Used cars market analysis and price predictor

from the kaggle´s used cars dataset

Gustavo Mayeregger

2020-05-02

Introduction

This work consists in explore, clean and analyze the used car market dataset. We have picked up the dataset from Kaggle[1], the dataset can be downloaded from here: https://www.kaggle.com/orgesleka/used-cars-database/download and consist in a dataset collected by Orges Leka of 360.000 cars offered on eBay in the German market.

It´s important emphasize that we wan´t to study the market of cars that are in function. This will be a problematic criterion in the cleaning process because the same car can cost 100€ or 5000€ depending on the condition.

Ones we have a cleaned dataset and the is analysis done we going to use a regression model to predict the price of a used car (in the German market). This is useful to have a reference price for be sure that the price we found for a used car is convenient or no.

Exploration

```
library(tufte)
# invalidate cache when the tufte version changes
knitr::opts_chunk$set(tidy = FALSE, cache.extra = packageVersion('tufte'))
options(htmltools.dir.version = FALSE)
library(tidyverse)
library(knitr)
library(lubridate)
library(readr)
library(corrplot)
library(caret)



#load the autos dataset
autos <- read_csv("data/autos.csv")
```

The autos datasets consist in 357687 publications on eBay collected with a script.

```
nrow(autos)
```

```
## [1] 354687
```

The dataset have 20 variables

[1] is the is the world's largest data science community with powerful tools and resources to help you achieve your data science goals.

```
names(autos)
```

```
##  [1] "dateCrawled"         "name"                "seller"
##  [4] "offerType"           "price"               "abtest"
##  [7] "vehicleType"         "yearOfRegistration"  "gearbox"
## [10] "powerPS"             "model"               "kilometer"
## [13] "monthOfRegistration" "fuelType"            "brand"
## [16] "notRepairedDamage"   "dateCreated"         "nrOfPictures"
## [19] "postalCode"          "lastSeen"
```

As we mention earlier this dataset is collected from a German eBay and all variables and data is in German language.

We decided work with the original data and not translate because almost all data is ease are in English and the few ones that are in German we will translate in this section.

dataCrawled

POSIXct variable: The date and time that the add is deleted on eBay. Dates are between:

```
min(autos$dateCrawled)
```

```
## [1] "2016-03-05 14:06:22 UTC"
```

```
max(autos$dateCrawled)
```

```
## [1] "2016-04-07 14:36:58 UTC"
```

name

Character variable: This is the field were the user of eBay complete the name of the add and put the more relevant info.

```
autos$name[1:4]
```

```
## [1] "Golf_3_1.6"                       "A5_Sportback_2.7_Tdi"
## [3] "Jeep_Grand_Cherokee_\"Overland\"" "GOLF_4_1_4__3T\xdcRER"
```

seller

Character variable: Contains two variables: gewerblich (car dealer) and privat (private seller)

```
table(autos$seller)
```

```
##
## gewerblich     privat
##          3     354684
```

offerType

Character variable: The options are Angebot (offered car) and Gesuch (requested car)

```
table(autos$offerType)
```

```
##
## Angebot  Gesuch
##  354675      12
```

price

Numeric variable: The price that the seller ask for the car

```
min(autos$price)
```

```
## [1] 0
```

```
max(autos$price)
```

```
## [1] 2147483647
```

abtest

Character variable: Here are two variables: control and test. This variable seems that is a internal variable used by eBay

```
table(autos$abtest)
```

```
##
## control    test
##  170648  184039
```

vehicleType

Character variable:

```
table(autos$vehicleType)
```

```
##
##     andere        bus     cabrio      coupe kleinwagen      kombi  limousine
##       3209      28811      21823      18106      76474      64533      91524
##        suv
##      14056
```

We see this types of vehicles:
abdere: (others) **bus:** (wagon) big utilitarians vehicles to transport persons or merchandise **cabrio:** sports cars without roof **coupe:** medium size cars with 3 doors **kleinwagen:** (small car) urban small cars **kombi:** (station) cars for a big family use **limousine:** (sedan) cars with 5 doors **suv:** sport utility vans

yearOfRegistration

Numeric variable: Year of the car registration

```
summary(autos$yearOfRegistration)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1000    1999    2003    2005    2008    9999
```

gearbox

Character variable:

```
table(autos$gearbox)
```

```
##
## automatik   manuell
##     73632    261777
```

automatic: automatic gearbox **manuell:** manual gearbox

powerPS

Numeric variable: Is the power of the car

```
summary(autos$powerPS)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0    70.0   105.0   115.5   150.0 20000.0
```

model

Character variable: There are 250 different models in the dataset For example:

```
unique(autos$model)[1:10]
```

```
##  [1] "golf"    NA        "grand"   "fabia"   "3er"     "2_reihe" "andere"
##  [8] "c_max"   "3_reihe" "passat"
```

kilometer

Numeric variable: The kilometers that the car have been driven

```
summary(autos$kilometer)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    5000  125000  150000  125606  150000  150000
```

monthOfRegistration

Numeric variable: Month of the car registration

```
table(autos$monthOfRegistration)
```

```
##
##     0     1     2     3     4     5     6     7     8     9    10    11    12
## 35979 23471 21411 34513 29552 29190 31671 27619 22632 23988 26099 24324 24238
```

fuelType

Character variable: Type of fuel used by the car

```
table(autos$fuelType)
```

```
##
##  andere  benzin     cng  diesel elektro  hybrid     lpg
##     197  213642     547  102917      99     266    5121
```

andere: others **benzin:** gasoline **cng:** compressed natural gas **diesel elektro:** electric car **hybrid:** car powered with both: electric and and gasoline engine **lpg:** liquefied petroleum gas

brand

Character variable: There are 39 different brands of cars in the dataset For example:

```
unique(autos$brand)[1:10]
```

```
##  [1] "volkswagen" "audi"       "jeep"       "skoda"      "bmw"
##  [6] "peugeot"    "ford"       "mazda"      "nissan"     "renault"
```

notRepairedDamage

Character variable: This variable tell us if the car have a damage to be repaired.

```
table(autos$notRepairedDamage)
```

```
##
##     ja   nein
##  34686 251224
```

ja: yes have a damage **nein:** no have a damage

dataCreated

POSIXct variable: The date and time that the add is posted on ebay. Dates are between:

```
min(autos$dateCreated)
```

```
## [1] "2014-03-10 UTC"
```

```
max(autos$dateCreated)
```

```
## [1] "2016-04-07 UTC"
```

nrOfPictures

Numeric variable: A internal variable used by eBay. All data is 0

```
table(autos$nrOfPictures)
```

```
##
##      0
## 354687
```

postalCode

Character variable: The postal code of the seller of the car to locate the place of the car.

```
autos$postalCode[1:10]
```

```
##  [1] "70435" "66954" "90480" "91074" "60437" "33775" "67112" "19348" "94505"
## [10] "27472"
```

lastSeen

POSIXct variable: The date and time that the add was saw for time on eBay. Dates are between:

```
min(autos$dateCreated)
```

```
## [1] "2014-03-10 UTC"
```

```
max(autos$dateCreated)
```

```
## [1] "2016-04-07 UTC"
```

DATA CLEANING

First in the dataset we see two columns that for sure are only used by eBay. Those columns are abtest and nrOfPictures (all zeros)

There is only 3 cars offered by dealers and the others all are private. The price by a dealer is generally higher than a private.

There is only 12 cars requested and generally when the car is requested the price is lower.

Also the postalCode variable is are not useful in our project

We will remove the 6 columns.

```
autos_clean <- autos %>% select(-abtest, -nrOfPictures, -seller, -offerType, -postalCode)
```

Cleaning yearOfRegistration

The years go from 1000 to 9999. Here are many errors for sure. Let´s take only the cars until the 100 years of age.

```
autos_clean <- autos_clean %>% filter(yearOfRegistration >= year(max(dateCreated)) - 100 &
                                      yearOfRegistration <= year(max(dateCreated)))
```

Generating the age variable

In the market of used car is very important know the age of the car. This we can calculate from the difference between the add in eBay and the year of the registration. When a buyer search a car look the age of the car in years, there is no useful have a monthOfRegistration variable.

```
autos_clean <- autos_clean %>%  mutate(age = year(dateCreated) - yearOfRegistration) %>%
  select(-monthOfRegistration)
```

Generating the daysOnEbay variable

Other data that can be useful is the days that the add were posted in eBay. this data we can obtain from the difference between the dataCrawled and the dataCreated.

The lastSeen variable is not useful for us

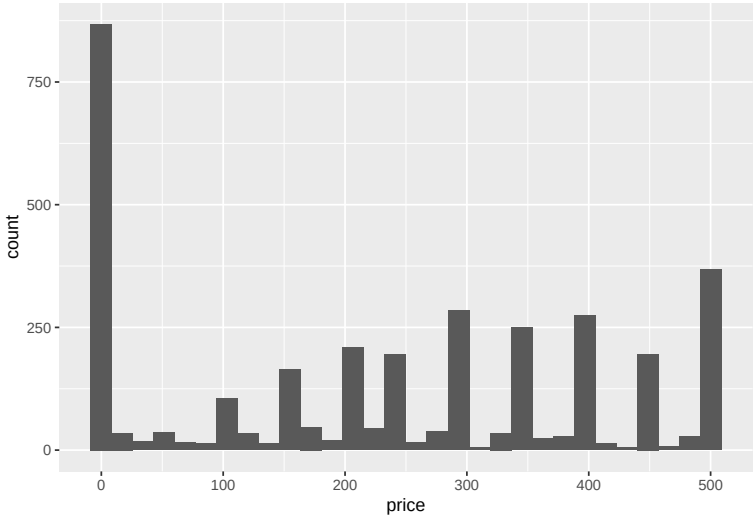
```
autos_clean <- autos_clean %>%
  mutate(daysOnEbay = round(as.numeric(difftime(dateCrawled, dateCreated,
                                                units = "days")))) %>%
  select(-dateCrawled, -dateCreated, -lastSeen)
```

Cleaning Price

In the prices columns we see prices from 0 to 2x10^9. A super car like a Bugatti or Ferrari cost 2 millions. Also we can ignore cars with the price lower than 100€ (usually only for repair).

First we will try cleaning the low prices zone inspecting the dataset for cars with prices under the 500€ and a age 4 years or below. We suppose that this are not functional cars.

```
autos_clean %>% filter(age<=4 & price<=500) %>% ggplot(aes(price)) + geom_histogram()
```



There are 3300 “cars” below 500€ and a age of 4 or less but 1000 of them have a 0 price. Also we will delete all cars below a price of 200€ because we need a functional car. We can delete this data for a first step and set a top limit of 300000€ for the price

```
autos_clean <- autos_clean %>% filter(price > 200 & price < 300000 &
                                      !(price <= 500 & age <=4))
```

Let´s inspect the remaining cars with a age of 2 or below and a price lower than 1000€ Looking in the names we found that exist many cars in Leasing or “financierung” (financed), this explain the lower prices. Also there are names like “zum schlachten” (to sacrifice) or “schaden” (damaged) or “bastler” (hobbyist car). There are so many financed cars, damaged cars, parts and other merchandising related to the car market in the range under the 200€ that we can eliminate all data below the 500€

Let´s look in all names the leasing and financed cars

```
key_words <- c("leasing", "finanzierung")
autos_clean %>% filter(str_detect(str_to_lower(autos_clean$name),
                                  paste(key_words, collapse = "|"))) %>%
  select(brand, model, price, age) %>% arrange(desc(price)) %>% slice(70:80) %>% kable()
```

brand	model	price	age
nissan	andere	4950	14
opel	astra	4500	11
volkswagen	passat	4299	14
chrysler	voyager	3800	14
bmw	1er	3000	1
bmw	1er	3000	1
mitsubishi	andere	2900	13
chevrolet	andere	2599	9
hyundai	andere	2599	9
bmw	x_reihe	2500	2
bmw	1er	2500	2

There are very new financed or leasing cars and the prices sometimes are the cash and others the quotes We can delete all the cars with 2 or less years and the price below 3000€

```
autos_clean <- autos_clean %>% filter(!(str_detect(str_to_lower(autos_clean$name),
                                                   paste(key_words, collapse = "|")) &
```

```
                                        price <= 3000 & age <= 2))
```

Now let´s eliminate all damage and hobbyist cars

```
key_words <- c("schlachten", "schaden", "bastler")
autos_clean <- autos_clean %>% filter(!(str_detect(str_to_lower(autos_clean$name),
                                                   paste(key_words, collapse = "|"))))
```

Cleaning model

```
autos_clean %>% filter(is.na(model) | model=="andere") %>% nrow()
```

```
## [1] 36885
```

We see 42121 “andere” or NA into the model variable, is a great amount of data without the model.

