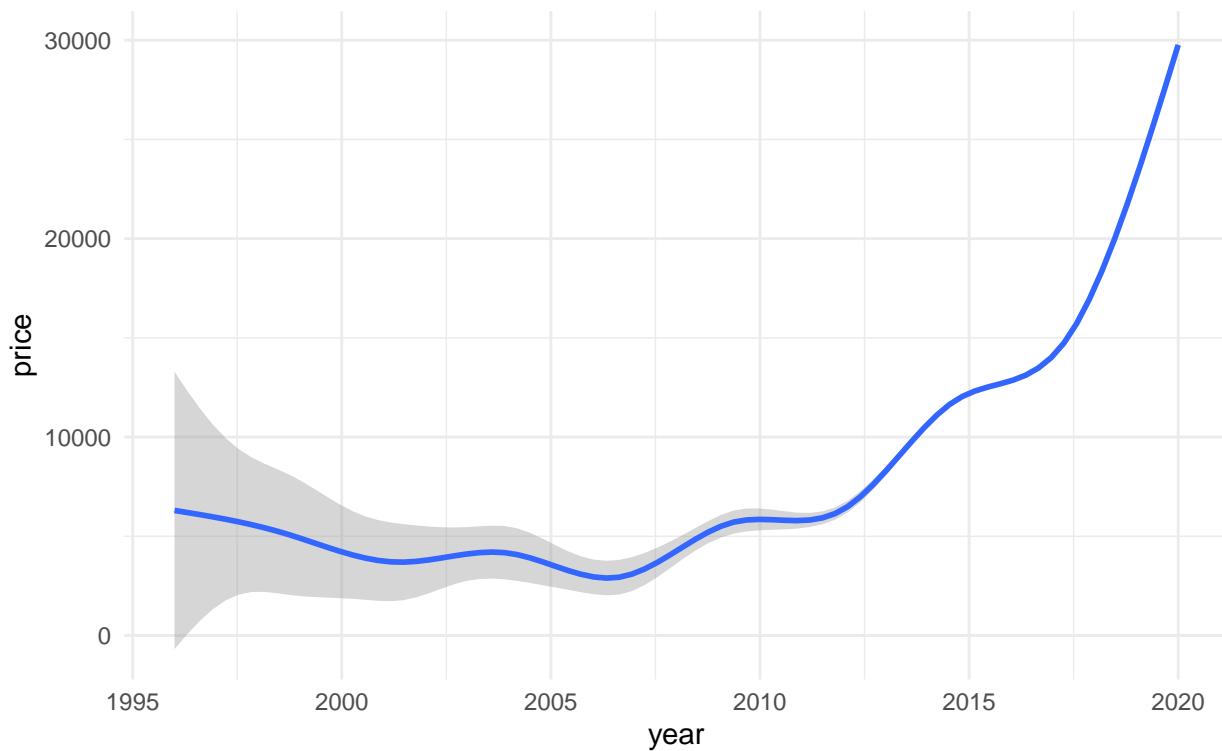
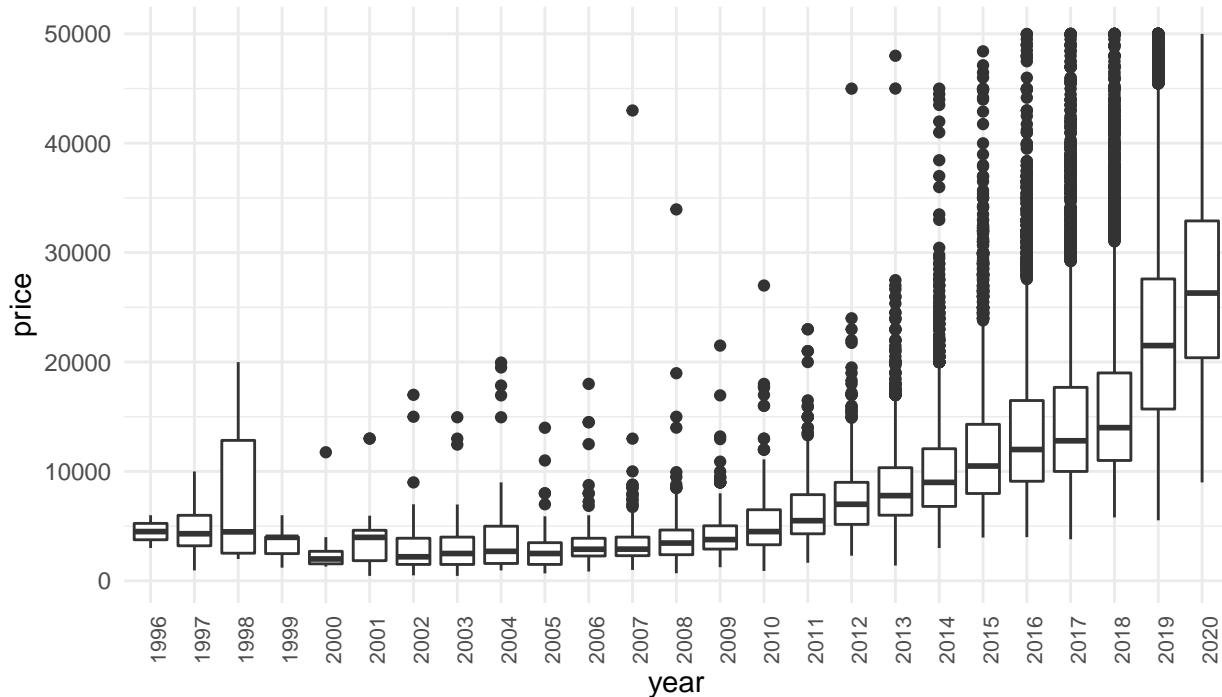
50% of total cars have price between 10,000 and 20,000 £.

