## horizontal line



Big Data Investment

Used Car Report

12/23/2020

**─**

Filipe Soares

CSDA 1020

**Overview**

Big Data Investment is looking to expand into the used car business. Customers normally turn to used cars in an attempt to get something that is reliable yet affordable to avoid the large prices found with new cars at the dealership. It can be a challenge to find what cars the public desires the most when the pool to choose from is so vast with such variety in styles and prices. Throughout the report there will be a series of analyses performed on a dataset of used vehicles and recommendations on which vehicles should be purchased will be made at the end.

**Goal**

To clean and analyze the data to provide insight that will determine the best cars to purchase based on the following questions:

1. Which makers are most common and what conclusions can be made by analyzing their average mileage and price? What years manufactured are the most popular in the dataset?
2. Is there any benefit in buying cars with more engine power?
3. What are the most common models on sale and are there any correlations between them?

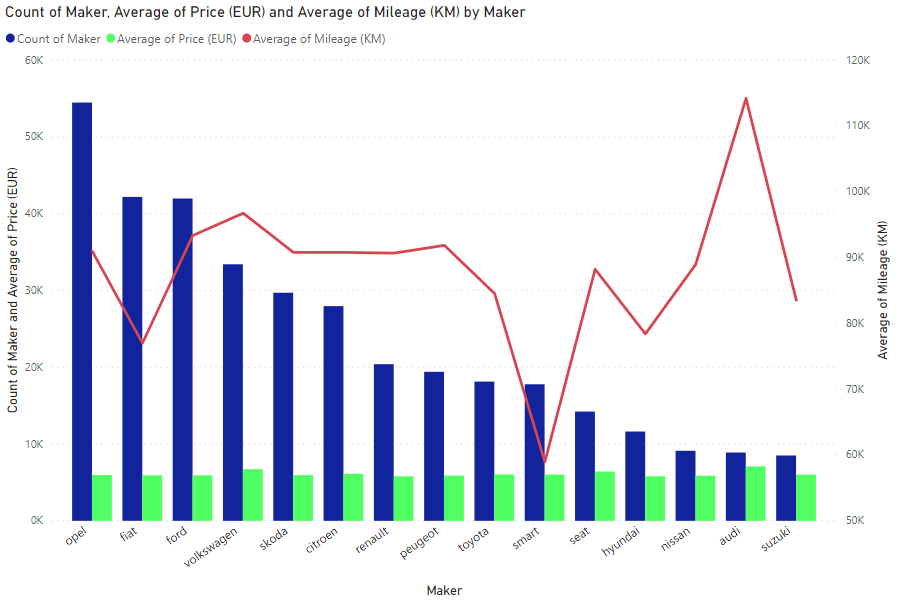
**Analysis**

Data Cleaning

This section will provide a brief summary of how the data was cleaned and prepared for this report. The full details and screenshots can be found in the appendix. The initial dataset contained 3,552,912 vehicles. Using Apache Spark, a systematic approach of creating new dataframes to filter the data led to the final dataset containing 413,924 cars. Unwanted factors were removed and null values were removed for maker, model, mileage and price. There was intent to do analysis based on the more recently viewed ads, however an unexpected discovery about the vehicles seen in the year 2017 led to a change in direction. All of those vehicles had the same price of EUR 1,295.34. The final dataset contained only vehicles listed with a price equal to or greater than EUR 1,500 and less than or equal to EUR 10,000, mileage between 10,000 to 150,000 km, and were manufactured during or after the year 2000. The final table also had the column titled “date\_last\_seen” dropped entirely because it was unlikely to be of value considering all the vehicles remaining in the table had not been viewed recently.

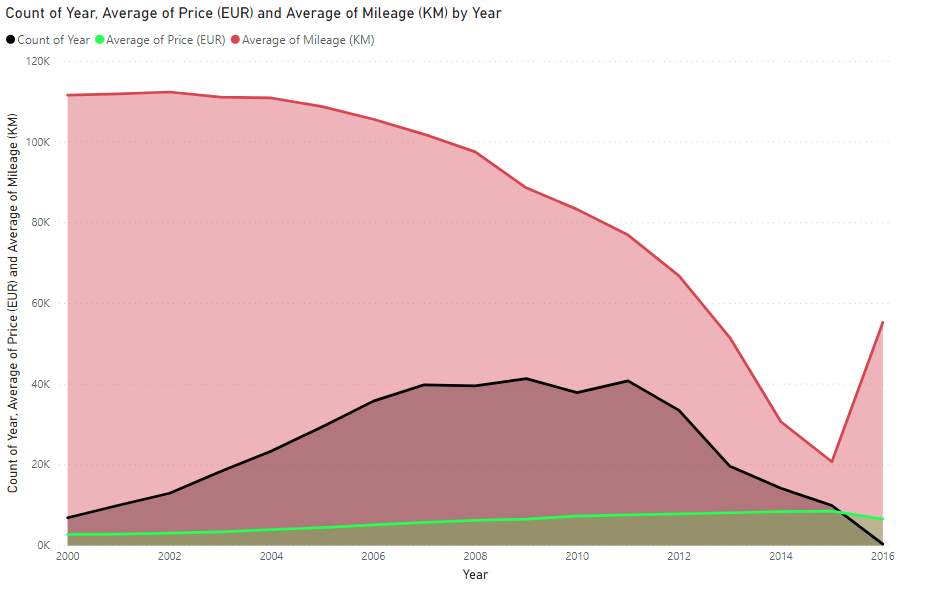
Common Makers and Popular Years

The primary goal is to identify which car makers are the most popular and what insights can be derived from their average mileage and average price. A bar and line graph containing the 15 most common car makers and the average price and mileage for each company was created to solve this problem. The top 15 makers are Opel, Fiat, Ford, Volkswagen, Skoda, Citroen, Renault, Peugeot, Toyota, Smart, Seat, Hyundai, Nissan, Audi and Suzuki. Figure 1 reveals that most of the top 15 makers have an average mileage between 80,000 and 100,000 kilometres with the exception of Fiat, Smart and Hyundai. The top 15 makers also have an average price between EUR 5,800 and EUR 6,200. The only makers to fall out of this price range are Seat, Volkswagen, Audi (all above the range) and Renault (under EUR 5,800). One standout in terms of maintaining value while having a high amount of average kilometres driven is Audi, with an average price of over EUR 7,000 and an average just above 114,000 kilometres. Volkswagen also maintained a higher price and had the 2nd highest average kilometres driven, indicating it may be wise to invest in this brand. Overall the top 15 vehicles are very similar in terms of mileage and price, however the lower kilometres driven for Smart cars while having the same average price as the others indicates that drivers are quicker to sell this brand. This likely means customers are less satisfied with Smart cars and they should be avoided.