In the name variable that is a free camp filled by the user with the relevant information and the model is there in almost all cases.

We will search the the model of the car in the name variable and if we find the model then store it in the missing model.

```
autos_clean$model[autos_clean$model == "andere"] <- NA

ind_no_model <- which(is.na(autos_clean$model))
all_models <- unique(na.omit(autos_clean$model))

search_model_in_name <- function(name){
  ind <- which(all_models %in% str_split(str_to_lower(name), pattern = "_")[[1]])
  ifelse(length(ind) == 1, return(all_models[ind]), return(NA))
}

autos_clean$model[ind_no_model] <- sapply(autos_clean$name[ind_no_model],
                                          search_model_in_name, USE.NAMES = FALSE)
```

If we look now the NAs in the model variable we have

```
sum(is.na(autos_clean$model))
```

```
## [1] 28305
```

We recuperate 10083 missing models and all the unknown models are NA and we need to remove them

```
autos_clean <- autos_clean %>% drop_na(model)
```

Cleaning brand

We can do the same method of the model cleaning searching the brand in the name variable. There aren´t NA in the brand but if we look the dataset there is a “sonstige_autos” (others cars)

```
ind_no_brand <- which(autos_clean$brand == "sonstige_autos")
length(ind_no_brand)
```

```
## [1] 286
```

We have 320 data with no brand Let´s search in the name variable and replace if we find

```
all_brands <- unique(autos_clean$brand[-ind_no_brand])

search_brand_in_name <- function(name){
```

```
  ind <- which(all_brands %in% str_split(str_to_lower(name), pattern = "_")[[1]])
  ifelse(length(ind) == 1, return(all_brands[ind]), return(NA))
}

autos_clean$brand[ind_no_brand] <- sapply(autos_clean$name[ind_no_brand],
                                          search_brand_in_name, USE.NAMES = FALSE)
```

```
sum(is.na(autos_clean$brand))
```

```
## [1] 246
```

We recuperate 53 missing brands and now we need delete the NAs

```
autos_clean <- autos_clean %>% drop_na(brand)
```

Generating the brandModel variable to help the cleaning process

Now that we have the brand and model clean we can create a variable that combine both. This variable is useful to the data exploring and analysis. Only the brand sometimes is not sufficient, into a brand exist many models with a great spread of prices.

```
autos_clean <- autos_clean %>% mutate(brandModel = paste(brand, model, sep = "_"))
```

Generating the vehicleClass variable

When a user search a car to buy the first thing he do is decide the class of the vehicle he is needing.

The class of a vehicle is related to the prestige of the brand. A used car can be affordable but later the cost of the parts of a high class vehicle can cost more than the entire vehicle.

Also having a class type can help in the cleaning and analysis process.

Add the vehicle class column: high, mid_high, mid, mid_low, low

```
autos_clean <- autos_clean %>%
  mutate( vehicleClass = case_when(brand %in% c("mercedes_benz", "bmw", "audi", "jaguar",
                                                "porsche", "land_rover", "saab",
                                                "jeep") ~ "high",
                                   brand %in% c("volvo", "chrysler", "mini",
                                                "volkswagen", "alfa_romeo",
                                                "subaru") ~ "mid_high",
                                   brand %in% c("ford", "mazda", "nissan", "opel",
                                                "honda", "toyota", "skoda", "seat",
                                                "mitsubishi", "suzuki") ~ "mid",
                                   brand %in% c("chevrolet", "kia", "hyundai", "fiat",
                                                "lancia", "citroen", "peugeot", "renault",
                                                "smart") ~ "mid_low",
                                   brand %in% c("daewoo", "trabant", "lada",
                                                "rover", "daihatsu", "dacia") ~ "low"
                                  ))
```

Cleaning gearbox

Same process like the model and brand searching the gearbox in the name

```
ind_no_gear <- which(is.na(autos_clean$gearbox))
length(ind_no_gear)
```

```
## [1] 10318
```

There is 13654 datas without the gearbox variable

```
all_gear <- unique(autos_clean$gearbox[-ind_no_gear])

search_gear_in_name <- function(name){
  ind <- which(all_gear %in% str_split(str_to_lower(name), pattern = "_")[[1]])
  ifelse(length(ind) == 1, return(all_gear[ind]), return(NA))
}

autos_clean$gearbox[ind_no_gear] <- sapply(autos_clean$name[ind_no_gear],
                                           search_gear_in_name, USE.NAMES = FALSE)
```

```
sum(is.na(autos_clean$gearbox))
```

```
## [1] 10198
```

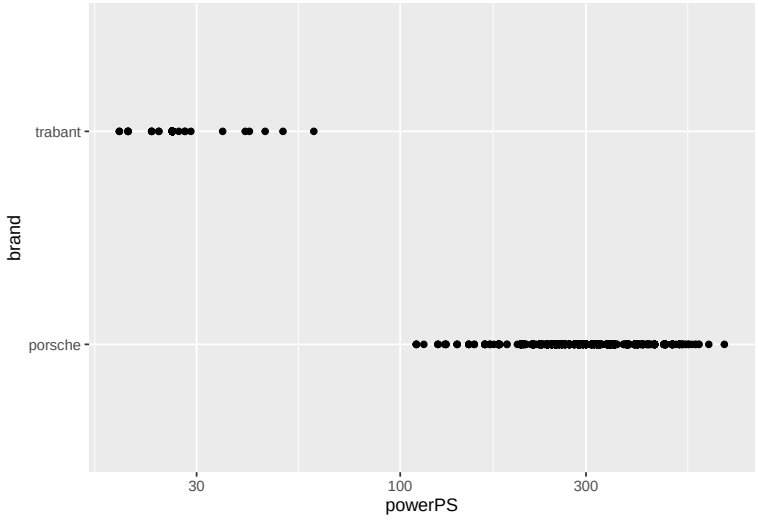
After the search we found 137 missing gearbox in the name variable. All the unknown are NA and we need to remove them

```
autos_clean <- autos_clean %>% drop_na(gearbox)
```

Cleaning powerPS

#In the brands we found trabant and porsche like the brands with less and more power in their cars.

```
autos_clean %>% filter(brand %in% c("trabant", "porsche") & powerPS >= 10) %>%
  ggplot(aes(powerPS, brand)) + geom_point() + scale_x_log10()
```

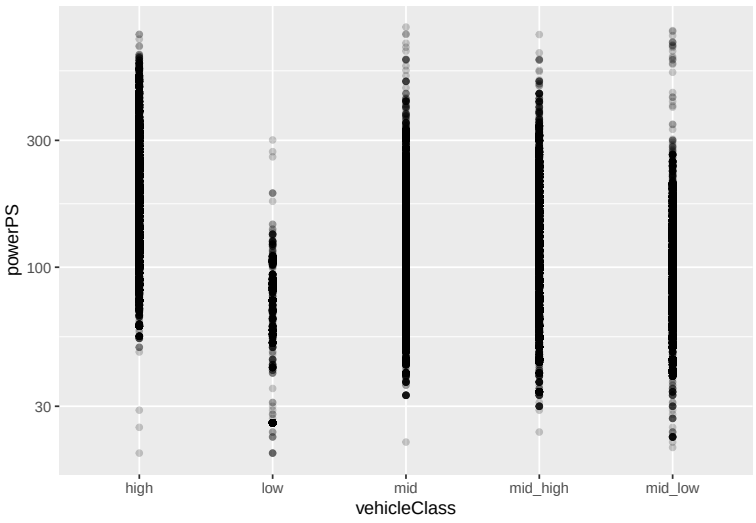


Considering this we can limit the powerPS variable between 20 and 800

```
autos_clean <- autos_clean %>% filter(powerPS >= 20 & powerPS <= 800)
```

Plot of the powerPS per vehicleClass

```
autos_clean %>% ggplot(aes(vehicleClass, powerPS)) + geom_point(alpha=0.2) +
  scale_y_log10()
```



Here can see some inconsistencies

If we inspect the data we find:

Less power in high class

```
autos_clean %>% filter(vehicleClass == "high") %>%
  select(brandModel, powerPS) %>% top_n(-10, powerPS) %>% arrange(powerPS) %>% kable()
```

brandModel	powerPS
mercedes_benz_c_klasse	20
audi_a6	25
mercedes_benz_c_klasse	29
mercedes_benz_b_klasse	48
land_rover_defender	50
mercedes_benz_a_klasse	50
audi_polo	50
audi_80	54
audi_80	54
mercedes_benz_200	54
audi_80	54
mercedes_benz_200	54
mercedes_benz_200	54

Most power in mid_high class

```
autos_clean %>% filter(vehicleClass == "mid_high") %>%
  select(brandModel, powerPS) %>% top_n(10, powerPS) %>% arrange(desc(powerPS)) %>% kable()
```

brandModel	powerPS
volkswagen_polo	750
volkswagen_golf	640
volkswagen_polo	606
volkswagen_golf	603
volkswagen_polo	601
volkswagen_polo	601
volkswagen_golf	550

brandModel	powerPS
volkswagen_golf	544
volkswagen_golf	506
volkswagen_golf	502

Less power in mid_high class

```
autos_clean %>% filter(vehicleClass == "mid_high") %>%
  select(brandModel, powerPS) %>% top_n(-10, powerPS) %>% arrange(powerPS) %>% kable()
```

brandModel	powerPS
volkswagen_kaefer	24
volkswagen_polo	29
volkswagen_kaefer	30
volkswagen_kaefer	30
volkswagen_kaefer	30
volkswagen_kaefer	30
volkswagen_kaefer	30
volkswagen_kaefer	30
volkswagen_kaefer	30
volkswagen_kaefer	30
volkswagen_kaefer	30

Most power in mid class

```
autos_clean %>% filter(vehicleClass == "mid") %>%
  select(brandModel, powerPS) %>% top_n(10, powerPS) %>% arrange(desc(powerPS)) %>% kable()
```

brandModel	powerPS
opel_astra	800
ford_fiesta	754
ford_fiesta	750
skoda_fabia	696
ford_mustang	672
ford_mustang	650
opel_corsa	606
ford_mustang	604
opel_corsa	604
opel_corsa	603

Most power in mid_low class

```
autos_clean %>% filter(vehicleClass == "mid_low") %>%
  select(brandModel, powerPS) %>% top_n(10, powerPS) %>% arrange(desc(powerPS)) %>% kable()
```

brandModel	powerPS
fiat_punto	776
fiat_punto	771
citroen_c2	743
fiat_500	703

brandModel	powerPS
fiat_500	702
renault_twingo	700
renault_kangoo	685
peugeot_100	682
fiat_500	678
chevrolet_matiz	671