2.2.11 Search price correlations

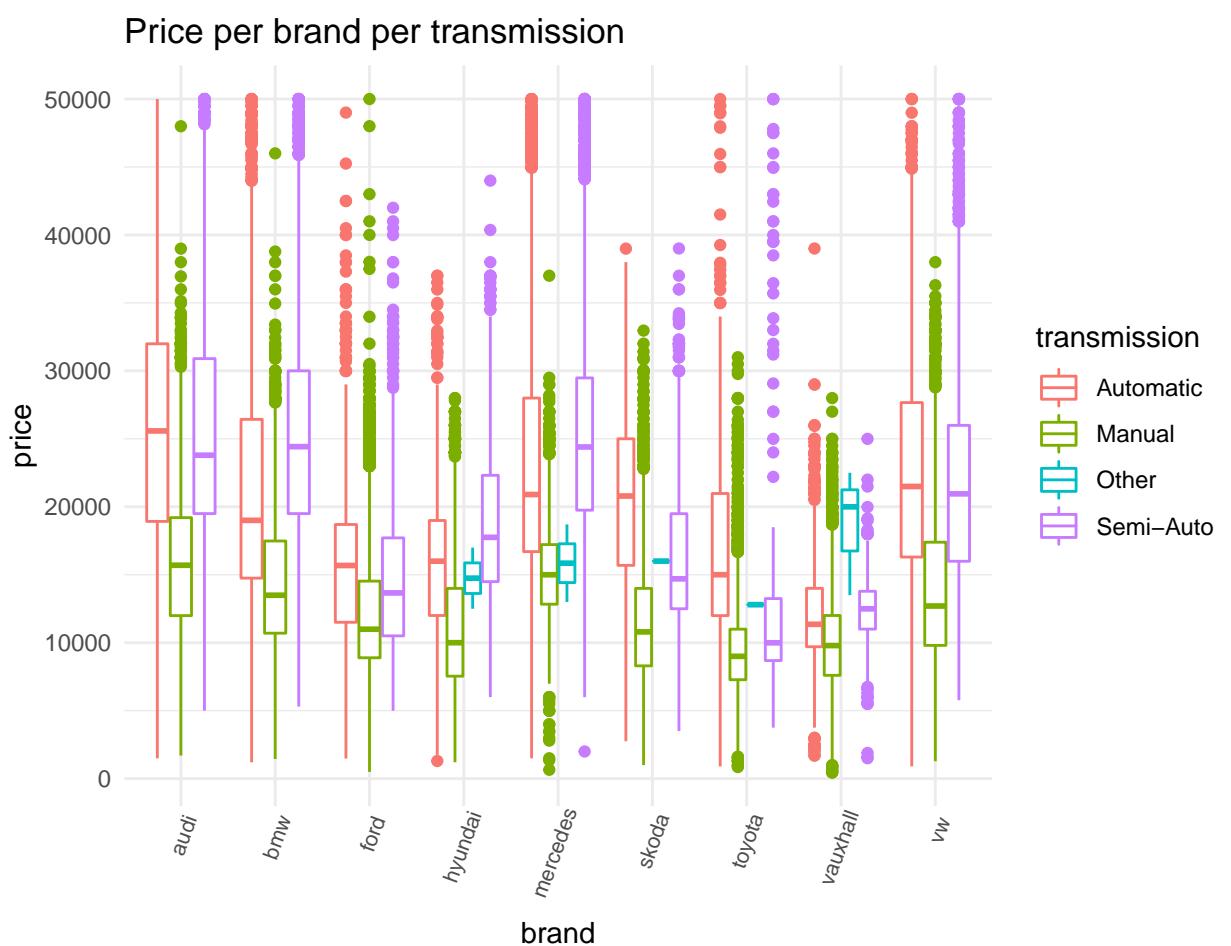
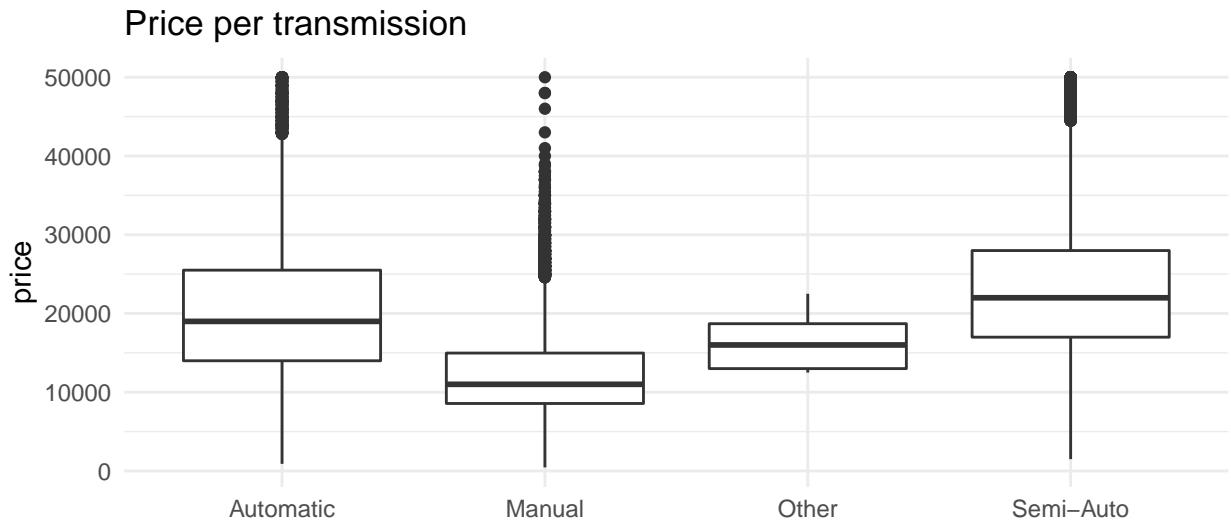


Some brand are more expensive in general.

