Report

1.The memory usage of the data is around 6.1 mb. How can we reduce the memory usage of the data set?

1.Use Appropriate Data Types:

Choose the right data types for your variables. For example, use int8 or int16 instead of int32 or int64 if the values can fit within the smaller data types.

Use float32 instead of float64 if your data doesn't require high precision.

2. Downcast Numeric Data:

Downcast numeric columns using Pandas to reduce memory usage.

df['column_name'] = pd.to_numeric(df['column_name'], downcast='integer')

3. Remove Unnecessary Columns:

Remove columns that are not needed for analysis, reducing the overall memory footprint.

4. Use Categorical Data:

Convert categorical variables to the category data type, especially if the number of unique categories is significantly lower than the number of rows. This can be done using the. astype ('category') method in pandas.

5. Compress Data:

Use appropriate compression techniques, such as gzip or parquet format, especially if you are saving your data to disk. Libraries like Pandas and Dask support reading and writing compressed files.

6. Handle Missing Values Efficiently:

If a column has a lot of missing values, consider using a data type like float32 instead of float64 if possible. If appropriate, you can fill missing values with a placeholder instead of using a NaN, reducing memory usage.

7. Use Sparse Data Structures:

For matrices with a lot of zero values, consider using sparse data structures. Libraries like SciPy provide data structures for handling sparse matrices efficiently.

8. Process Data in Chunks:

When processing large datasets, consider processing the data in smaller chunks using libraries like Dask or Modin. These libraries can handle data in chunks, reducing the overall memory requirement.

9. Optimize Text Data:

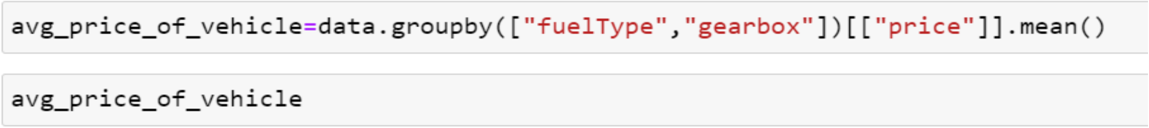
If your dataset contains text data, consider techniques like text encoding, stemming, or lemmatization to reduce the size of text fields.

10. Use External Storage or Databases:

For extremely large datasets, consider storing the data in external databases (e.g., SQL databases) and fetching only the necessary parts of the data into memory when needed.

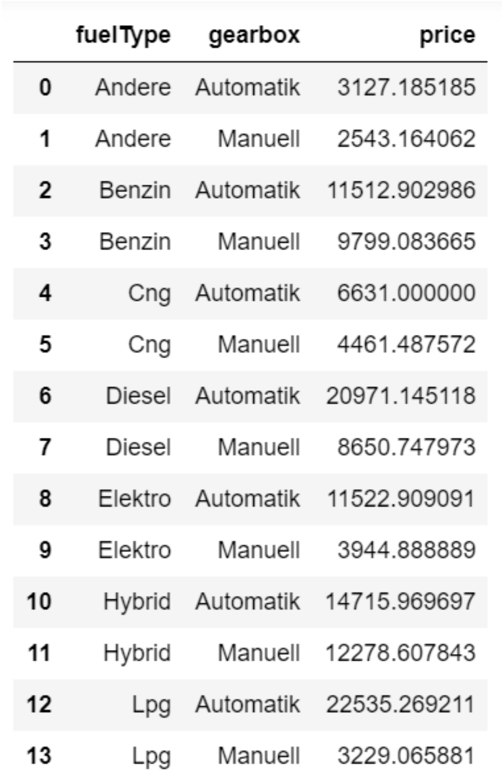
2. What is the Average price of vehicle by fuel type and gearbox type. Give a plot.