Most power in low class

```
autos_clean %>% filter(vehicleClass == "low") %>%
  select(brandModel, powerPS) %>% top_n(10, powerPS) %>% arrange(desc(powerPS)) %>% kable()
```

brandModel	powerPS
daihatsu_cuore	301
rover_rangerover	272
rover_mustang	260
rover_rangerover	190
rover_rangerover	190
rover_discovery	190
rover_freelander	177
rover_defender	145
daihatsu_sirion	145
daewoo_nubira	139
daewoo_nubira	139

Viewing this data we find many errors between the rover and land_rover brands To fix this we can only include the rover 200 and discard all the others with the rover brand

```
autos_clean <- autos_clean %>% filter(!(brand=="rover" & model != "200"))
```

Using the data we inspected in the most and less powered cars in the different classes we can clear the errors. - In high class delete all cars with more power than 50HP (mercedes benz a class), - In the mid_high class delete all cars below 550HP (golf VR6) and above 30HP (vw kaefer), - In the mid class delete all cars below 550HP (ford mustang), - In the mid low class delete all cars below 290HP (renault megane) and - In the low class delete all cars below 145HP (daihatsu sirion)

```
autos_clean <- autos_clean %>% filter((vehicleClass == "high" & powerPS >= 50) |
                                      (vehicleClass == "mid_high" & powerPS <= 550) |
                                      (vehicleClass == "mid_high" & powerPS >= 30) |
                                      (vehicleClass == "mid" & powerPS <= 550) |
                                      (vehicleClass == "mid_low" & powerPS <= 290) |
                                      (vehicleClass == "low"  & powerPS <= 145))
```

Cleaning vehicleType

```
sum(is.na(autos_clean$vehicleType) | autos_clean$vehicleType == "andere")
```

```
## [1] 8367
```

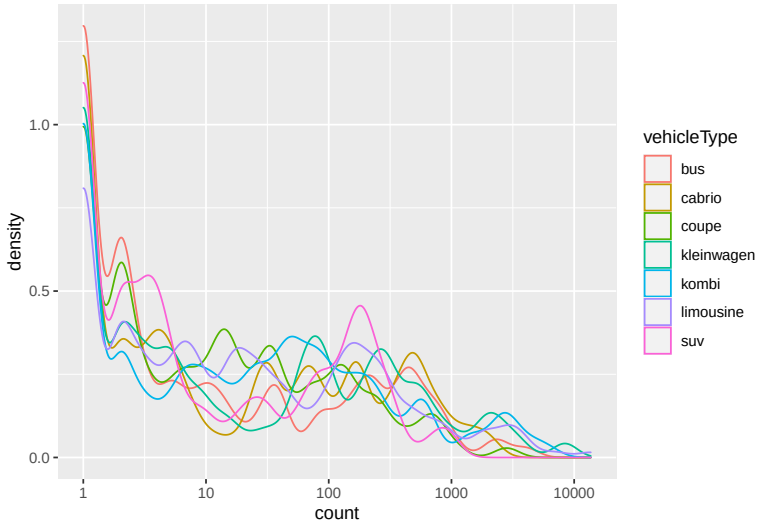
We have 8367 “andere” (others) and NAs in this variable

All we can do is remove the NA and “andere”

```
autos_clean <- autos_clean %>% filter(vehicleType != "andere" & !is.na(vehicleType))
```

Let´s look a pic with the vehicle types and the number of individual models into them

```
autos_clean %>% group_by(vehicleType, brandModel) %>% summarize(count=n()) %>%
  ggplot(aes(count, color=vehicleType)) + geom_density(bw=0.1) + scale_x_log10()
```



We can find that there is a lot of unique models into each type of cars

Let´s look for example in the kleinwagen (small cars) type the models that count only 1.

```
autos_clean %>% group_by(vehicleType, brandModel) %>% summarize(count=n()) %>%
  filter(vehicleType=="kleinwagen", count == 1) %>% .$brandModel
```

```
##  [1] "audi_80"               "audi_a4"               "audi_a6"
##  [4] "audi_golf"             "audi_polo"             "citroen_twingo"
##  [7] "daihatsu_terios"       "fiat_100"              "fiat_900"
## [10] "fiat_croma"            "fiat_doblo"            "fiat_fiesta"
## [13] "fiat_ypsilon"          "ford_kuga"             "honda_accord"
## [16] "hyundai_500"           "jaguar_s_type"         "lada_niva"
## [19] "mazda_5_reihe"         "mazda_6_reihe"         "mazda_90"
## [22] "mazda_rx_reihe"        "mercedes_benz_e_klasse" "mercedes_benz_fortwo"
## [25] "mercedes_benz_s_klasse" "mercedes_benz_vito"    "mini_90"
## [28] "mitsubishi_carisma"    "nissan_primera"        "opel_90"
## [31] "opel_antara"           "opel_calibra"          "opel_fox"
## [34] "opel_omega"            "opel_vivaro"           "peugeot_4_reihe"
## [37] "peugeot_cc"            "peugeot_golf"          "seat_toledo"
## [40] "skoda_up"              "smart_golf"            "smart_polo"
## [43] "smart_twingo"          "subaru_impreza"        "subaru_kaefer"
## [46] "suzuki_grand"          "suzuki_jimny"          "toyota_verso"
## [49] "trabant_fiesta"        "volkswagen_sharan"
```

There are many errors like subaru_impreza, suzuki_grand, mercedes_benz_e_klasse, trabant_fiesta, audi_a4, audi_16 and many others.

With this in consideration we can think that this unique brandModel entries are error and we will remove all these unique entries from all the vehicle types

```
autos_clean <- autos_clean %>% group_by(vehicleType, brandModel) %>%
  mutate(count=n()) %>% filter(n() != 1) %>% select(-count) %>% ungroup()
```

Cleaning fuelType

```
sum(is.na(autos_clean$fuelType) | autos_clean$fuelType=="andere")
```

```
## [1] 7965
```

We have 7965 andere and NAs in the fuelType variable

With the same process like the model, brand and gearbox we will search the fuelType in the name

```
autos_clean$fuelType[autos_clean$fuelType == "andere"] <- NA
ind_no_fuel <- which(is.na(autos_clean$fuelType))
all_fuel <- unique(autos_clean$fuelType[-ind_no_fuel])

search_fuel_in_name <- function(name){
  ind <- which(all_fuel %in% str_split(str_to_lower(name), pattern = "_")[[1]])
  ifelse(length(ind) == 1, return(all_fuel[ind]), return(NA))
}

autos_clean$fuelType[ind_no_fuel] <- sapply(autos_clean$name[ind_no_fuel],
                                            search_fuel_in_name, USE.NAMES = FALSE)
```

```
sum(is.na(autos_clean$fuelType))
```

```
## [1] 7609
```

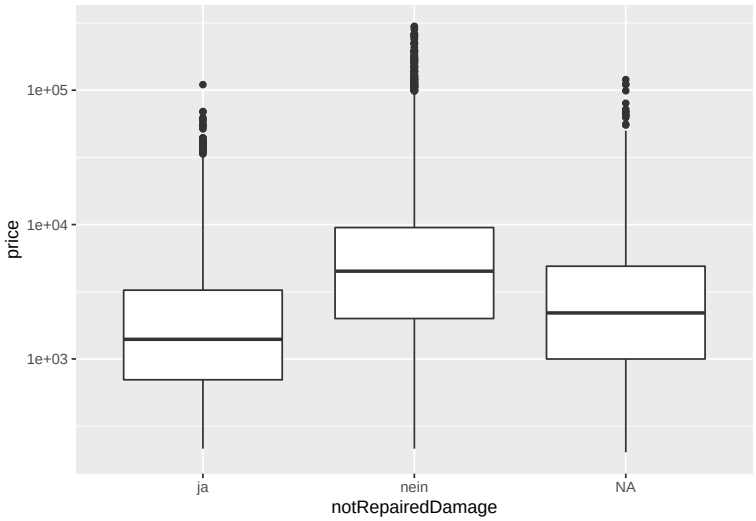
After the search we found 86 missing fuelType in the name variable. All the unknown are NA and we need to remove them

```
autos_clean <- autos_clean %>% drop_na(fuelType)
```

Cleaning the notRepairedDamage

#Plot of the notRepairedDamage vs the price variable

```
autos_clean %>% ggplot(aes(notRepairedDamage, price)) + geom_boxplot() + scale_y_log10()
```



From the plot we can see that the avg price of a damage vehicle is near 2000€, the avg price for the not damaged is near 8000$ and the NA are in the middle with a avg price of 4000€.

With this we conclude that a damaged car is 1/4 the price of an undamaged. Our dataset will only include the not damaged cars because the price of a damaged car is unpredictable.

The NAs data has 1/2 the price of an undamaged car. There is a great chance that are included damaged cars in the NAs data. Also we will not include the NAs.

Deleting the damaged cars and the notRepairedDamage NAs and then removing the column

```
autos_clean <- autos_clean %>% filter(notRepairedDamage != "ja" & !is.na(notRepairedDamage)) %>%
  select(-notRepairedDamage)
```

Deleting models with low offers

Let´s count the models and peak the ones with 3 or less cars

```
autos_clean %>% group_by(brandModel) %>% summarize(count=n()) %>% filter(count <= 3) %>%
  .$brandModel
```

```
##  [1] "alfa_romeo_a6"          "alfa_romeo_grand"       "audi_c4"
##  [4] "audi_up"                "bmw_cooper"             "bmw_i3"
##  [7] "bmw_up"                 "chevrolet_c3"           "dacia_90"
## [10] "fiat_90"                "ford_100"               "ford_90"
## [13] "ford_cl"                "ford_roadster"          "hyundai_grand"
## [16] "jaguar_roadster"        "jeep_100"               "jeep_a1"
## [19] "kia_90"                 "lada_samara"            "lancia_elefantino"
## [22] "lancia_spider"          "land_rover_serie_3"     "mazda_up"
## [25] "mercedes_benz_ka"       "mercedes_benz_roadster" "mitsubishi_200"
## [28] "nissan_80"              "opel_100"               "opel_a2"
## [31] "opel_gl"                "peugeot_ducato"         "peugeot_move"
## [34] "renault_80"             "renault_transporter"    "saab_90"
## [37] "saab_gl"                "seat_caddy"             "skoda_up"
## [40] "suzuki_80"              "suzuki_90"              "toyota_fortwo"
## [43] "toyota_gl"              "volkswagen_90"          "volkswagen_roadster"
## [46] "volkswagen_sprinter"    "volvo_80"
```

As we can see there are many errors in this table and we can eliminate this entries.

Also in our protect we are interested in search prices of popular cars that are offered in eBay and a car with low offers can´t be a reference in price for us.