**Figure 1**

Another dimension of the analysis was to group the vehicles by year and identify which are the most popular. This will provide an ideal vehicle age range, allowing the company to avoid buying cars that are “too old” or new cars that are on sale likely due to a lack of satisfaction. It was determined to only include cars from the year 2000 and later as there will likely be little value in cars made before then.

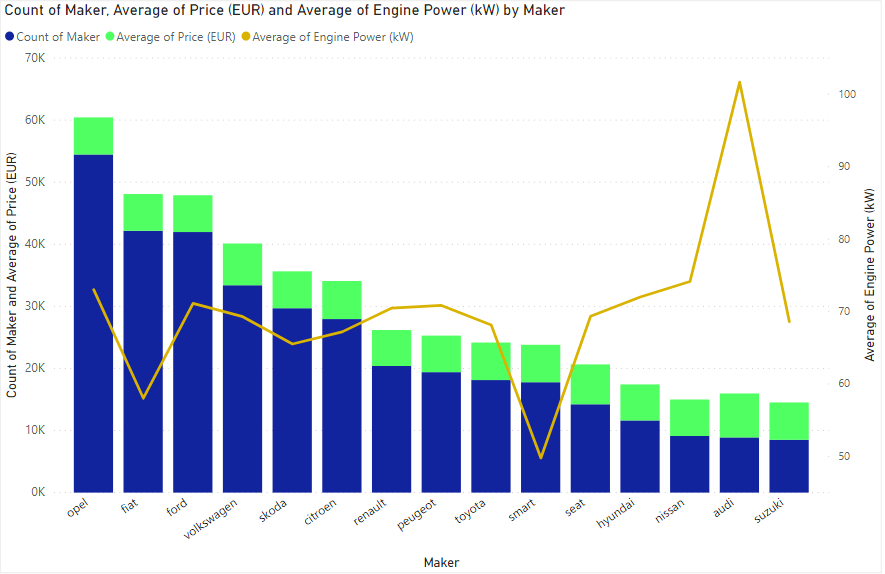


**Figure 2**

When looking at the average price, there is a steady increase in price beginning in 2004 until 2015. Prior to 2004, the average price was fairly stable, as well as the average mileage. After 2004 the average mileage begins to decrease slowly until 2008, where the decline in mileage becomes more pronounced until 2015. The count of vehicles is at its highest between 2006 and 2011. It is expected to have an increase in price and decrease of mileage as the year manufactured gets closer to present day. 235,284 cars were made between 2006 and 2011, representing more than half of the data. The range for average mileage is between 77,000 and 106,000 km, which is similar to the range of kilometres driven for the most common models. The price range for these years is between EUR 5,100 and EUR 7,600 which is almost 4 times wider than the range for the common models. When looking at vehicles made between 2006-2009 the price and mileage ranges are more similar to the popular brands at 88,000-106,000km and EUR 5,100 - EUR 6,500. There are many vehicles for sale in the year 2011 but the average price of just over EUR 7,600 will likely be seen as too expensive compared to the rest of the market.

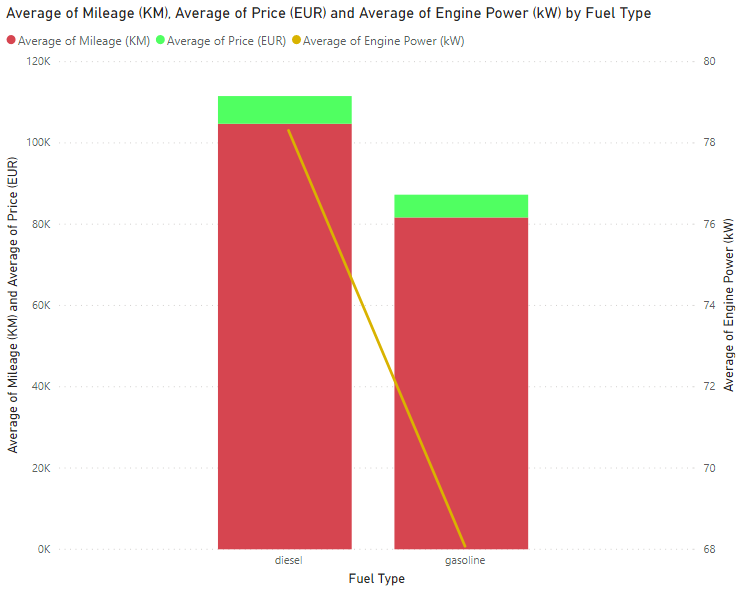
Engine Power

The next factor for analysis was engine power. This attribute is often mentioned in advertisements and in car dealerships as a selling point, but its value in the used car market was investigated in this report. Firstly, it was measured alongside average price and count of cars by maker. Immediately it was seen that the shape of the line for engine power is extremely similar to that of average mileage in Figure 1.



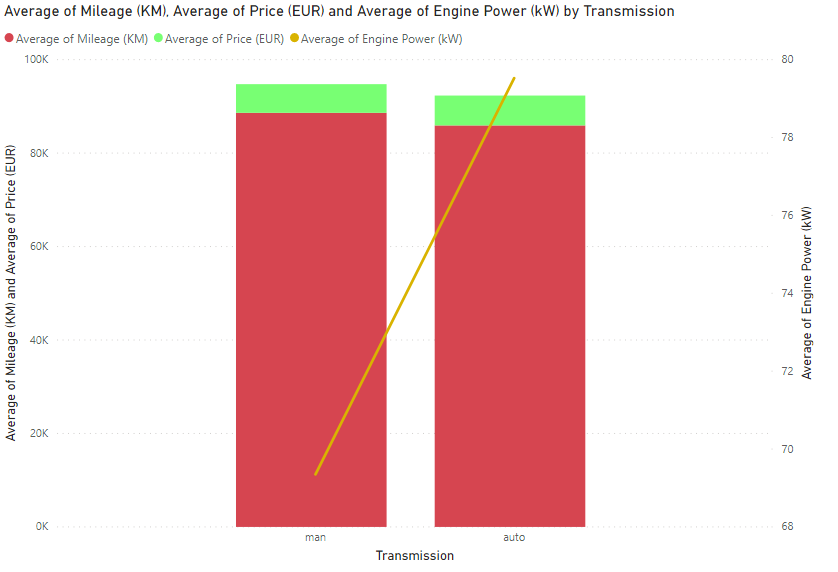
**Figure 3**

The majority of the vehicles had an average engine power between 67-73 kW, with only 3 exceptions in the Fiat, Smart (below 67) and Audi (above 73) brands. This indicates that engine power on the surface, engine power is not an important factor. Figure 4 reveals the relationship between fuel type, average price, average mileage and average engine power.