Price per year

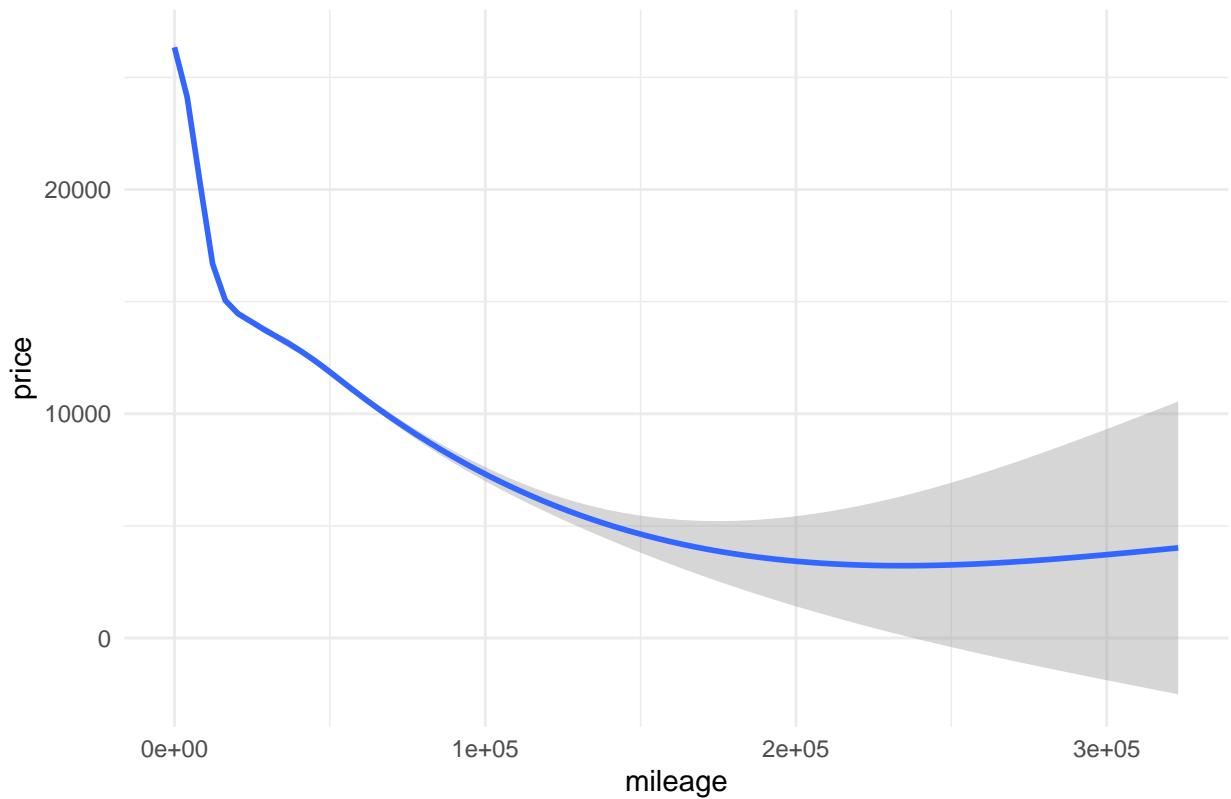


There is an obvious trend of price per year, the most a car is new the most it is expensive.



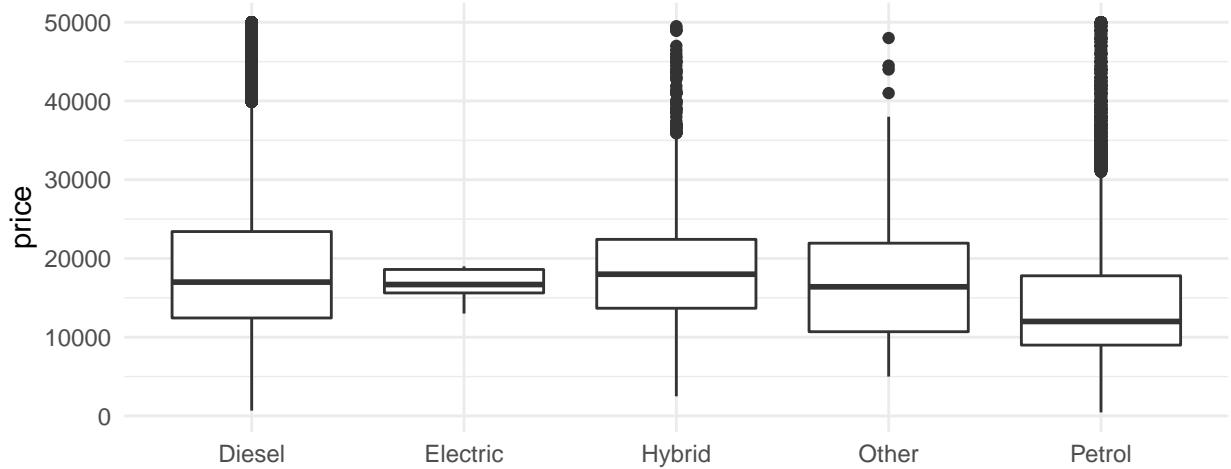
Generally, Automatic and Semi-Auto transmission are more expensive than Manual.