For this reason we will conserve only the models with more than 3 cars offered on eBay.

```
autos_clean <- autos_clean %>% group_by(brandModel) %>% mutate(count=n()) %>%
  filter(count > 3) %>% ungroup() %>% select(-count)
```

Cleaned dataset summary

After all the cleaning process we have this 14 variables:

```
names(autos_clean)
```

```
##  [1] "name"               "price"              "vehicleType"
##  [4] "yearOfRegistration" "gearbox"            "powerPS"
##  [7] "model"              "kilometer"          "fuelType"
## [10] "brand"              "age"                "daysOnEbay"
## [13] "brandModel"         "vehicleClass"
```

And we have now 198457 cars of the original 354697 raw dataset

```
nrow(autos_clean)
```

```
## [1] 198457
```

DATA ANALYSIS

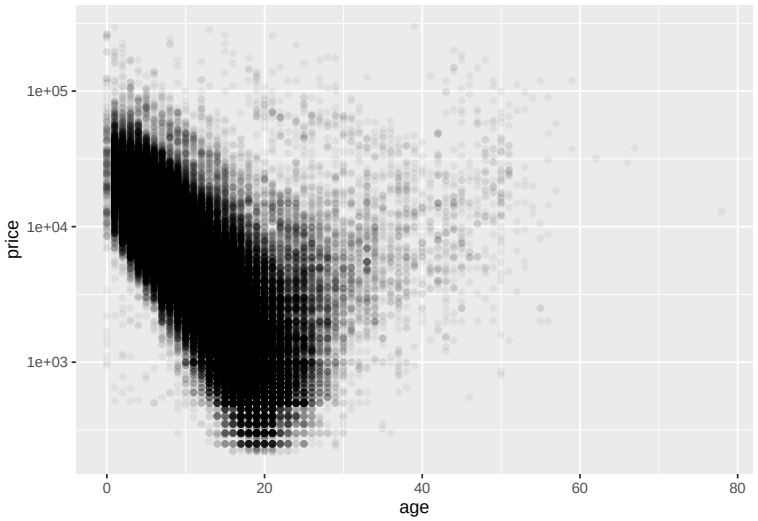
In this section we will analyze the relation that have the variables with the price, that is the variable more important for us for our predictor algorithm.

Price vs age

A very important factor to buy a car is the age of the car. As we know, a 0km car in only a day can loose 20% in value. But also we know that cars with a high age can increase their values.

Lets make a graph of all dataset cars vs their ages

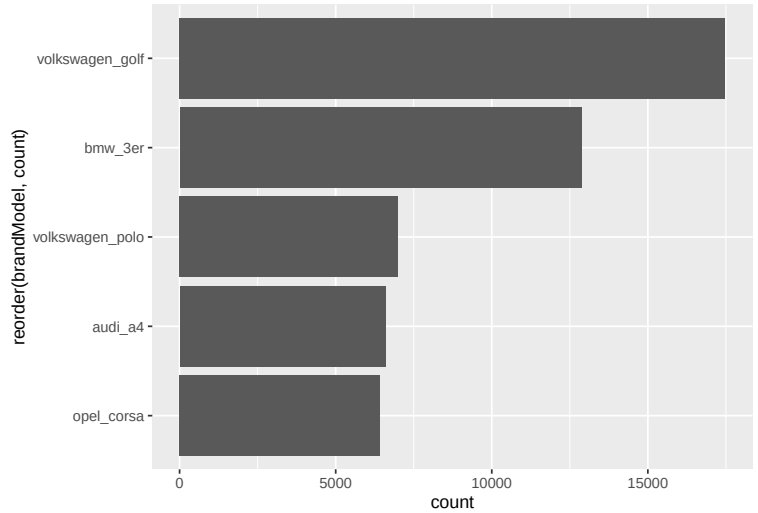
```
autos_clean %>% ggplot(aes(age, price)) + geom_point(alpha=0.05) + scale_y_log10()
```



We can see clearly that the price in general is decreasing 10 times in the first 20 years and then begins to rise.

Let´s find the 5 most offered models to analyze their prices in time.

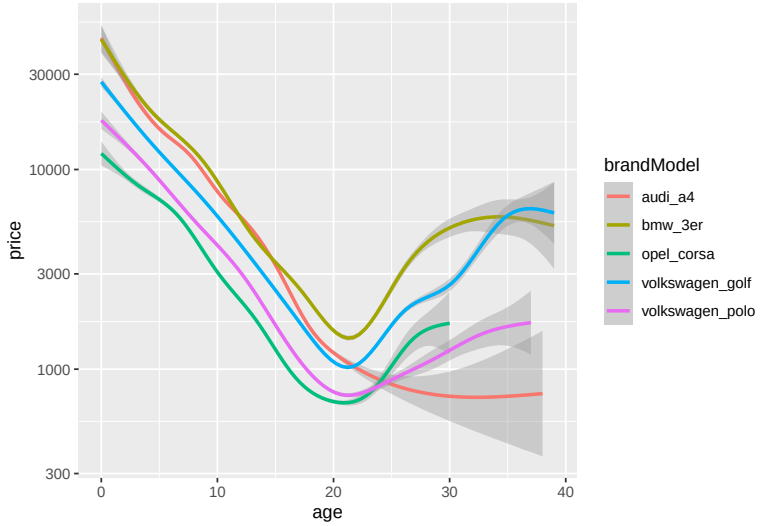
```
most_offer_models <- autos_clean %>% group_by(brandModel) %>% summarize(count=n()) %>%
  top_n(5, count) %>% arrange(desc(count))

most_offer_models %>% ggplot(aes(count, reorder(brandModel, count))) + geom_col()
```



This 5 models can be very representative because we have high, mid and low class cars

Let´s view what happened with this models prices in time

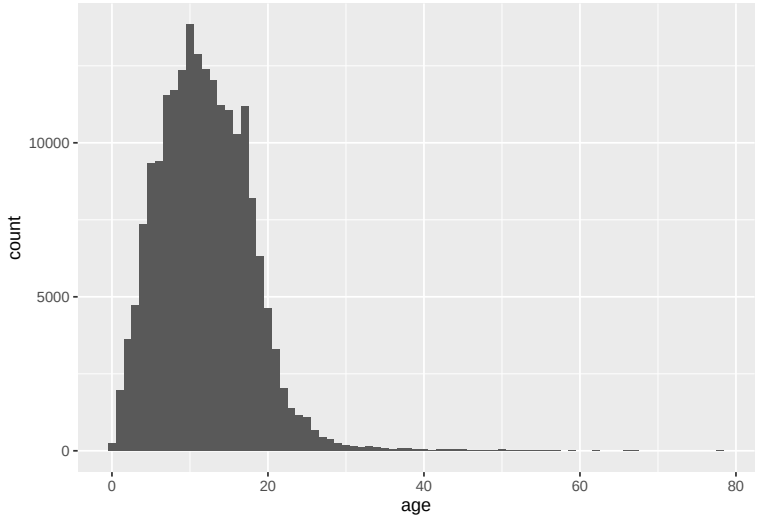
```
autos_clean %>% filter(brandModel %in% most_offer_models$brandModel & age <= 40) %>%
  ggplot(aes(age, price, color=brandModel)) + geom_smooth() + scale_y_log10()
```



All the brands have the same curve in the first 20 years (price decrease 10 times) and then begin to rice in the next 20 years some models more than others.

Another factor that can be useful is analyze the number of cars offered by age

```
autos_clean %>% group_by(age) %>% summarize(count=n()) %>%
  ggplot(aes(age, count)) + geom_col()
```

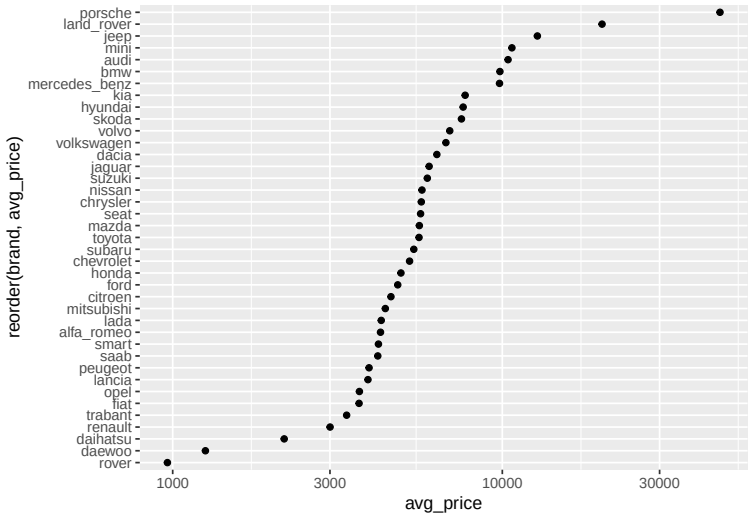


We can see that the mayor number of offered cars are near the 12 years of age. This data can be useful if we need to analyze a particular case and we don´t like the effect of the age. For avoid this age effect we need to pick cars near the 12 years of age.

Price vs brand

If we make a plot of average price per brand we look as we can imagine that are brands more expensive like others.

```
autos_clean %>% group_by(brand) %>% summarize(avg_price = mean(price)) %>%
  ggplot(aes(avg_price, reorder(brand, avg_price))) + geom_point() + scale_x_log10()
```

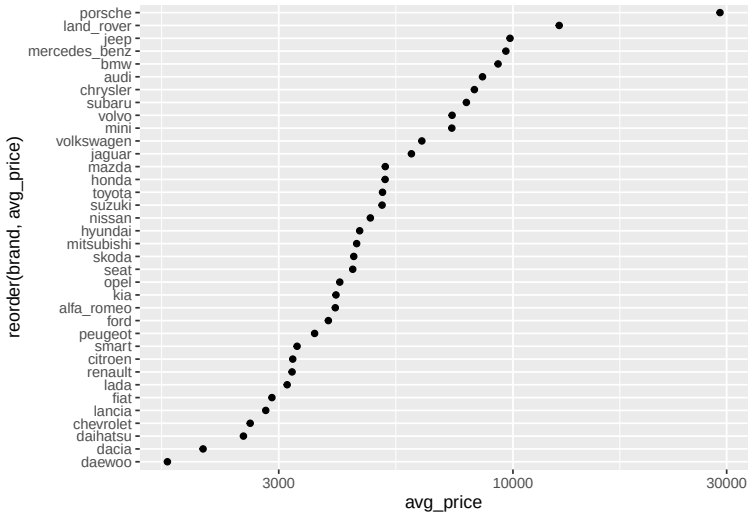


Is curious that brands like mini, kia and hyundai are in average more expensive than others brands like volvo, jaguar and chrysler.

We can think that mini, kia and hyundai are relative new brands in comparison with the others that be in the market more years (less price).

Now we take the cars near 10 years of age to eliminate the influence of the age.

```
autos_clean %>% group_by(brand) %>% filter(age>9 & age<12) %>%
  summarize(avg_price = mean(price)) %>%
  ggplot(aes(avg_price, reorder(brand, avg_price))) + geom_point() + scale_x_log10()
```

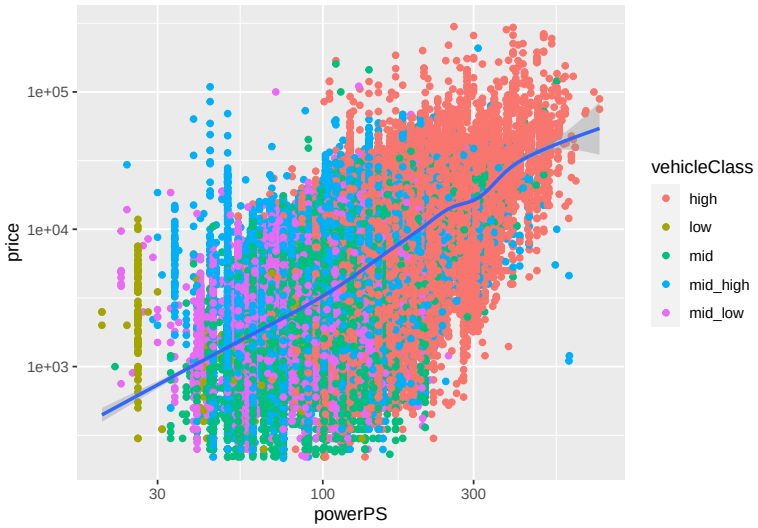


Now the graph shows the price of the cars as we expected, and also confirm the importance of the car age.