Here is the Average price of vehicle by fuel type and gearbox type by using plot

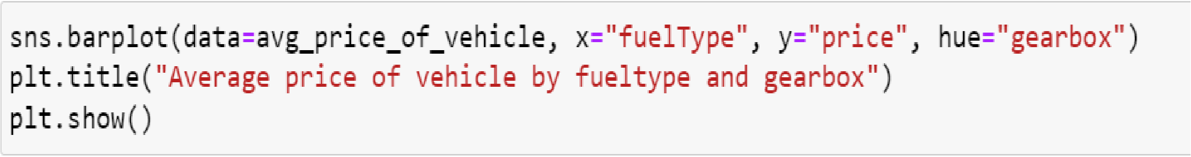


```
avg_price_of_vehicle=data.groupby(["fuelType","gearbox"])[["price"]].mean()
```

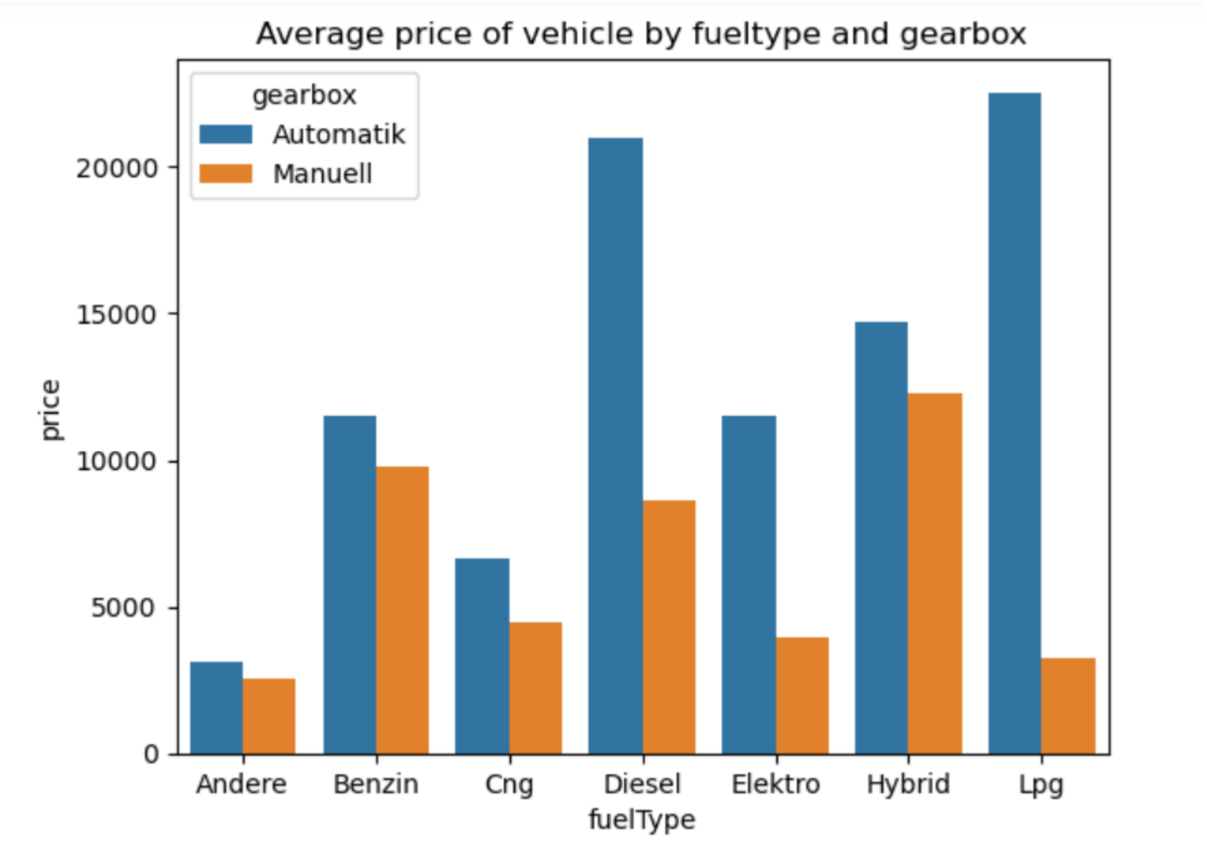
```
avg_price_of_vehicle
```



	fuelType	gearbox	price
0	Andere	Automatik	3127.185185
1	Andere	Manuell	2543.164062
2	Benzin	Automatik	11512.902986
3	Benzin	Manuell	9799.083665
4	Cng	Automatik	6631.000000
5	Cng	Manuell	4461.487572
6	Diesel	Automatik	20971.145118
7	Diesel	Manuell	8650.747973
8	Elektro	Automatik	11522.909091
9	Elektro	Manuell	3944.888889
10	Hybrid	Automatik	14715.969697
11	Hybrid	Manuell	12278.607843
12	Lpg	Automatik	22535.269211
13	Lpg	Manuell	3229.065881