**Figure 4**

Diesel powered cars generally have more power (about 78kW to 68kW) and are more expensive on average with a price of EUR 6,818 compared to EUR 5,604 for gasoline powered cars. These vehicles also have a much higher average mileage which may indicate stronger engines last longer and maintain their value better than weaker ones. To investigate further, the relationship between engine power, transmission, average price and mileage was observed in Figure 5.

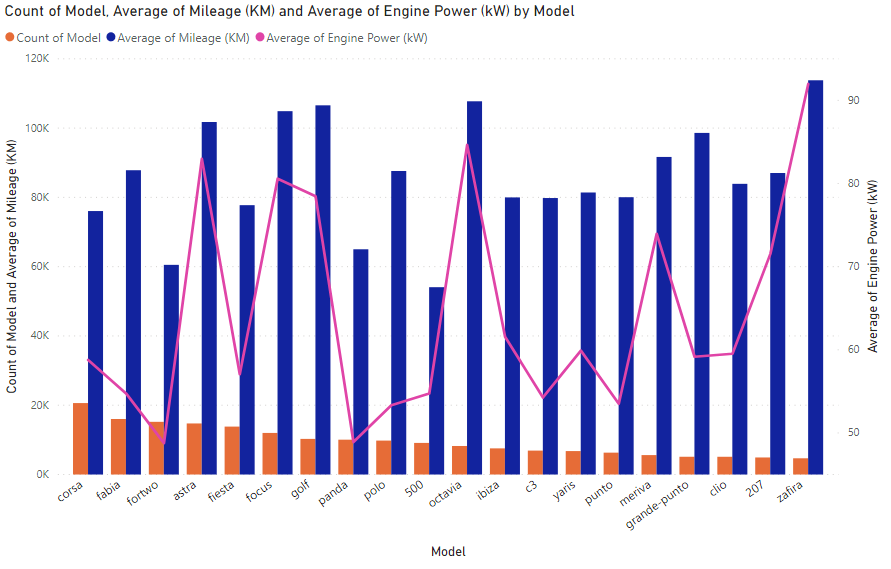


**Figure 5**

Once again the higher powered engines are more expensive, however the difference between manual and automatic transmission (EUR 6,129 to EUR 6,393) is much smaller than the difference in price between fuel types. Automatic cars tend to be stronger (79kW) than cars with manual transmission (69kW). The difference in mileage is also smaller than in fuel type, with manual transmission cars being driven about 3,000 kilometres more than automatic. It appears the majority of the vehicles belonging to the top 20 makers are more likely to be of manual transmission and gasoline powered as the average engine power for both (69kW and 68kW respectively) aligns more with the average power for the top makers (67-73kW).

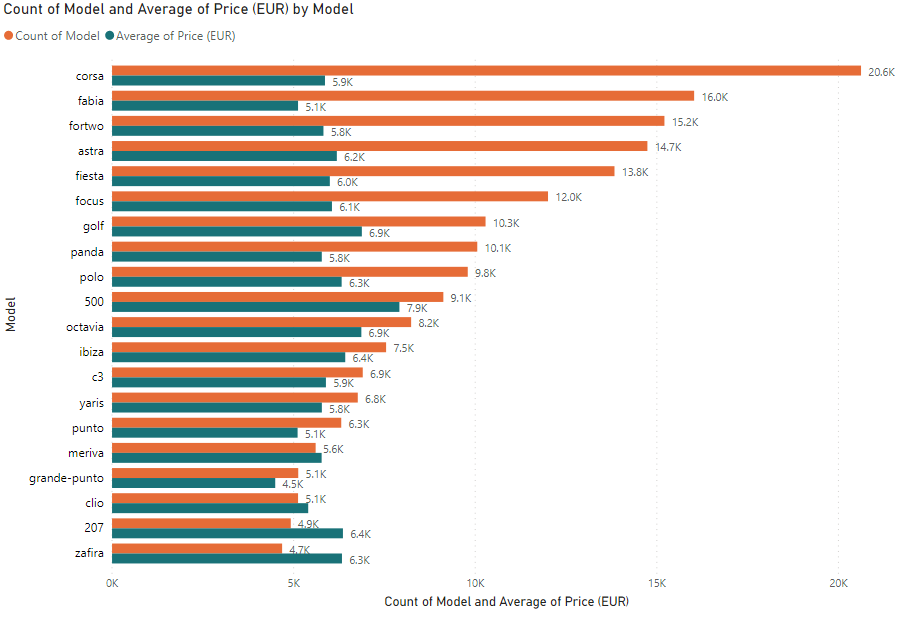
Common Models

The final part of analysis was trying to determine what the most common models were and if there were correlations between them. Average mileage, engine power and price were observed to identify the correlations. First, a bar and line graph was constructed containing the count of cars by model, the average mileage and engine power. The graph was sorted in descending order by count of models from left to right. The top 20 models are Corsa, Fabia, Fortwo, Astra, Fiesta, Focus, Golf, Panda, Polo, 500, Octavia, Ibiza, C3, Yaris, Punto, Meriva, Grande-Punto, Clio, 207 and Zafira. Figure 6 displays a wide range of kilometres driven as well as inconsistent engine power levels. Therefore, those two factors do not appear to be very relevant in determining what model is most popular.



**Figure 6**

The next factor analyzed was price. A horizontal bar graph was made to try to draw a correlation between model and price, with the models sorted in descending order by count. Looking at the figure it can be seen that the majority of the top 20 models are between EUR 5,000 and EUR 6,500. The only vehicles in the top 20 that fall out of this range are Golf, 500, Octavia and the Grande-Punto. This price range is generally affordable for those looking for used cars but also likely not deemed as being too cheap. People look to the used car market to avoid the higher prices found at dealerships for used cars. Another correlation found between these vehicles by looking at the names of the models is that all of them are available as a hatchback or 4-door sedan. This style appears to be very popular as well and it may be worthwhile to invest in this type of vehicle.



**Figure 7**

**Recommendations**

After thorough analysis, it is recommended that Big Data Investment purchase vehicles that fall into the following categories:

* Cars valued between EUR 5,500 to EUR 6,500; prices in this range appear often throughout the report. Smart cars are to be avoided due to low mileage yet similar price to its competitors. This indicates users were likely unhappy with the vehicle and quick to sell after very little usage. Volkswagen and Audi had the highest average price, however they had the highest usage in terms of average kilometres driven. Audi’s average price is beyond the desired range but it would be useful to invest in Volkswagen as the average price is just over EUR 6,700 (only EUR 200 beyond the desired range), is the 4th most common maker and has the 2nd highest average mileage (only behind Audi). This highlights the ability of Volkswagen to maintain its value over time which translates to higher resale value.
* Cars manufactured between the years 2006-2009; average prices before 2006 are much cheaper and have been heavily used. Average prices for vehicles made after 2009 increase beyond EUR 7,000, which is outside of the range found to be the most common in this dataset.
* Cars where the engine power is between 67-73kW. Cars with manual transmission tend to fall into this range, as do cars powered by gasoline.
* Hatchbacks and 4-door sedans as they make up the entirety of the top 20 models list. Most of these models also fall into the aforementioned price range except for the Golf, 500, Grande-Punto, and Octavia. While it appears that excluding Golf is contradictory to the prior recommendation of Volkswagen, it must be noted that the Volkswagen Polo falls into the desired price range and kilometres driven.