Price vs powerPS

Generally the price of a car is associated to the power of the engine. Small urban cars have engine less than 100HP and the more bigger is the car more power need the the engine to move it.

```
autos_clean %>% ggplot(aes(powerPS, price)) + geom_point(aes(color=vehicleClass)) +
  geom_smooth() + scale_x_log10() + scale_y_log10()
```



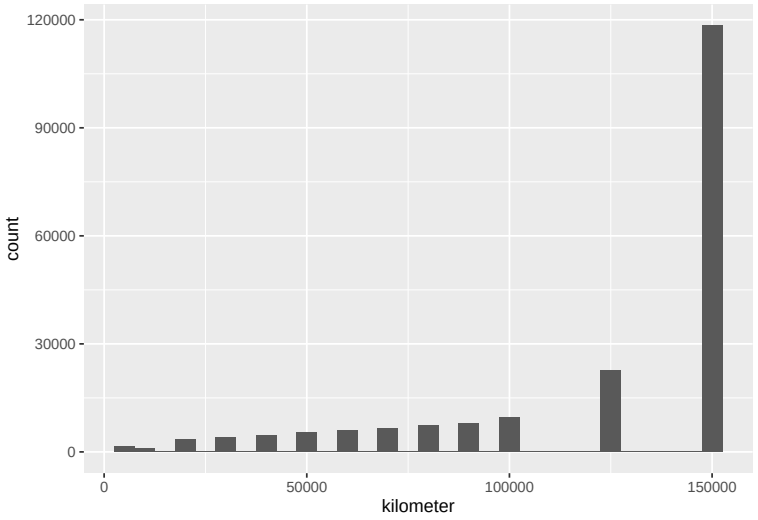
In the graph we can see a linear relation between the power and price of the car. Also we can note that the high is the class of a vehicle the the power and price is greater.

Price vs kilometer

Other important variable to be considered by a car buyer is the kilometer that have the car. To more kilometer car have, more wear is present and the price must be lower. Only in the market of the collection cars don´t matter the kilometer of the car, the buyer only see the condition of the car.

Let´s look the distribution of the kilometer variable.

```
autos_clean %>% ggplot(aes(kilometer)) + geom_histogram()
```



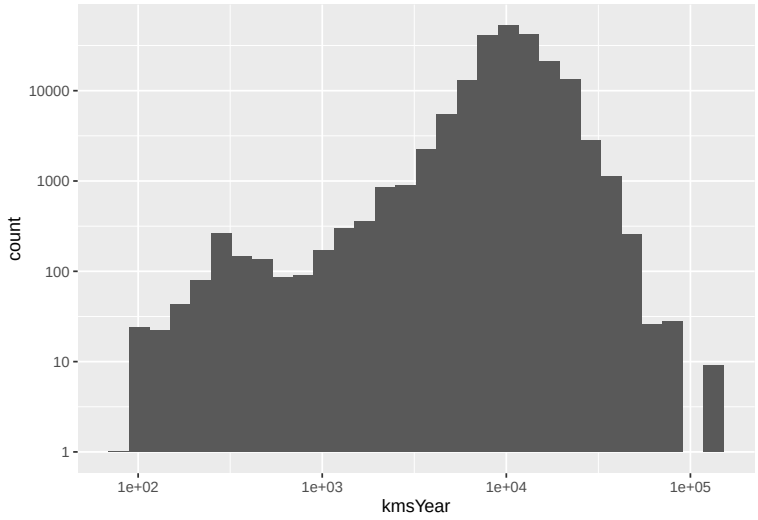
First thing that we note in this graph is that the kilometer variable is not continuous, we have categorical values of the kilometer like 5000 kms, 10000 kms or 150000 kms.

For sure the dominating kilometer is for cars with 150000 kms or more and the 60% of the market is dominated by the cars in this category. Considering this we can consider that the majority off users sells their cars after

make a lot of kilometers and the car begins to present some problems. This cars have a lower price and the number of buyers that can pay for them is greater.

Is really important analyze the kilometers per year. Let´s create a new variable kmsYear and plot it.

```
autos_clean %>% filter(age != 0) %>% mutate(kmsYear=round(kilometer/age)) %>%
  ggplot(aes(kmsYear)) + geom_histogram() + scale_y_log10() + scale_x_log10()
```



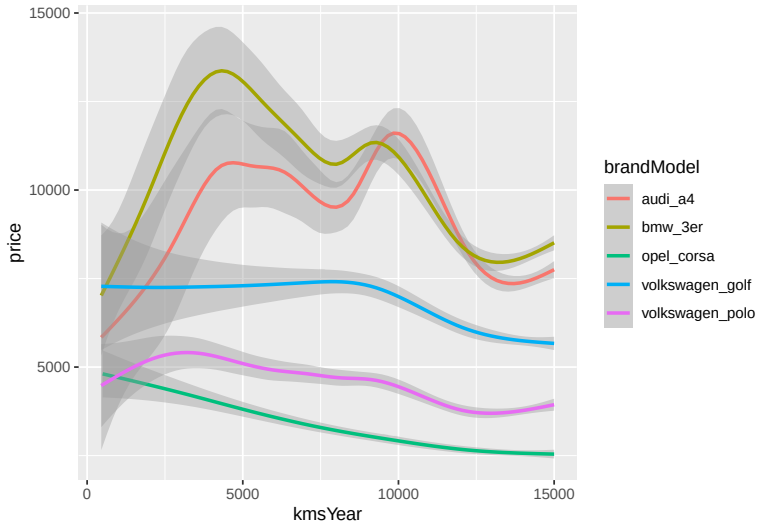
We can find that few people drive their cars only less than 1000 kms/year and other few drivers use their cars over 50000 kms/year.

The mayor part of drivers use their cars around 10000 kms/year. With this in mind for example a 10 years age car with 50000 kms is a car with low use, we expect that a car with this age have around 100000 kms.

Lets analyze whats going on with the group of most offered cars that we search previously.

When comparing the price with the kms/year of this 5 cars and pick only the ones around a age of 10 years (to lower the age effect) we have:

```
autos_clean %>% mutate(kmsYear=round(kilometer/age)) %>%
  filter(brandModel %in% most_offer_models$brandModel & age>9 & age<12) %>%
  ggplot(aes(kmsYear, price, color=brandModel)) + geom_smooth()
```



If we look at the audi_a4 and the bmw_3er we can see a great difference in prices when the kms/year change.

For kms/year lower than 8000, we note a significant increase in the prices and for higher than 8000 kms/year we have a decrease in the prices.

With the opel_corsa and volkswagen polo (that are the most cheaper from this group) we note the same curve but with little variations in the prices against the kms/year of the car.

The volkswagen_golf (the medium range price of the group) have a similar behavior of the bmw and audi but with lower differences in prices.

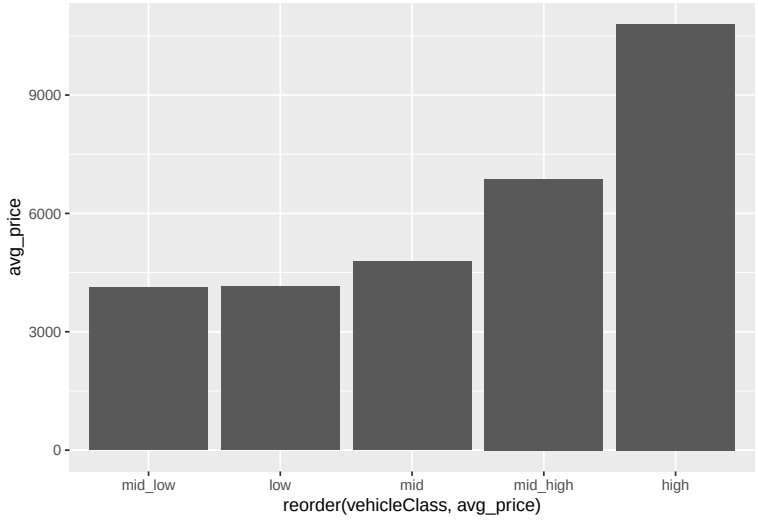
We can conclude that in the way the price of a car became higher the more higher is the influence of the car kilometers. A expensive car with many kms can be more expensive to repair and maintain and the risk of the buy increase. For this reason the higher is the price (risk) of the car the lower is the price. In the sector of the low class cars the prices are lower and repair a damage car is not a problem.

Price vs vehicleClass

As we saw in the last pic the price is influenced by the class of the car, we did this classification looking the level of prices and prestige of each brand.

Let´s plot the avg price of each class

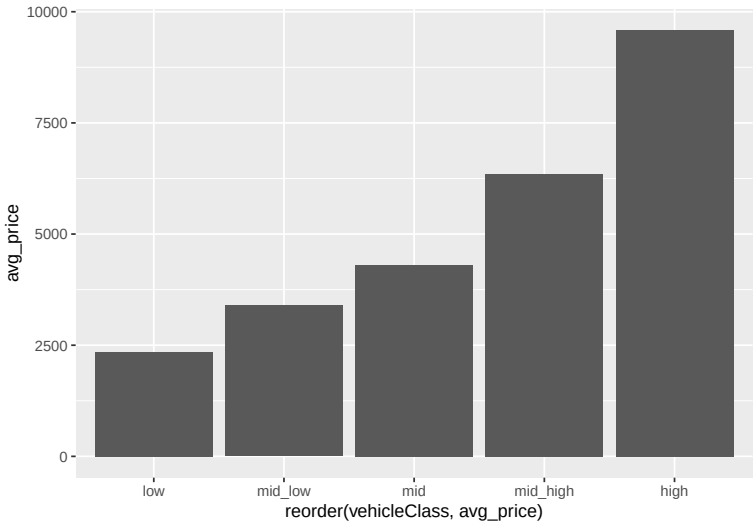
```
autos_clean %>% group_by(vehicleClass) %>%
  summarize(avg_price=mean(price)) %>%
  ggplot(aes(reorder(vehicleClass, avg_price), avg_price)) + geom_col()
```



As we can see, there is not a marked difference as we expected. This is because the effect of the age is present. Perhaps are more older or newer cars in one class than others.

If we reduce the effect of the age taking only the cars around the 10 years

```
autos_clean %>% filter(age>9 & age<12) %>% group_by(vehicleClass) %>%
  summarize(avg_price=mean(price)) %>%
  ggplot(aes(reorder(vehicleClass, avg_price), avg_price)) + geom_col()
```

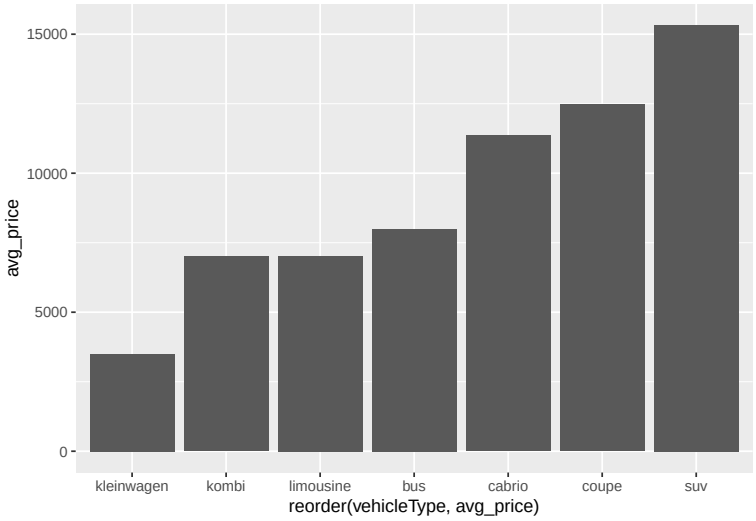


Now is clear the differences class by class, and we can see the strong relation between the class and the prices.