Price per mileage

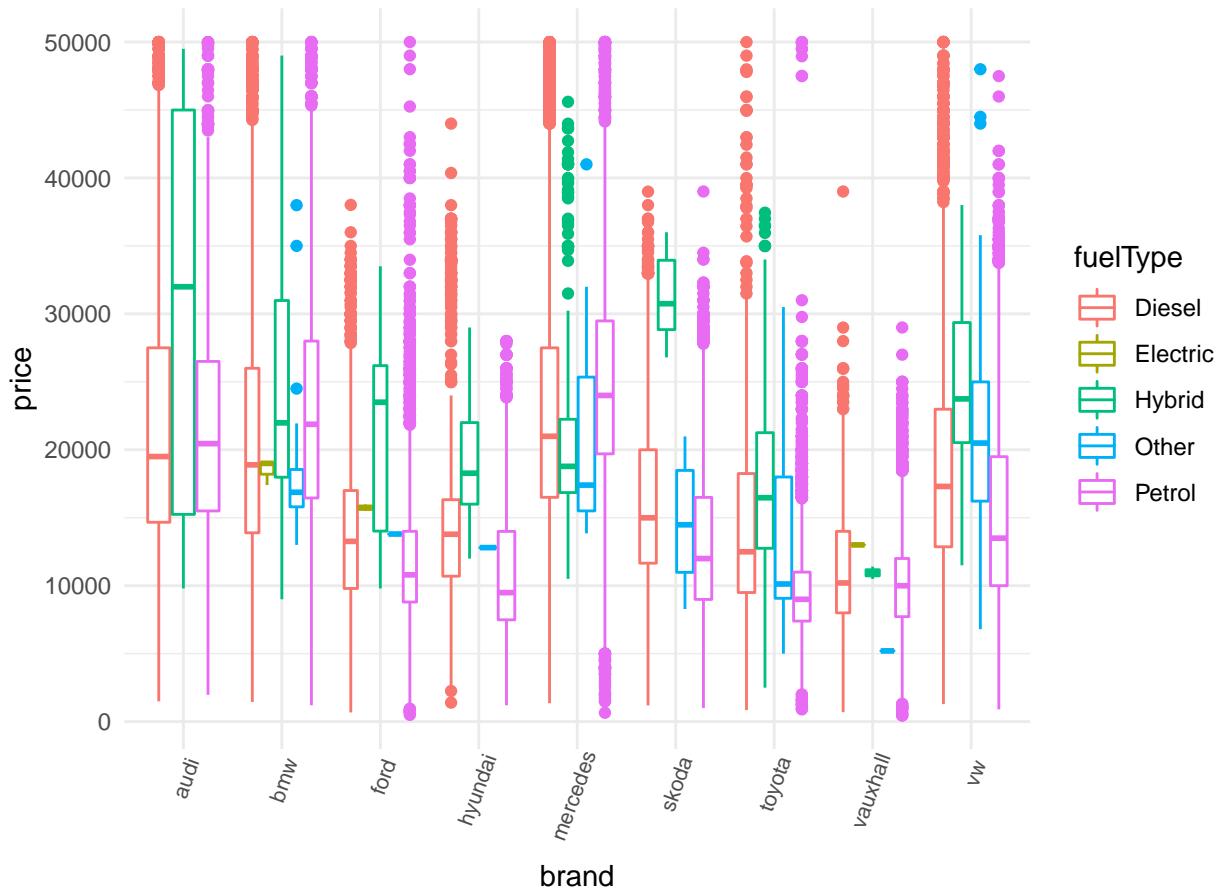


There is an inverse trend between mileage and price.