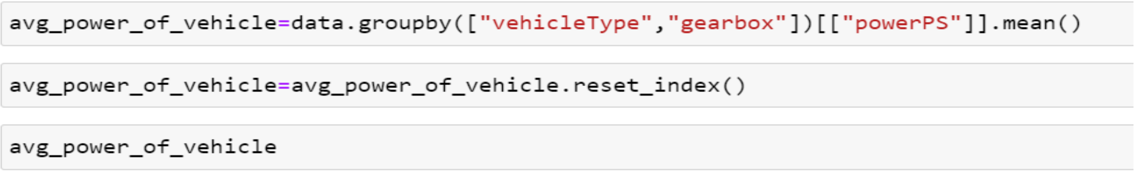


```
sns.barplot(data=avg_price_of_vehicle, x="fuelType", y="price", hue="gearbox")
plt.title("Average price of vehicle by fueltype and gearbox")
plt.show()
```



3.What is the Average power of a vehicle by vehicle type and gearbox type. Give a plot.

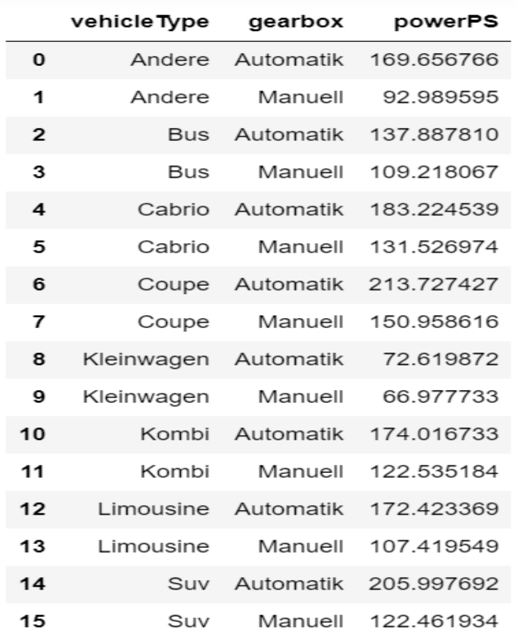
The Average power of a vehicle by vehicle type and gearbox type by using a plot.



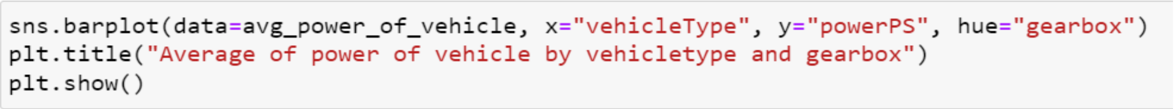
```
avg_power_of_vehicle=data.groupby(["vehicleType","gearbox"])[["powerPS"]].mean()
```

```
avg_power_of_vehicle=avg_power_of_vehicle.reset_index()
```