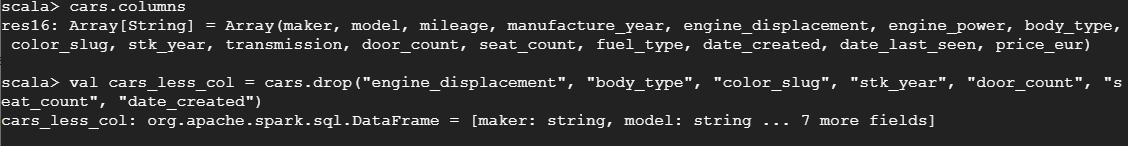
# Appendix

## Data Cleaning (Using Apache Spark)

First, all columns were listed (***cars.columns***) and then unwanted columns were removed by creating a new dataframe titled “***cars\_less\_col***”. The undesired columns were dropped using the “***drop***” function. The new dataframe was created with the following code:

***val cars\_less\_col - cars.drop(“engine\_displacement”, “body\_type”, “color\_slug”, “stk\_year”, “door\_count”, “seat\_count”, “date\_created”)***

The resulting dataframe had 9 columns, as indicated in the final line of Image 1.

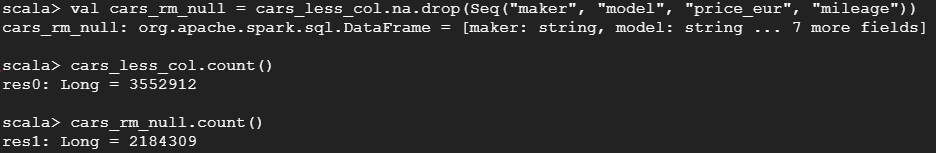


**Image 1**

After creating the new dataframe, the “***count***” function was used to confirm that none of the rows were lost. The initial data had 3,552,912 rows. Using “***cars\_less\_col.count()***” it was revealed that the new dataframe had the same number of rows. It was then decided to remove all null values in the maker, model, price\_eur and mileage fields using “***.na.drop()***”. A new dataframe titled “cars\_rm\_null” was created with the following code:

***val cars\_rm\_null = cars\_less\_col.na.drop(Seq(“maker, “model”, “price\_eur”. “mileage”))***

Another use of the “***count***” function was used to confirm that the new dataframe had less rows than the previous one. The count for cars\_no\_null was 2,184,309. All of this can be seen in Image 2.



**Image 2**

The initial plan was to look at the vehicles viewed most recently, so the aggregate “***min***” and “***max***” functions were used on “***cars\_rm\_null***” to identify the date range in the “***date\_last\_seen***” field. The first query in Image 3 is as follows:

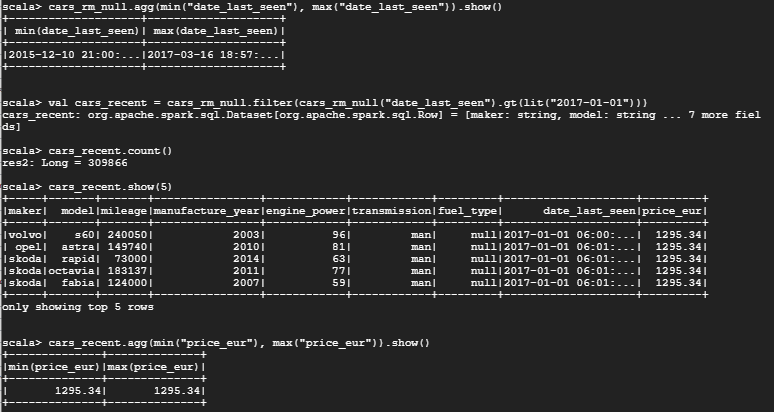
***cars\_rm\_null.agg(min(“date\_last\_seen”), max(“date\_last\_seen”)).show()***

The result revealed a range from December 10th, 2015 to March 16th, 2017. It was decided to create a new dataframe with only vehicles viewed since January 1st, 2017. This covers a two and a half month period that should indicate what kind of vehicle people are currently interested in, which may be more valuable than cars seen over a year beforehand. This was done using the following code:

***val cars\_recent = cars\_rm\_null.filter(cars\_rm\_null(“date\_last\_seen”).gt(lit(“2017-01-01”)))***

This provided vehicles having a “date last seen” greater than (gt in the code) January 1st 2017. At first it appeared this filter was effective, as it reduced our number of vehicles to 309,866 using “***cars\_recent.count()***”. However, using the “***cars\_recent.show(5)***” query it was revealed that multiple vehicles had the exact same price (EUR 1295.34). To investigate this concern, the aggregate “***min***” and “***max***” function was used again to identify the range of vehicle prices in the “cars\_recent” dataframe. The last query in Image 3 illustrates the result for the following line:

***cars\_recent.agg(min(“price\_eur”), max(“price\_eur”)).show()***



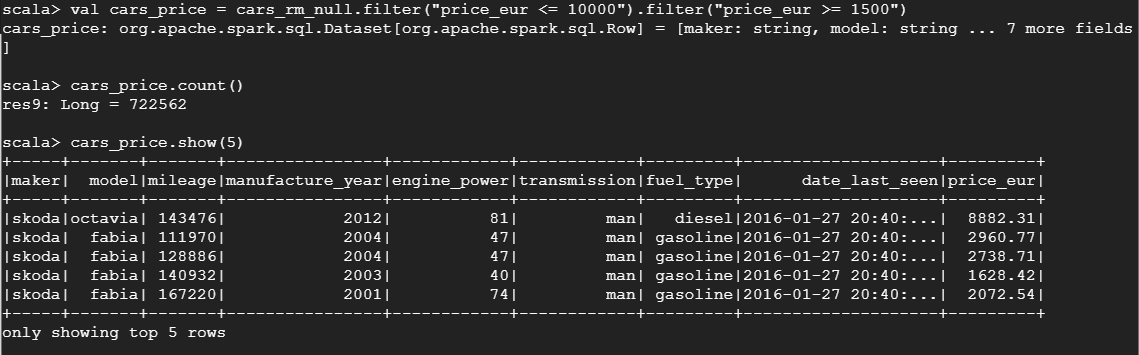
**Image 3**

It was revealed that the maximum and minimum price in this table are the same, meaning all vehicles in the dataframe have been priced at EUR 1295.34.