Price per fuelType

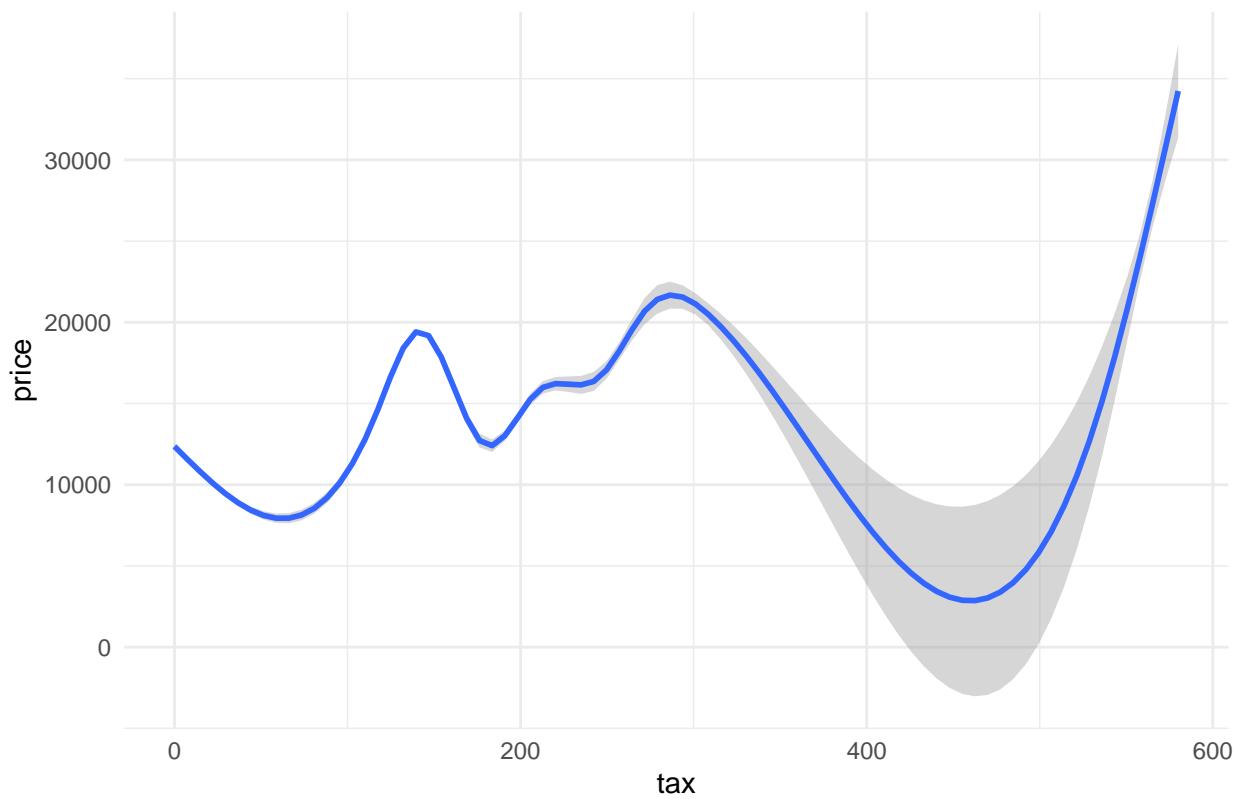


Price per brand per fuel type

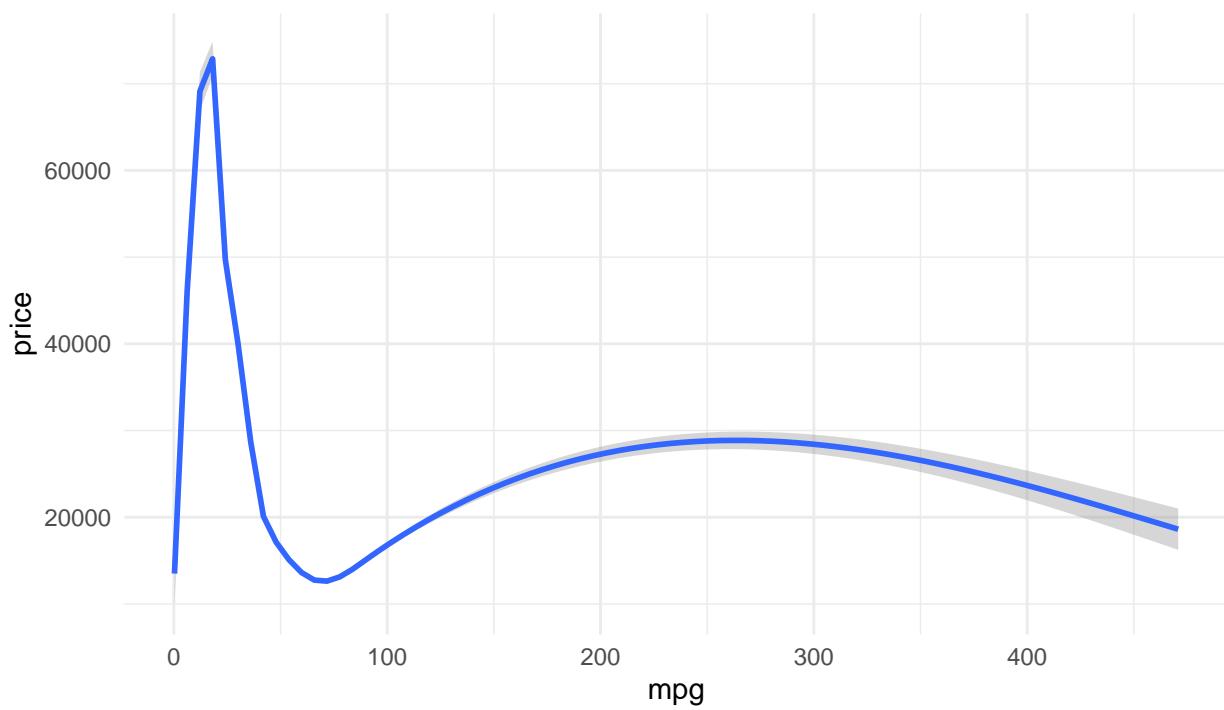


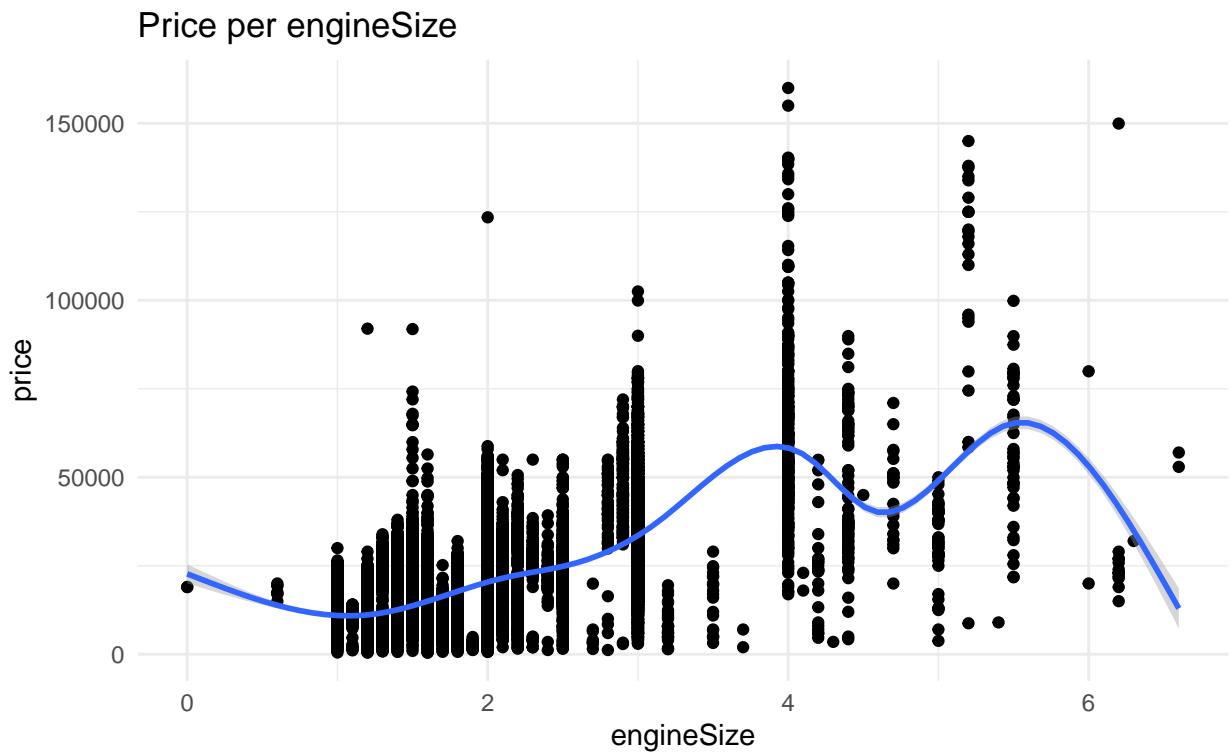
Petrol is less expensive than Diesel. But Mercedes and BMW are exceptions because of luxury petrol cars.

Price per tax



Price per mpg





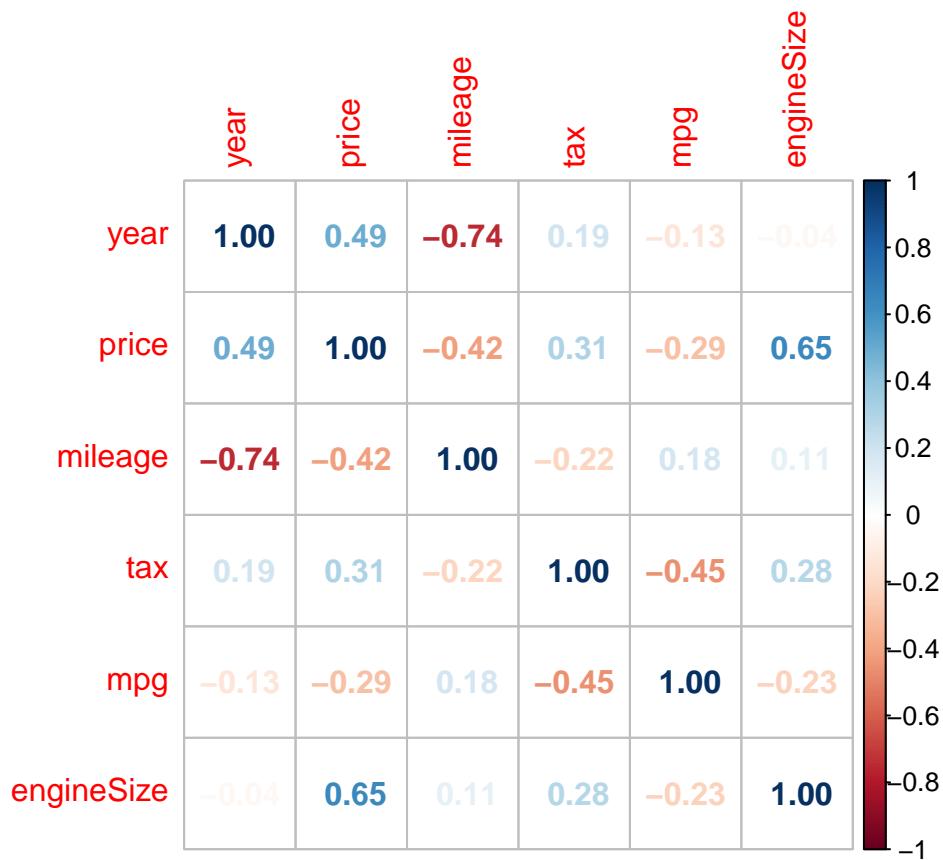
From the plots :

The engine size seems to be correlated to the price of the car although it's not a linear correlation.

There is no visible correlation with tax and price.

In general cars with high mpg cost less than cars with low mpg.

2.2.12 Corrplot



We will use most correlated variables to build a machine learning model step-by-step.

3 Results

3.1 Data preparation for a step-by-step approach

We add a column of mile_interval depending on mileage quartile from 1 to 4.

After that, we split cars data set to train_set and test_set. test_set will be 20% of cars.