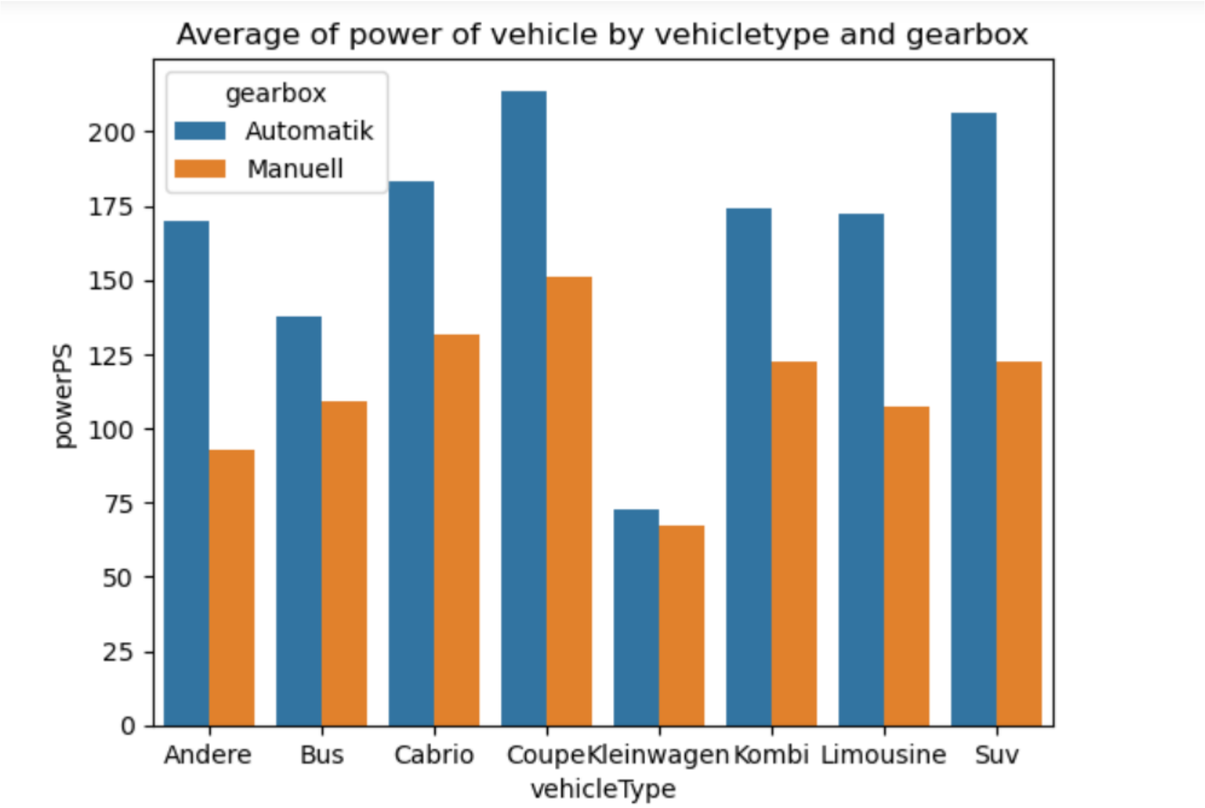
```
avg_power_of_vehicle
```



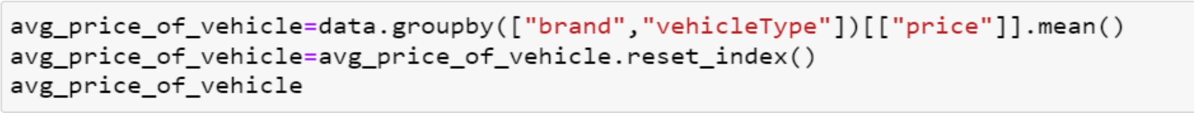
	vehicleType	gearbox	powerPS
0	Andere	Automatik	169.656766
1	Andere	Manuell	92.989595
2	Bus	Automatik	137.887810
3	Bus	Manuell	109.218067
4	Cabrio	Automatik	183.224539
5	Cabrio	Manuell	131.526974
6	Coupe	Automatik	213.727427
7	Coupe	Manuell	150.958616
8	Kleinwagen	Automatik	72.619872
9	Kleinwagen	Manuell	66.977733
10	Kombi	Automatik	174.016733
11	Kombi	Manuell	122.535184
12	Limousine	Automatik	172.423369
13	Limousine	Manuell	107.419549
14	Suv	Automatik	205.997692
15	Suv	Manuell	122.461934



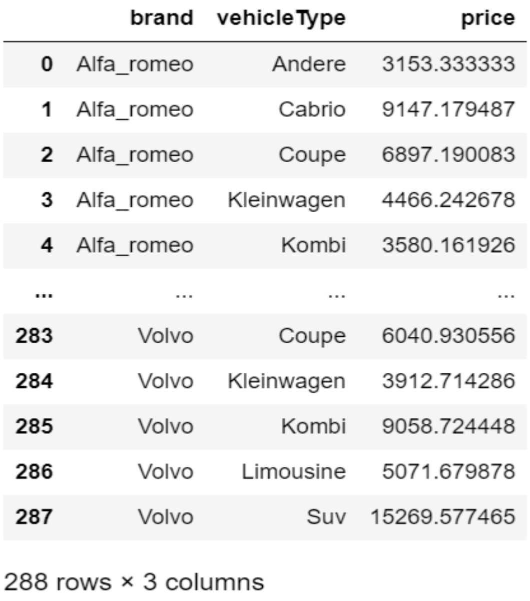
```
sns.barplot(data=avg_power_of_vehicle, x="vehicleType", y="powerPS", hue="gearbox")
plt.title("Average of power of vehicle by vehicletype and gearbox")
plt.show()
```