Following the discovery regarding the vehicles viewed in 2017, it was decided to ignore use of the “date last seen” column and focus on filtering the data using the “***price\_eur***” field. A price range between EUR 1,500 and EUR 10,000 was deemed to be appropriate as it would include a variety of makes and models.. The dataframe was created with the filter function for prices greater than EUR 1,500 and less than EUR 10,000:

***val cars\_price = cars\_rm\_null.filter(“price\_eur <= 10000”).filter(“price\_eur >= 1500”)***

Using “***count***” it was revealed that there are 722,562 cars in the latest dataframe.

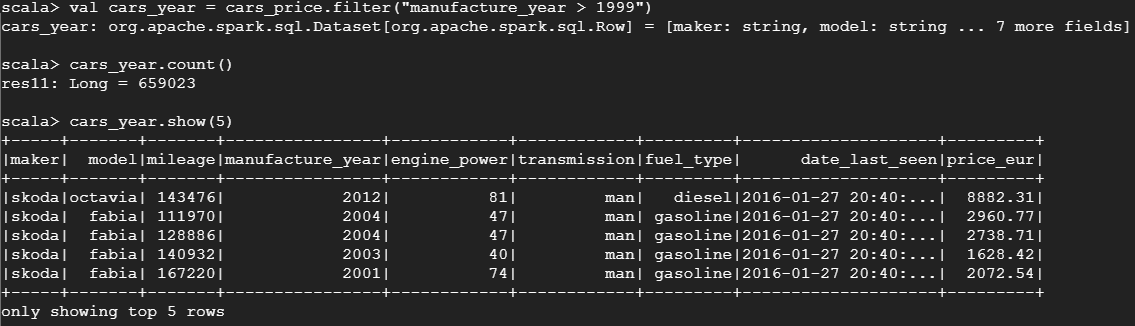


**Image 4**

Filtering by the year vehicles were manufactured was the next step. Only cars from the year 2000 and later were considered. The filter function was used to create the new dataframe “***cars\_year***” from the “***cars\_price***” dataframe:

***val cars\_year = cars\_price.filter(“manufacture\_year > 1999”)***

This removed about 63,500 vehicles, with a total of 659,023 cars remaining.



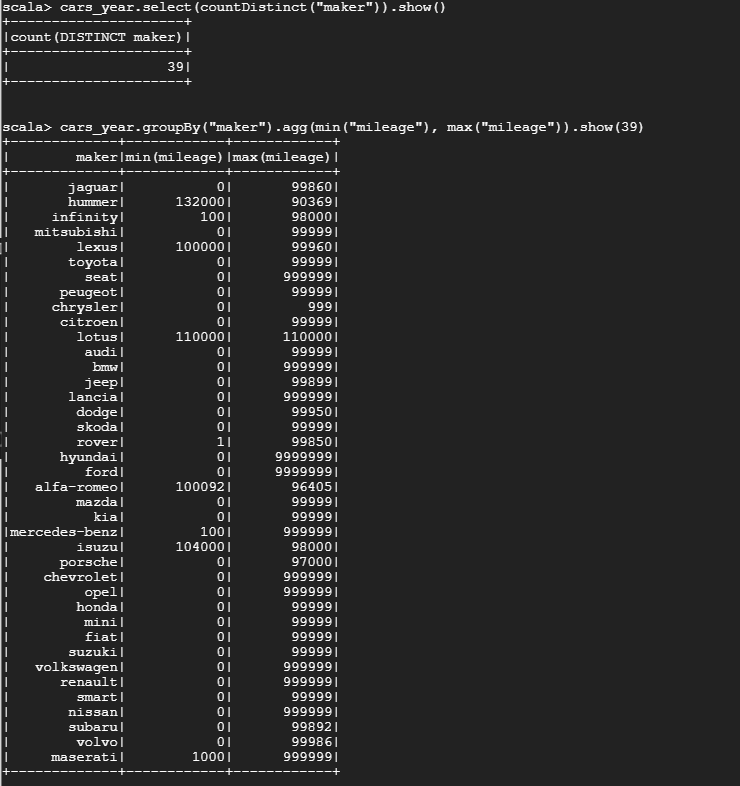
**Image 5**

Mileage was the last method of filtering used to create the dataframe to be analyzed in the report. First the “***select(countDistinct)***” function was used to determine how many makers there are, and then the “***groupBy***” function combined with aggregate “***max***” and “***min***” functions would be used to see what range of mileage was in the “***cars\_year***” dataframe for each car maker:

***cars\_year.select(countDistinct(“maker”)).show()***

Followed by:

***cars\_year.groupBy(“maker”).agg(min(“mileage”), max(“mileage”)).show(39)***

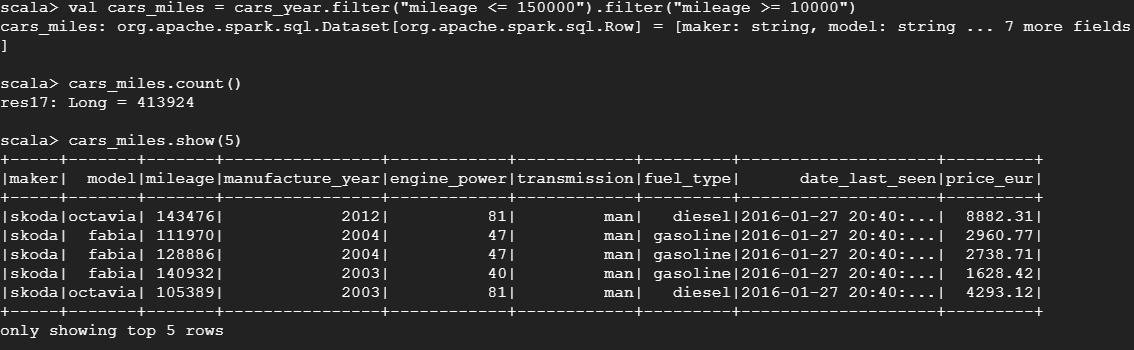


**Image 6**

Image 6 revealed that the range was very large, with some makers having a maximum mileage of 999,999 km and having a minimum of 0 km. Values at such extremes were likely to scare off potential customers and probably exist due to error, so the minimum range for the final table became 10,000 km and the maximum became 150,000 km.