We make sure car's model in test_set are also in train_set

3.2 First model

If we predicted all cars with the mean. The RMSE will be :

method	rmse
just the average	9638.289

model effect

If we group_by model after using "just the average" for predicting.

method	rmse
just the average	9638.289
model_effect	5918.040

year effect

If we group_by year after using "just the average" and model for predicting.

method	rmse
just the average	9638.289
model_effect	5918.040
year_effect	4413.738

engineSize effect

If we group_by engineSize after using "just the average", model and year for predicting.

method	rmse
just the average	9638.289
model_effect	5918.040
year_effect	4413.738
engineSize_effect	4076.645

mile_interval effect

If we group_by mile_interval after using "just the average", model, year and engineSize for predicting.

method	rmse
just the average	9638.289
model_effect	5918.040
year_effect	4413.738
engineSize_effect	4076.645
mile_interval_effect	4032.909

3.3 Data processing for next models

We modify the structure of the data to accurate next machine learning models.
We define the numeric columns :

```
## [1] "year"      "price"     "mileage"    "tax"       "mpg"  
## [6] "engineSize"
```

3.3.1 Transform categorical variables to binary

We add 4 columns representing the transmission classes and 5 columns representing the FuelType classes :

```
##   model year price mileage tax mpg engineSize Automatic Manual Semi-Auto  
## 1   A1 2017 12500 15735 150 55.4 1.4 0 1 0  
## 2   A6 2016 16500 36203 20 64.2 2.0 1 0 0  
## 3   A1 2016 11000 29946 30 55.4 1.4 0 1 0  
## 4   A4 2017 16800 25952 145 67.3 2.0 1 0 0  
## 5   A3 2019 17300 1998 145 49.6 1.0 0 1 0  
## 6   A1 2016 13900 32260 30 58.9 1.4 1 0 0  
## Diesel Electric Hybrid Other Petrol  
## 1 0 0 0 0 1  
## 2 1 0 0 0 0  
## 3 0 0 0 0 1  
## 4 1 0 0 0 0  
## 5 0 0 0 0 1  
## 6 0 0 0 0 1
```

And we add 186 columns representing the classes of car_model variable to binarize it. It can make execution slower.

```
##   year price mileage Diesel Other Insignia Meriva Puma  
## 1 2017 12500 15735 0 0 0 0 0  
## 2 2016 16500 36203 1 0 0 0 0  
## 3 2016 11000 29946 0 0 0 0 0  
## 4 2017 16800 25952 1 0 0 0 0  
## 5 2019 17300 1998 0 0 0 0 0  
## 6 2016 13900 32260 0 0 0 0 0
```

3.3.2 Scale continuous variables

```
##          year price mileage      tax      mpg engineSize  
## 1 -0.03281208 12500 -0.3546856 0.4716854 0.01112298 -0.4858527  
## 2 -0.50778249 16500 0.6183849 -1.5800019 0.55474553 0.6006778  
## 3 -0.50778249 11000 0.3209205 -1.4221798 0.01112298 -0.4858527  
## 4 -0.03281208 16800 0.1310415 0.3927743 0.74624893 0.6006778  
## 5  0.91712874 17300 -1.0077571 0.3927743 -0.34717371 -1.2102064  
## 6 -0.50778249 13900 0.4309305 -1.4221798 0.22733649 -0.4858527
```

3.4 We define train and test test using the same seed as before

3.5 Linear model with caret

We will train the model. It take 3 minutes to execute !

Then predict the outcomes.

And finally calculate RMSE.

method	rmse
just the average	9638.289
model_effect	5918.040
year_effect	4413.738
engineSize_effect	4076.645
mile_interval_effect	4032.909
lm	5333.349

Until now, linear model is more accurate than all previous.

3.6 rpart model

We will train the model.

Then predict the outcomes.

And finally calculate RMSE.

method	rmse
just the average	9638.289
model_effect	5918.040
year_effect	4413.738
engineSize_effect	4076.645
mile_interval_effect	4032.909
lm	5333.349
rpart	7545.082

Looks like that rpart doesn't work good.