4. What is the Average price of a vehicle by brand as well as vehicle type. Use heatmap to explain this.

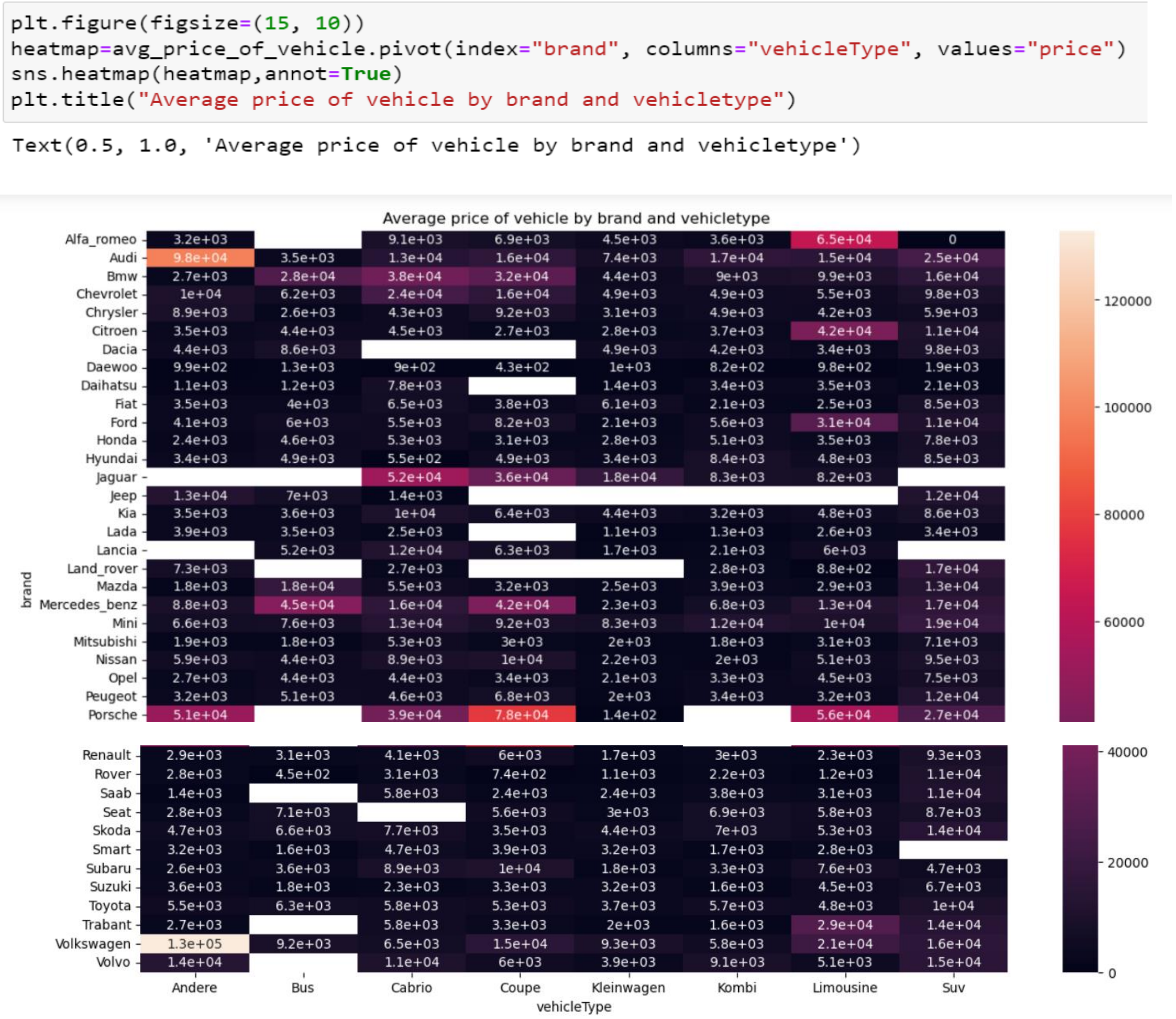


```
avg_price_of_vehicle=data.groupby(["brand","vehicleType"])[["price"]].mean()
avg_price_of_vehicle=avg_price_of_vehicle.reset_index()
avg_price_of_vehicle
```