Price vs vehicleType

If we take the average price per type of vehicle and plot we obtain this

```
autos_clean %>% group_by(vehicleType) %>%
  summarize(avg_price=mean(price)) %>%
  ggplot(aes(reorder(vehicleType, avg_price), avg_price)) + geom_col()
```

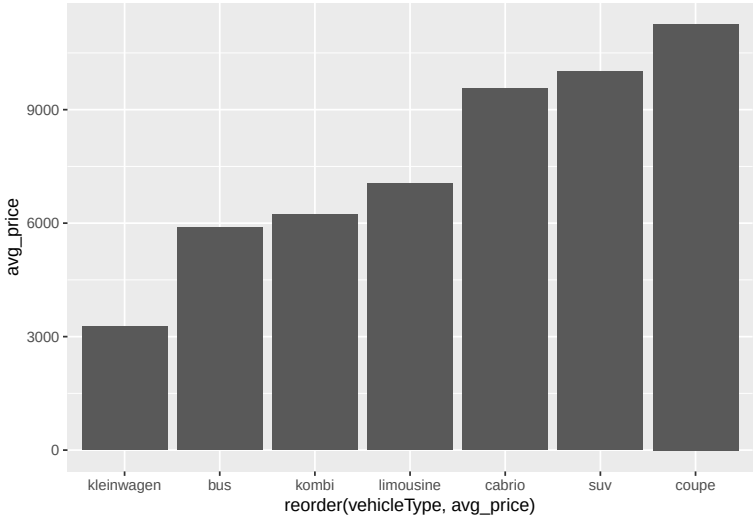


There is a logic here, the kleiwagen (small cars) is the cheaper and the cabrio, coupe and suv are the more expensive types.

But we have been expecting that the coupes (sport cars) were more expensive than the suvs. The suv market is relatively a new segment in the car market and for sure there are more newest models (more price) offering on eBay than the coupe type (have been always in the car industry).

Let´s reduce the age effect to corroborate this

```
autos_clean %>% filter(age>9 & age<12) %>% group_by(vehicleType) %>%
  summarize(avg_price=mean(price)) %>%
  ggplot(aes(reorder(vehicleType, avg_price), avg_price)) + geom_col()
```

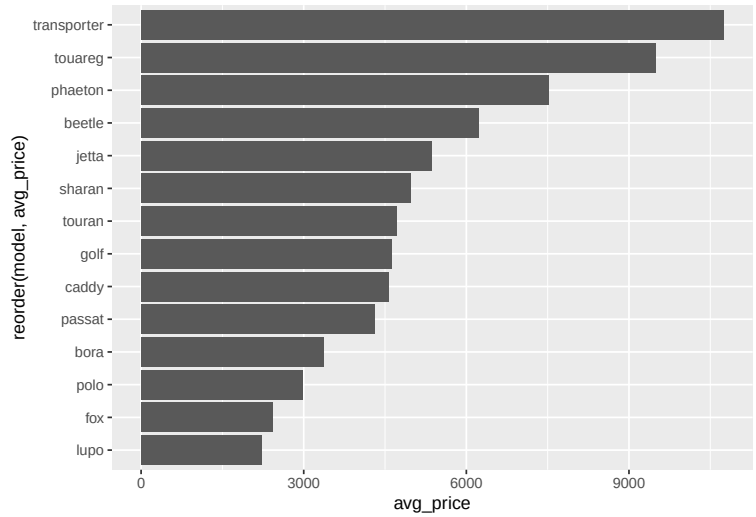


Now we can see that the sports cars (more powerful) are the type with higher prices. And also is clear that the suv market is a very attractive segment for all brands, inclusive brands like Lamborghini today are selling powerful suvs.

Price vs model

To analyze the price of a car against it´s model we will analyze a specific brand. Let´s choose volkswagen, the most offered brand on eBay, and take cars around 10 years old to reduce the effect off the age.

```
autos_clean %>% filter(brand=="volkswagen" & age>10 & age<14) %>% group_by(model) %>%
  summarize(avg_price=mean(price)) %>%
  ggplot(aes(avg_price, reorder(model, avg_price))) + geom_col()
```



It´s obvious that are models more expensive like others. Generally all brands try to cover all the spectrum of the market. There are brands like Porsche or Ferrari that only have super cars, but also they have great difference in prices between the models that offer.

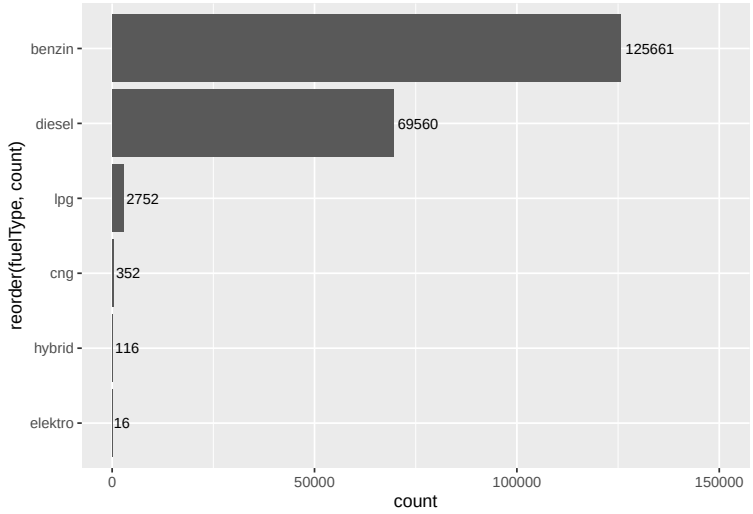
Price vs fuelType

CNG (compressed natural gas) is the cheapest fuel in the market and the 3rd more used fuel. There are many models generally in the small cars market that came with this fuel from fabric. Also any car with gasoline or diesel can be adapted to work with cng.

The LPG is a gaseous fuel like the cng but have the twice of heat power. Is more used in heavy vehicles, generally to work purposes or transport.

Let´s plot the distribution of the market by type of fuel

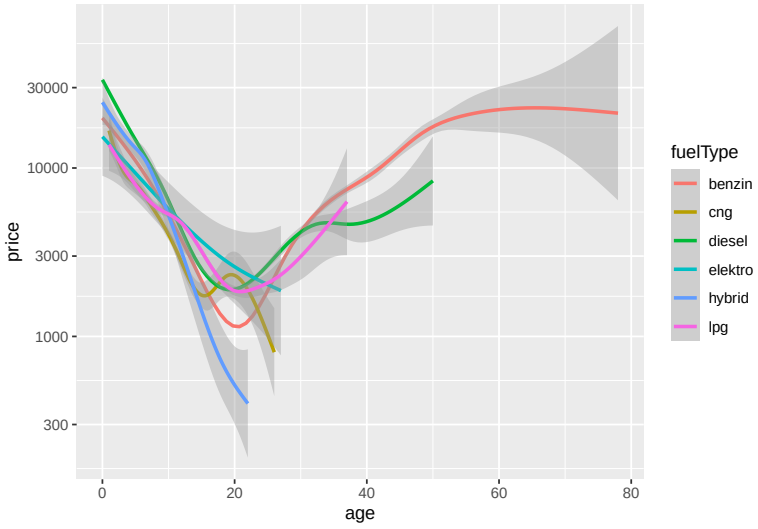
```
autos_clean %>% group_by(fuelType) %>% summarize(count=n()) %>%
  ggplot(aes(reorder(fuelType, count), count)) +
  geom_col() + coord_flip(y=c(0,150000)) +
  geom_text(aes(label= count), hjust=-0.1, size=3)
```



The used car market is clearly dominated by the benzin and diesel cars, only 16 electric and 116 hybrid cars are offered on eBay. Less than the 2% of the used car don´t have benzin or diesel engines.

As we can see in the next graph, the market of electric and hybrid cars is new (20 years) and there are few offered.

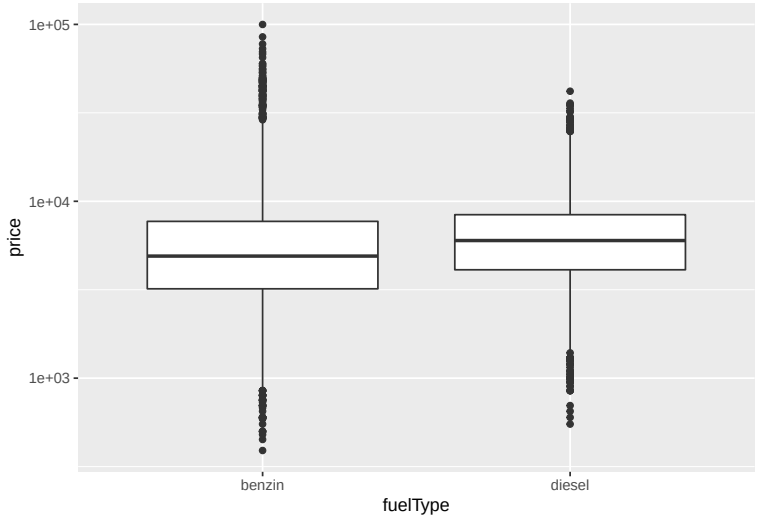
```
autos_clean %>% ggplot(aes(age, price, color=fuelType)) + geom_smooth() + scale_y_log10()
```



Also we can see that the electric car is the type of car with the lower depreciation in time. This explain the reason that is rare view a electric car posted on eBay.

Let´s plot fuelType vs prices considering only benzin and diesel cars reducing the age effect.

```
autos_clean %>% filter(fuelType %in% c("benzin", "diesel") & age>9 & age<12) %>%
  ggplot(aes(fuelType, price)) + geom_boxplot() + scale_y_log10()
```



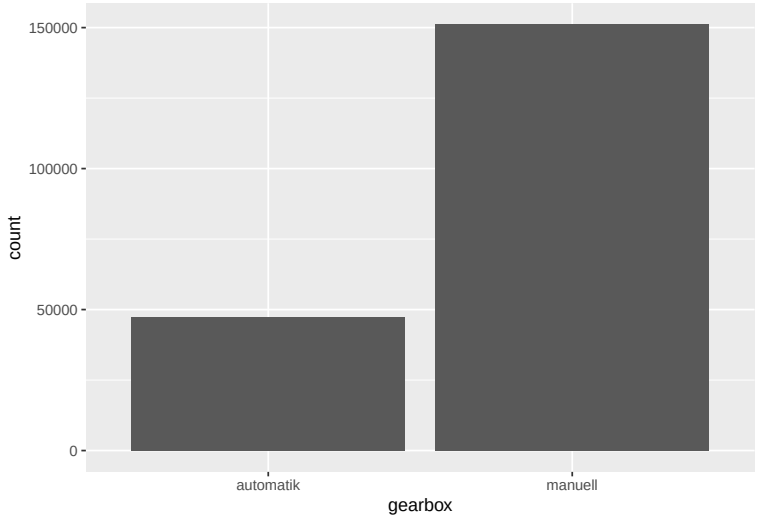
Diesel cars are more expensive than gasoline (benzin) cars. Diesel engines have a longer useful life than gasoline engines, and also have less fuel consumption valuing this cars.