***val cars\_miles = cars\_year.filter(“mileage <= 150000”).filter(“mileage >= 10000”)***

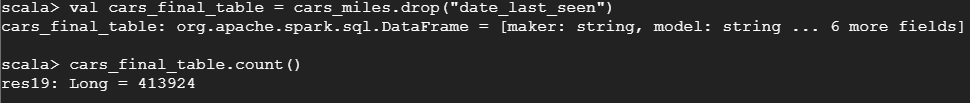
While this seems like a big range, it was believed that this would include vehicles that could attract the most customers. This range also allows analysis into how certain cars maintain value over time, allowing the company to make more informed decisions on which vehicles to buy. The number of rows for the new dataframe was reduced to 413,924.



**Image 7**

Image 8 contains the final step in creating the final “clean” dataframe is to remove a column that was almost forgotten. The “***date\_last\_seen***” was determined to have little importance as the recently viewed cars all had a value of EUR 1295.34. As this column is no longer necessary, the “***drop***” function was removed to create the final table:

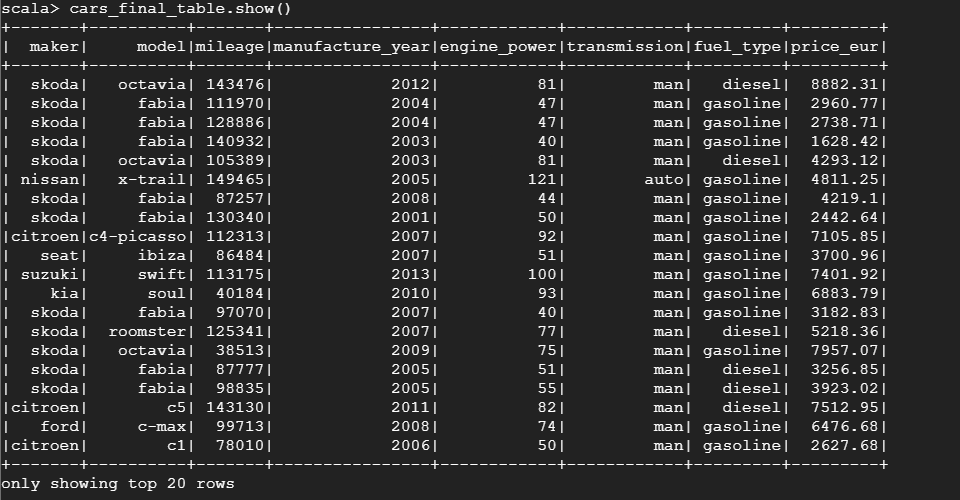
***val cars\_final\_table = cars\_miles.drop(“date\_last\_seen”)***

******

**Image 8**

The “***show***” function was used to illustrate the final clean table:

***cars\_final\_table.show(10)***



**Image 9**

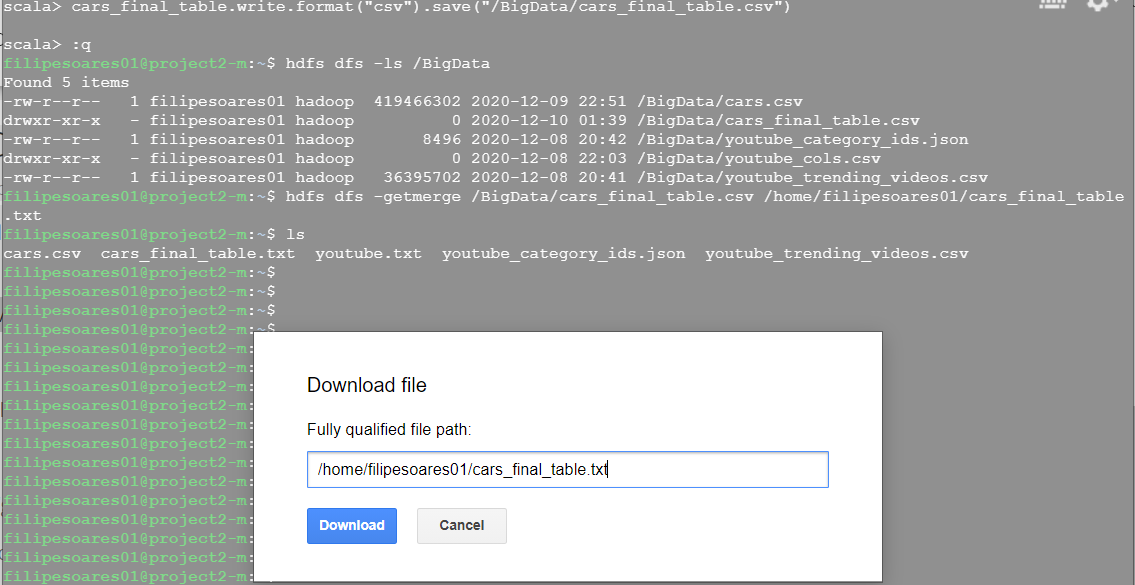
To use this data in Power BI, the ***cars\_final\_table*** dataframe was written into csv format and saved into hdfs:

***cars\_final\_table.write.format(“csv”).save(“/BigData/cars\_final\_table.csv”)***

A folder with many “part” files was created. To get the data into one main text file, the “-getmerge” function was used in hadoop and the file was saved to the local directory:

***hdfs dfs -getmerge /BigData/cars\_final\_table.csv /home/filipesoares01/cars\_final\_table.txt***

This file was then downloaded using the secure shell.



**Image 10**

## Code Used To Inspire Visualizations (all visualizations created with Power BI)

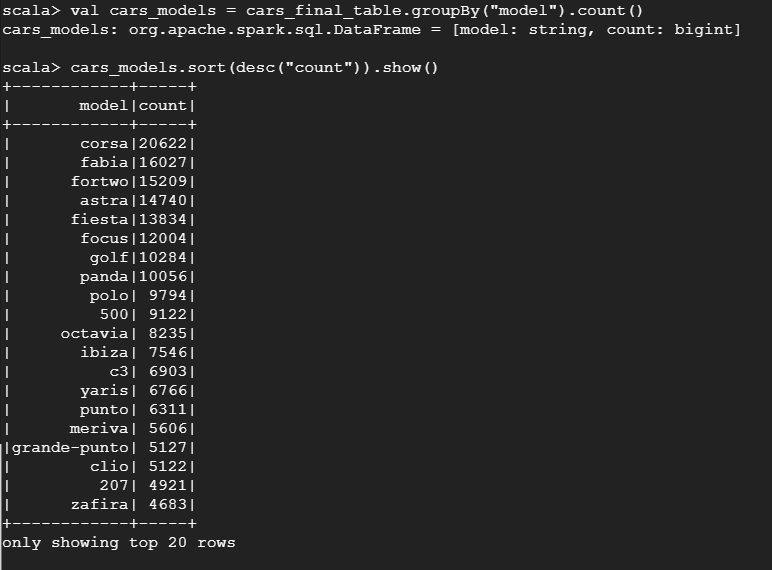
As part of the analysis it was decided that there would be an investigation into which models were the most common in the final dataframe. To view the count of models in descending order (most common to least), a separate dataframe would be created using the data from the “**cars\_final\_table**”. First the dataframe was created using the “***groupBy***” and count functions:

***val cars\_models = cars\_final\_table.groupBy(“model”).count()***

Then the new table would be viewed in descending order using the “***sort***” function:

***cars\_models.sort(desc(“count”)).show()***

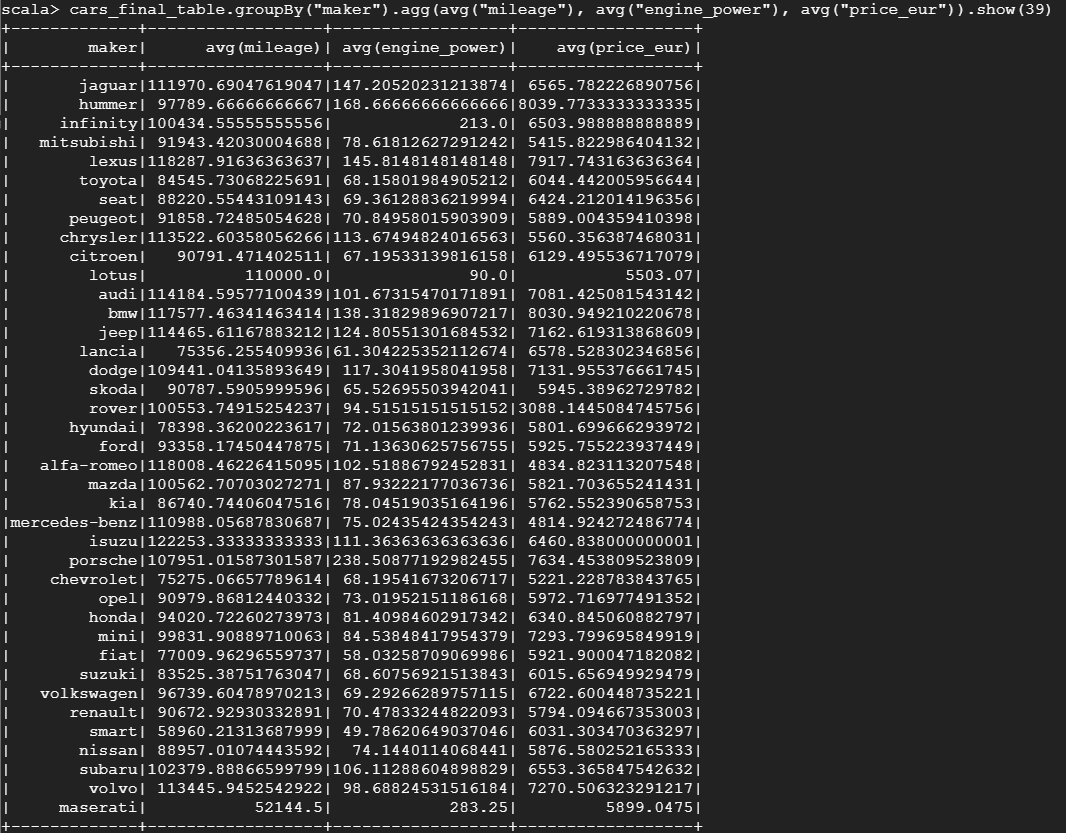
These queries were used in the creation of Figure 6 and 7 and can be found in Image 11.



**Image 11**

As seen in Image 12, the “***groupBy***” and aggregate “***avg***” functions were used to identify the average price, engine power and mileage for all 39 makers in “***cars\_final\_table***”. This inspired the creation of Figures 1 and 3:

***cars\_final\_table.groupBy(“maker”).agg(avg(“engine\_power”), avg(“price\_eur”), avg(“mileage”)).show(39)***



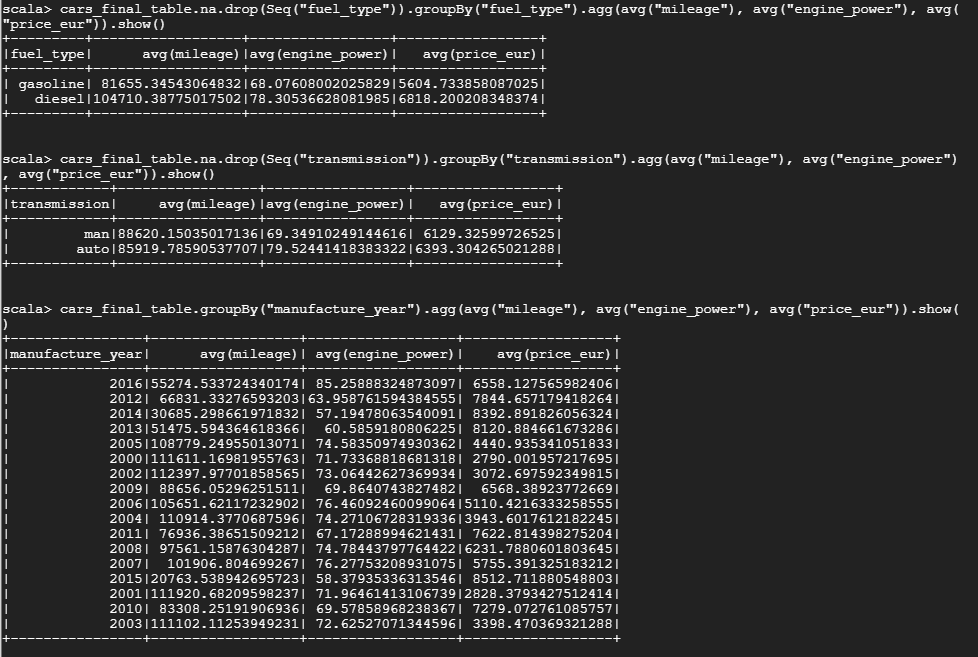
**Image 12**

The same query structure was used to view the average engine power, mileage and price for each fuel type, transmission and manufacture year, however null values for fuel type and transmission were removed by using “***na.drop(Seq)***” as they would not provide insight for the figures created. The following queries inspired the creations of Figures 2, 4 and 5 and can be viewed in Image 13.