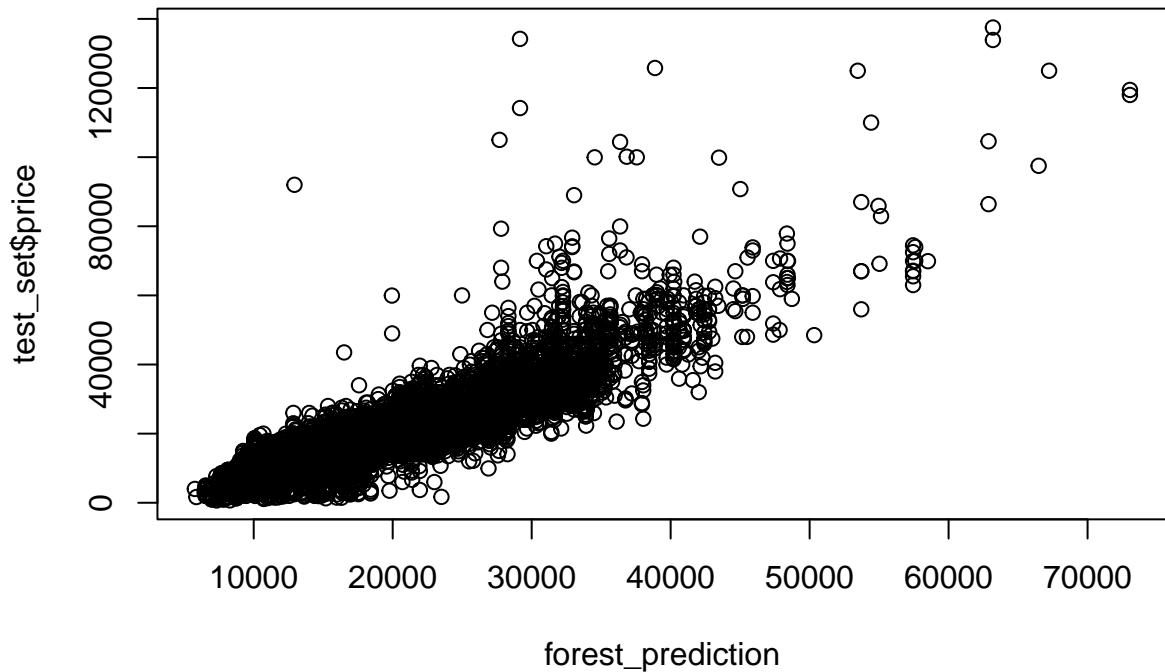
3.7 random forest

Another time, We will train the model. It take about 7 minutes !!

Then predict the outcomes.

And calculate RMSE.

method	rmse
just the average	9638.289
model_effect	5918.040
year_effect	4413.738
engineSize_effect	4076.645
mile_interval_effect	4032.909
lm	5333.349
rpart	7545.082
random_forest	4817.679



The RF is better than rpart but worse than lm, training the model more takes much more time.

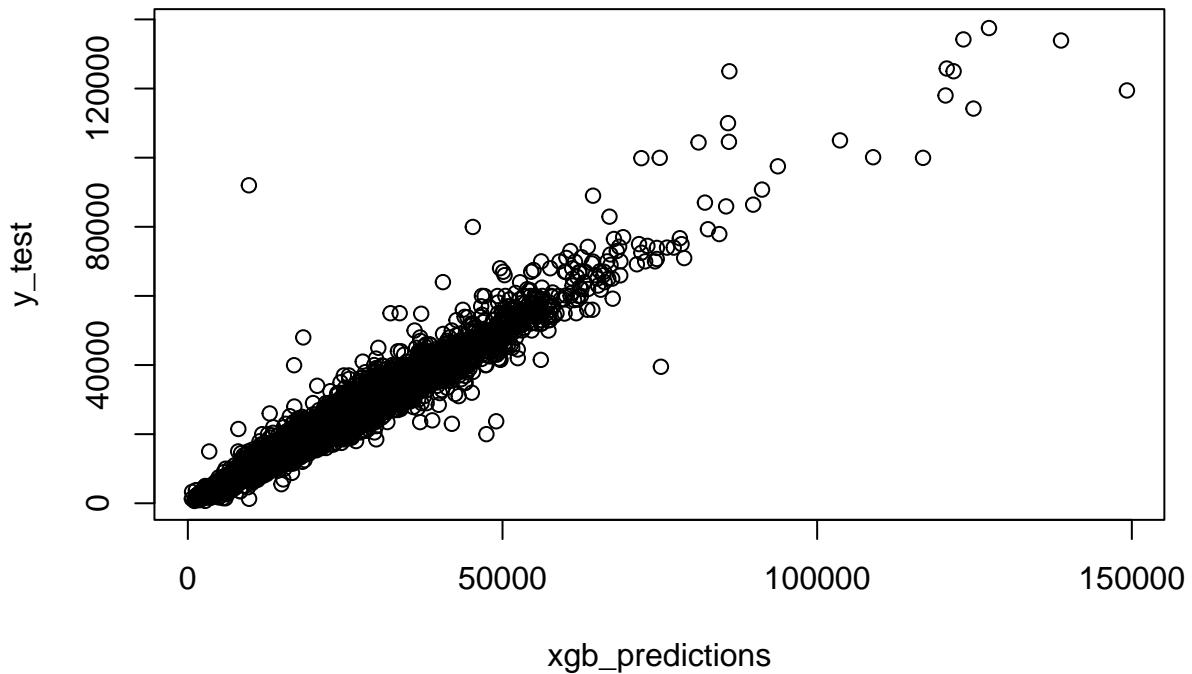
3.8 xgboost

We will train the model.

Then predict the outcomes.

And finally calculate RMSE (the Root Mean Square Error).

method	rmse
just the average	9638.289
model_effect	5918.040
year_effect	4413.738
engineSize_effect	4076.645
mile_interval_effect	4032.909
lm	5333.349
rpart	7545.082
random_forest	4817.679
xgboost	2066.609



It seems that the Xgboost has performed the best among the models we have been trying.

4 Conclusion

In this analysis we tried to build a system capable of generating used cars price based on cars characteristics. After the exploration we tried some models and We got different results depending on the approach used. xgboost was the most accurate among the others. Now let's compare some true_prices and predicted_prices we got with XGB:

true_price	predicted_price	absolute_difference
12500	13711	1211
16500	16285	215
12000	12850	850
16400	17177	777
17300	19666	2366
20200	22030	1830
19400	18512	888
15700	17214	1514
16600	19484	2884
12750	13345	595
16000	17148	1148
10200	10864	664
16000	14396	1604
11700	12171	471
16900	17585	685

true_price	predicted_price	absolute_difference
30700	34527	3827
26991	33025	6034
15995	16290	295
24495	23809	686
23999	27965	3966
25495	25634	139
26350	26002	348
30895	30712	183
32493	30228	2265
29990	24290	5700
22950	18188	4762
27990	24488	3502
32562	28622	3940
31291	26628	4663
21900	22033	133

Finally, for future work we could try matrix factorization for better prediction, the hope is to find info such as when is the ideal time to sell or buy certain cars (i.e. at what age and mileage are there significant drops in resale value).