Price vs gearbox

Let´s plot the distribution of both types of gearbox, manual and automatic

```
autos_clean %>% ggplot(aes(gearbox)) + geom_histogram(stat="count")
```

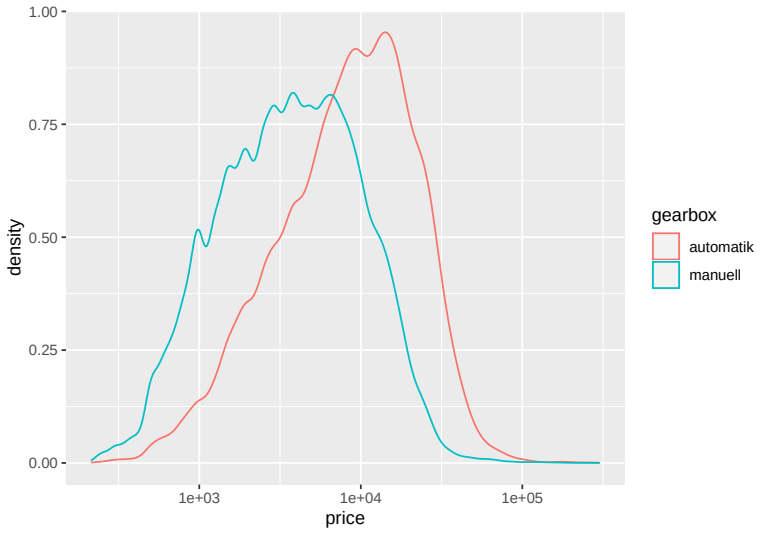
```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```



The market of the manual used cars is 3 times bigger than the automatic cars

Now let´s see what occur when comparing the gearbox type vs the prices

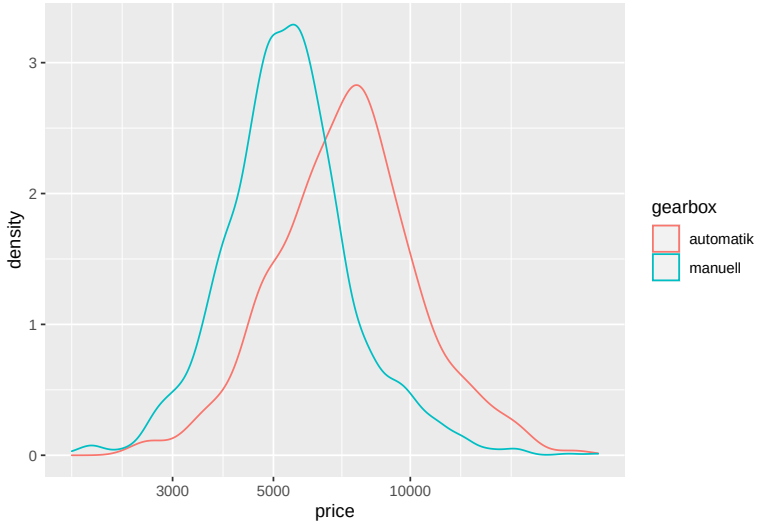
```
autos_clean %>% ggplot(aes(price, color=gearbox)) + geom_density() + scale_x_log10()
```



The automatic cars have a higher price than the manual cars as we expected.

Let´s look if this is true inspecting the gearbox difference in prices for a unique model like the volkswagen golf(more offered car) with a age around 10 years (to reduce the age effect).

```
autos_clean %>% filter(brandModel=="volkswagen_golf" & age>9 & age<12) %>%
  ggplot(aes(price, color=gearbox)) + geom_density() + scale_x_log10()
```

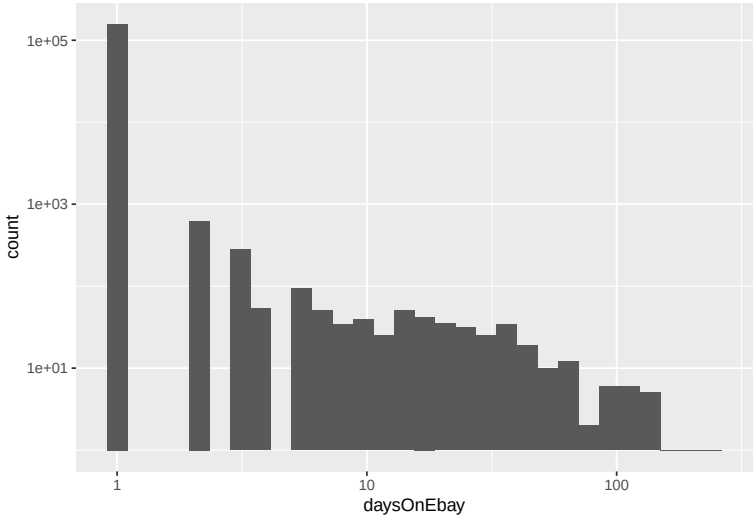


Inspecting this unique model (golf) we can find the same distribution as we found in general, more manual cars and the automatics more expensive.

Price vs daysOnEbay

First we will analyze the distribution of this variable

```
autos_clean %>% ggplot(aes(daysOnEbay)) + geom_histogram() +
                                          scale_y_log10() + scale_x_log10()
```



The 50% of the cars on eBay have been offered only 1 day and a few ones more than 10 days. The oldest adds have been posted on eBay 200 days.

Let´s plot the price of the car vs the days offered on eBay.

```
{rmessage=FALSE, warning=FALSE, out.width="60%", fig.align="center"} autos_clean %>%
ggplot(aes(daysOnEbay, price)) + geom_smooth() +   geom_point(alpha=0.05) + scale_y_log10()
+ scale_x_log10()
```

We can see that the cars posted more days on eBay generally are cars with higher prices. But this relation is not really significant to consider.

Correlation

In this section we will analyze the correlation between the numeric variables like the price, age, powerPS and kilometer.

The correlation of two variables is a metric of how good is the relation of them. We have a perfect relation when the result is 1 and no relation if we have a 0. Also the correlation is positive when the relation is direct and negative when indirect.

Let´s create a graph with the correlation of the variables

```
autos_correlation <- autos_clean %>% select(price, kilometer, powerPS, age)
corrplot.mixed(cor(autos_correlation))
```

Considering the price, our principal variable, we see that the variable with more correlation (0.58) is the powerPS of the car, the second is the age (-0.47) and the third is the kilometer of the car (-0.44).

This three variables have almost the same relation and depends on the buyer choose one of the 3 to be the principal criterion to buy a car.

Other relation that we can see is the relation between the age of the cars offered and the power. The correlation is not great (-0.22) but is significant. We can conclude that the older cars have more power, or that the today market is dominated by urban cars with less power.

Predict model

Creating the train and test dataset

With all the analysis job we conclude that the variables that affect the price are: brandModel, age, powerPS, kilometer, price, vehicleType, vehicleClass, gearbox and fuelType.

Let´s remove all others variables

```
autos_pred <- autos_clean %>% select(price, powerPS, kilometer, age, brandModel,
                                     vehicleType, vehicleClass, fuelType, gearbox)
```

The train dataset will have the 80% of the data and the test dataset the other 20%.

```
set.seed(1)
test_index <- createDataPartition(y = autos_pred$price, times = 1, p = 0.2, list = FALSE)
autos_train <- autos_pred[-test_index[,1],]
temp <- autos_pred[test_index[,1],]

# Make sure brandModel in test set are also in train set
autos_test <- temp %>% semi_join(autos_train, by = "brandModel")

# Add rows removed from test set back into train set
removed <- anti_join(temp, autos_test)
```

```
## Joining, by = c("price", "powerPS", "kilometer", "age", "brandModel", "vehicleType", "vehicleClass",
```

```
autos_train <- rbind(autos_train, removed)

#Remove unused
rm(test_index, temp, removed)
```

Linear regression modeling

The chosen model is the linear regression with a 5 cross fold validation.

```
control <- trainControl(method = "cv", number = 5, p = 0.8)
fit_lm <- train(price ~ ., method = "lm", data = autos_train,
                trControl=control)
```

On the test dataset we predict the prices and store them in pred_price variable

```
autos_test$pred_price <- predict(fit_lm, autos_test)
```

Result

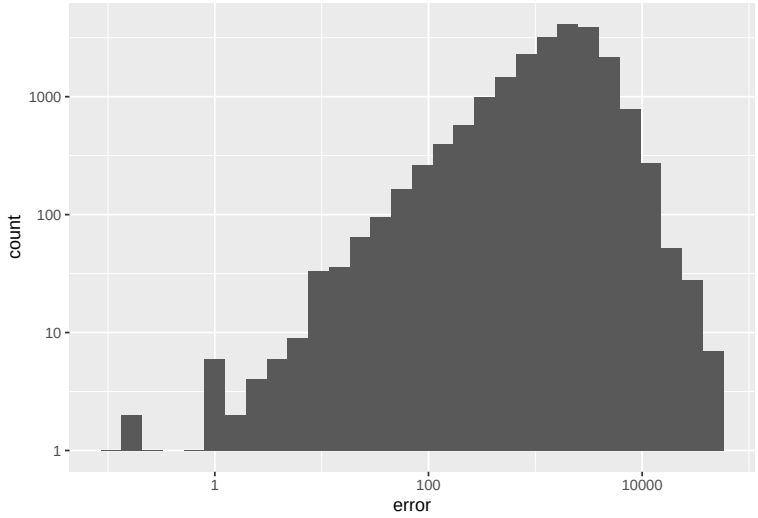
The total error of the prediction is 4785€

```
fit_lm$results$RMSE
```

```
## [1] 4785.426
```

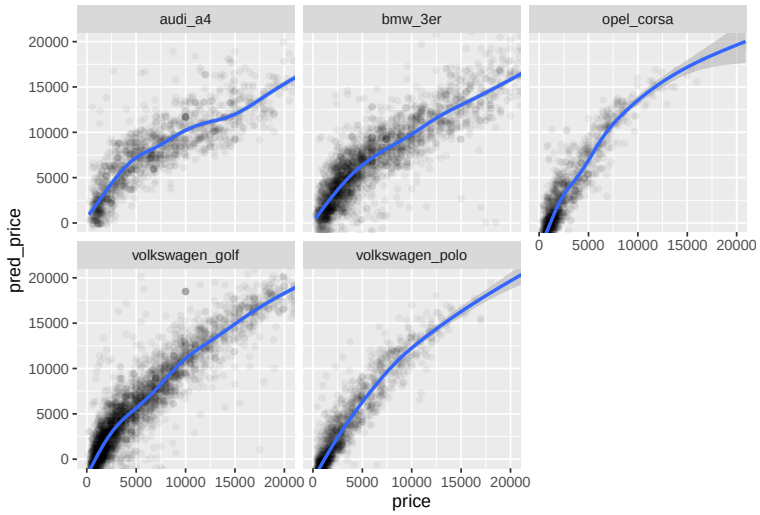
This is the distribution of the error

```
autos_test %>% mutate(error=pred_price-price) %>%
  ggplot(aes(error)) + geom_histogram() + scale_y_log10() + scale_x_log10()
```



Let´s see the actual prices vs the predictions in the 5 most offered models on eBay

```
autos_test %>% filter(brandModel %in% most_offer_models$brandModel) %>%
  ggplot(aes(price, pred_price)) + geom_point(alpha=0.05) + geom_smooth()+
  facet_wrap(~brandModel) + coord_cartesian(xlim=c(0,20000), ylim=c(0, 20000))
```



We can see that the predicted price is higher than the actual price for cars with prices lower than 10000€.But for cars with prices higher than 10000€ the predicted price tend to be lower than the actual.

Now look this 5 five cars but let´s take the ones around 10 years age to reduce the age effect

```
autos_test %>% filter(brandModel %in% most_offer_models$brandModel & age>9 & age<12) %>%
  group_by(brandModel) %>%
  summarize(avg_price=round(mean(price)),
```