***cars\_final\_table.na.drop(Seq(“fuel\_type”)).groupBy(“fuel\_type”).agg(avg(“engine\_power”), avg(“price\_eur”), avg(“mileage”)).show()***

***cars\_final\_table.na.drop(Seq(“transmission”)).groupBy(“transmission”).agg(avg(“engine\_power”), avg(“price\_eur”), avg(“mileage”)).show()***

***cars\_final\_table.groupBy(“manufacture\_year”).agg(avg(“engine\_power”), avg(“price\_eur”), avg(“mileage”)).show()***

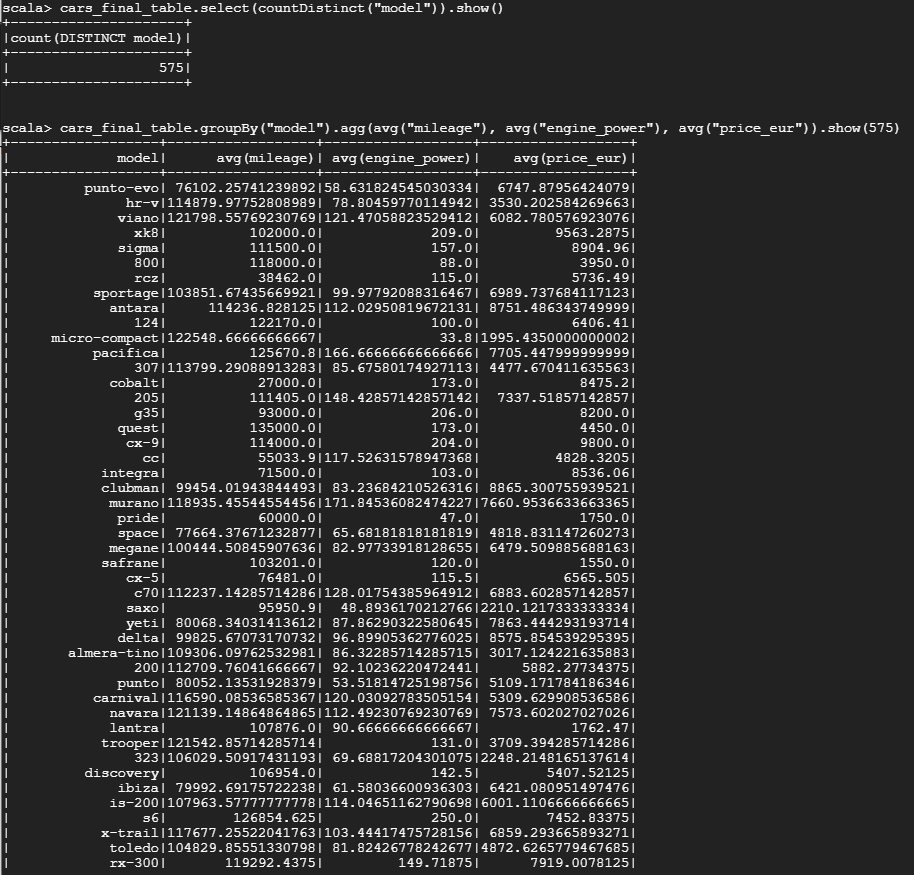


**Image 13**

The same query structure used for Image 12 and Image 13 was used for the “model” column. The vehicles were grouped by model and the average engine power, price and mileage were displayed beside each model. Image 14 displays the code used for Figures 6 and 7:

***cars\_final\_table.select(countDistinct(“model”)).show()*** identified the number of unique models in the dataset

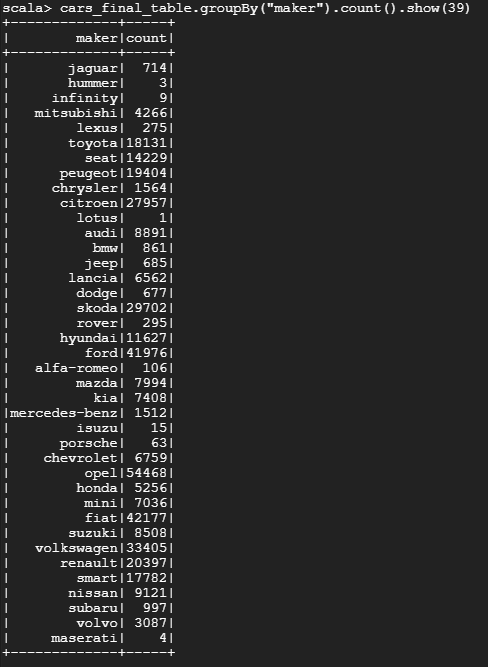
***cars\_final\_table.groupBy(“model”).agg(avg(“engine\_power”), avg(“price\_eur”), avg(“mileage”)).show(575)*** revealed the complete list of models, average price, engine power and mileage



**Image 14**

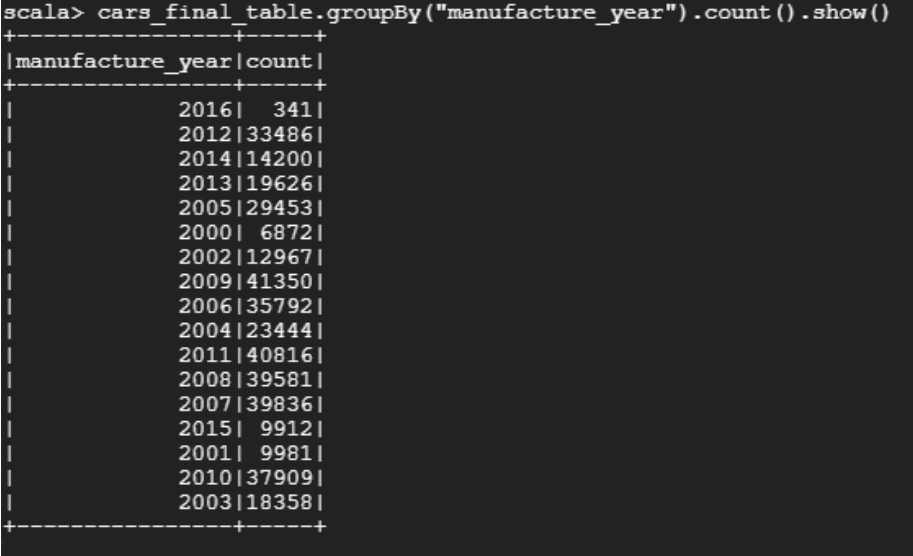
The count of vehicles belonging to different groups was also a method used to inspire the creations of figures in this report. Image 15 contains the code to group by maker. Image 16 contains the code to group by manufacture year, transmission and fuel type:

***cars\_final\_table.groupBy(“maker”).count().show(39)*** was represented in Figures 1 and 3



**Image 15**

***cars\_final\_table.groupBy(“manufacture\_year”).count().show()*** was represented in Figure 2



**Image 16**