```
            avg_predicted=round(mean(pred_price)),
            avg_error = round(sqrt(mean((pred_price-price)^2)))) %>%
  mutate(percentage_error=round(avg_error/avg_price*100)) %>% kable()
```

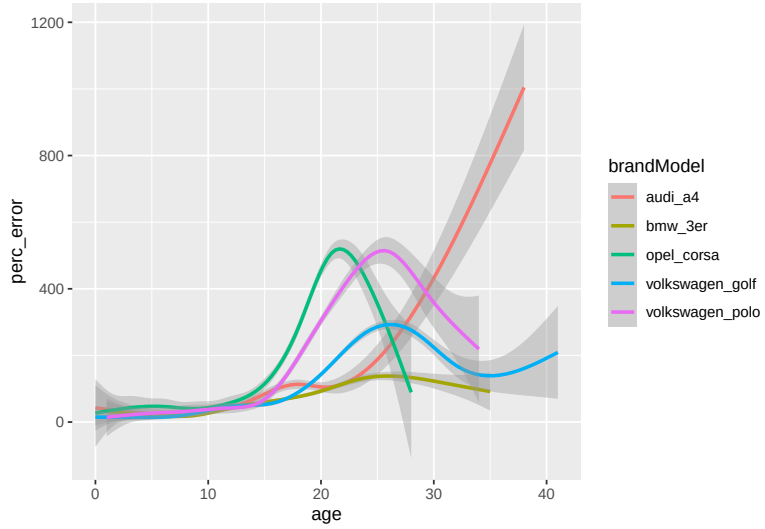
brandModel	avg_price	avg_predicted	avg_error	percentage_error
audi_a4	7871	9136	2585	33
bmw_3er	8814	9250	2235	25
opel_corsa	2882	3777	1937	67
volkswagen_golf	5963	7208	2272	38
volkswagen_polo	4154	5170	2214	53

The error in the opel_corsa is 67% of the actual price. In the bmw_3er the error is 25% of the price. In average the predicted prices in this models are higher than the average actual prices.

Because is not the same an error of 2000€ in a 3000€ car than a 2000€ error in a 12000€ we will calculate the percentage error from the actual price.

Let´s plot the percentage error or this 5 models against the age.

```
autos_test %>% filter(brandModel %in% most_offer_models$brandModel & age<50) %>%
  mutate(perc_error=(abs(pred_price-price))/price*100)  %>%
  ggplot(aes(age, perc_error, color=brandModel)) + geom_smooth()
```



We can see in the graph that for cars until the 15 years of age the percentage error is increasing but still admissible.

But for cars with more than a age of 15 years the error is very high, near 700% in the the opel and polo.

This happen because we can find in the dataset a lot of cars near the 200€ price with a age near 20 years. The range of prices in a car with more than 20 years is great. We can find the same car in 200€ or 2000€, all depends of the condition of the car.

To obtain a more accurate predicting model we need to limit the age range to 0 - 20 years.

Linear regression in reduced dataset(age $<$ 20 and price $<$ 20000)

As we explained in the previous model with the complete dataset the prediction generated for cars older than 20 years is have a large error because the principal variable became the condition of the car and we don´t

have that data.

There are few cars in the dataset with elevated prices, and this high prices causes high errors too.

Reduced dataset

Let´s reduce the clean dataset to the cars with ages of 20 years or lower and a price lower than 20000€

```
red_autos_pred <- autos_clean %>%
  filter(age<=20 & price<=20000) %>% select(price, powerPS, kilometer, age, brandModel,
                                            vehicleType, vehicleClass, fuelType, gearbox)
```

```
data.frame(autos_clean = nrow(autos_clean), reduced_dataset=nrow(red_autos_pred),
           percentage_reduced=round((1-nrow(red_autos_pred)/nrow(autos_clean))*100)) %>%          kable()
```

autos_clean	reduced_dataset	percentage_reduced
198457	174134	12

This change represent only a 12% reduction of our clean dataset, the reduced dataset now count 174134 cars.

Reducing the train and test datasets

```
red_autos_train <- autos_train %>% filter(age<=20 & price<=20000)
red_autos_test <- autos_test %>% filter(age<=20 & price<=20000)
```

Linear regression modeling

Let´s use the same linear regression model than the complete dataset

```
control <- trainControl(method = "cv", number = 5, p = 0.8)
red_fit_lm <- train(price ~ ., method = "lm", data = red_autos_train,
                    trControl=control)
```

On the test dataset we predict the prices and store them in pred_price

```
red_autos_test$pred_price <- predict(red_fit_lm, red_autos_test)
```

Result

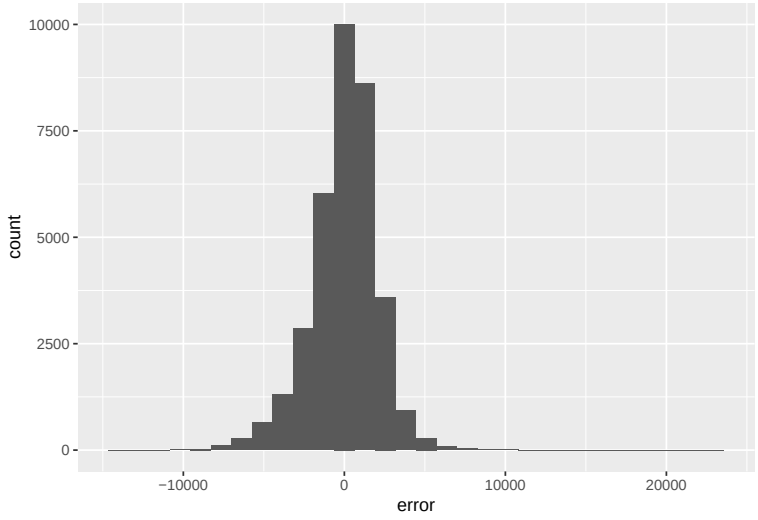
The total error of the prediction on the reduced dataset is 2068€, 57% lower than the complete dataset (4785€)

```
red_fit_lm$results$RMSE
```

```
## [1] 2068.329
```

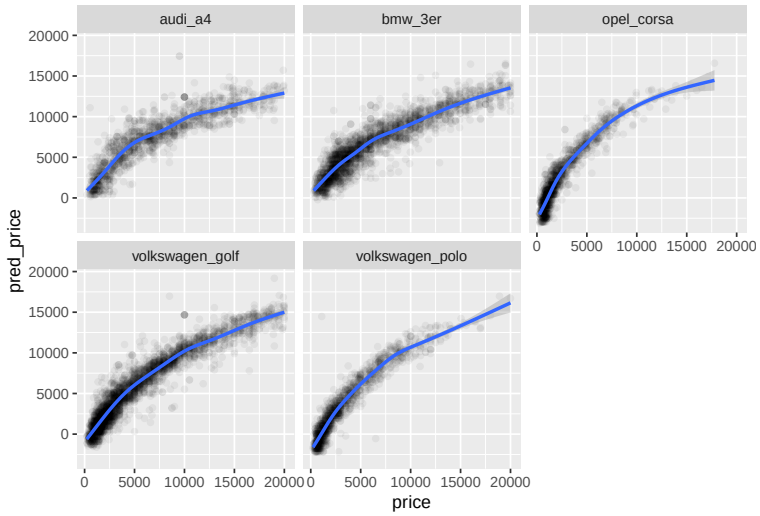
This is the distribution of the reduced error

```
red_autos_test %>% mutate(error=pred_price-price) %>%
  ggplot(aes(error)) + geom_histogram()
```



Let´s see what happen in the 5 most offered models on eBay

```
red_autos_test %>% filter(brandModel %in% most_offer_models$brandModel) %>%
  ggplot(aes(price, pred_price)) + geom_point(alpha=0.05) + geom_smooth()+
  facet_wrap(~brandModel)
```



We can see that the predicted price is higher than the actual price for cars with prices lower than 10000€. But for cars with prices higher than 10000€ the predicted price tend to be lower than the actual.

The dispersion is lower than the complete model thanks to the lowers errors.

Now look the five most offered cars but let´s take the ones with a age of 10 years to eliminate the age effect

```
red_autos_test %>% filter(brandModel %in% most_offer_models$brandModel &
                          age>9 & age<12) %>%
  group_by(brandModel) %>%
  summarize(avg_price=round(mean(price)),
            avg_predicted=round(mean(pred_price)),
            avg_error = round(sqrt(mean((pred_price-price)^2)))) %>%
  mutate(percentage_error=round(avg_error/avg_price*100)) %>% kable()
```

brandModel	avg_price	avg_predicted	avg_error	percentage_error
audi_a4	7871	8499	2009	26
bmw_3er	8771	8492	2034	23
opel_corsa	2882	4190	1480	51
volkswagen_golf	5916	6845	1754	30
volkswagen_polo	4154	5158	1321	32

The errors now are in a range of 22% - 52% of the actual prices. In the complete dataset the range were 25% - 67% in the same cars.

Finally we can inspect 10 random cars to look how close are the predictions

```
set.seed(103)
ind <- sample(c(1:nrow(red_autos_test)), 10)
red_autos_test[ind,] %>% mutate(predicted = round(pred_price)) %>%
  select(brandModel, price, predicted, age, kilometer) %>% kable()
```

brandModel	price	predicted	age	kilometer
bmw_5er	2990	3109	19	150000
bmw_3er	1300	3039	19	150000
hyundai_tucson	6200	6676	7	150000
seat_ibiza	2295	3730	11	150000
opel_astra	7990	7743	10	20000
audi_a3	1200	1033	19	150000
mazda_mx_reihe	3200	2553	17	150000
volkswagen_passat	15850	12794	3	90000
bmw_7er	6500	6305	18	150000
opel_corsa	2500	3422	11	150000

IT´S NOT TOO BAD!!!

Conclusion

The used car market is more complex than the 0 km market because are playing new variables like the age of the car and the kilometers and other factor very important is the condition of the car. This variable turn more important as the age increase and because we don´t have this variable the prediction of the price turn more difficult in cars with more than 15 years of age.

The first and big task were clean the downloaded dataset that is in a raw form and count with many errors introduced by the sellers on eBay. The dataset involve all cars offered in eBay, many of them in not working condition, some prices are the quotes of leasing or financed cars and even there are parts like tires, accessories like navigation systems and model cars for hobbies. If we want a good work for analysis and prediction is necessary a relatively clean dataset to work on it.

Once the dataset was cleaned we have analyze the data and found things that is obvious like the price of the car decrease in the time, but also we found interesting things like:

- The price decrease in average 10 times in 20 years and then in the next 20 years a few ones can recuperate the original price or more.
- The used cars market is dominated by cars near the 10 years old.

- The average kilometers per year done by a used car is around the 10000 kms/year. This can be a good hint to know how much a car is used.

- The price of a low class car don´t depend to much of the kilometer. We think that this is because the risk of buy a not good condition car is low (cheaper spare parts). But the effect of the kilometer in a high class car is great, high class cars with around 5000 kms/year can increase their value even twice than others with 10000 kms/year or more.

- The electric cars are the cars with lower depreciate in time. Looses their value a half that a gasoline car and for sure that is the reason that we can only a few posted on eBay.

- The hybrid cars are the ones with a higher depreciate rate.

The third task were model an algorithm to predict the price of a used car using our cleaned dataset. We choose the linear regression and the average error (RMSE) of the prediction was 4700€. This error is only 25% of the value in a high class car, but represent the 55% of the value in a low class car. This great error is caused by the great variations of price in cars with more than 15 years old.

A last task have been reduce our clean dataset eliminating the cars with a age greater than 20 years and a price over the 20000€. This reduction only represented a 12% of the data but the error of the prediction been reduced 60% and now is only 2000$. This new prediction over the reduced cleaned dataset is very low as we can saw inspecting random samples.

A future work can consist in implement others algorithms to reduce further more the error and keep collecting data from others markets like the north american to analyze and compare the behavior of different markets.