	brand	vehicleType	price
0	Alfa_romeo	Andere	3153.333333
1	Alfa_romeo	Cabrio	9147.179487
2	Alfa_romeo	Coupe	6897.190083
3	Alfa_romeo	Kleinwagen	4466.242678
4	Alfa_romeo	Kombi	3580.161926
...	...	...	...
283	Volvo	Coupe	6040.930556
284	Volvo	Kleinwagen	3912.714286
285	Volvo	Kombi	9058.724448
286	Volvo	Limousine	5071.679878
287	Volvo	Suv	15269.577465

288 rows × 3 columns



```
plt.figure(figsize=(15, 10))
heatmap=avg_price_of_vehicle.pivot(index="brand", columns="vehicleType", values="price")
sns.heatmap(heatmap,annot=True)
plt.title("Average price of vehicle by brand and vehicletype")
```

```
Text(0.5, 1.0, 'Average price of vehicle by brand and vehicletype')
```

Average price of vehicle by brand and vehicletype

brand	Andere	Bus	Cabrio	Coupe	Kleinwagen	Kombi	Limousine	Suv
Alfa_romeo	3.2e+03		9.1e+03	6.9e+03	4.5e+03	3.6e+03	6.5e+04	0
Audi	9.8e+04	3.5e+03	1.3e+04	1.6e+04	7.4e+03	1.7e+04	1.5e+04	2.5e+04
Bmw	2.7e+03	2.8e+04	3.8e+04	3.2e+04	4.4e+03	9e+03	9.9e+03	1.6e+04
Chevrolet	1e+04	6.2e+03	2.4e+04	1.6e+04	4.9e+03	4.9e+03	5.5e+03	9.8e+03
Chrysler	8.9e+03	2.6e+03	4.3e+03	9.2e+03	3.1e+03	4.9e+03	4.2e+03	5.9e+03
Citroen	3.5e+03	4.4e+03	4.5e+03	2.7e+03	2.8e+03	3.7e+03	4.2e+04	1.1e+04
Dacia	4.4e+03	8.6e+03			4.9e+03	4.2e+03	3.4e+03	9.8e+03
Daewoo	9.9e+02	1.3e+03	9e+02	4.3e+02	1e+03	8.2e+02	9.8e+02	1.9e+03
Daihatsu	1.1e+03	1.2e+03	7.8e+03		1.4e+03	3.4e+03	3.5e+03	2.1e+03
Fiat	3.5e+03	4e+03	6.5e+03	3.8e+03	6.1e+03	2.1e+03	2.5e+03	8.5e+03
Ford	4.1e+03	6e+03	5.5e+03	8.2e+03	2.1e+03	5.6e+03	3.1e+04	1.1e+04
Honda	2.4e+03	4.6e+03	5.3e+03	3.1e+03	2.8e+03	5.1e+03	3.5e+03	7.8e+03
Hyundai	3.4e+03	4.9e+03	5.5e+02	4.9e+03	3.4e+03	8.4e+03	4.8e+03	8.5e+03
Jaguar			5.2e+04	3.6e+04	1.8e+04	8.3e+03	8.2e+03	
Jeep	1.3e+04	7e+03	1.4e+03					1.2e+04
Kia	3.5e+03	3.6e+03	1e+04	6.4e+03	4.4e+03	3.2e+03	4.8e+03	8.6e+03
Lada	3.9e+03	3.5e+03	2.5e+03		1.1e+03	1.3e+03	2.6e+03	3.4e+03
Lancia		5.2e+03	1.2e+04	6.3e+03	1.7e+03	2.1e+03	6e+03	
Land_rover	7.3e+03		2.7e+03			2.8e+03	8.8e+02	1.7e+04
Mazda	1.8e+03	1.8e+04	5.5e+03	3.2e+03	2.5e+03	3.9e+03	2.9e+03	1.3e+04
Mercedes_benz	8.8e+03	4.5e+04	1.6e+04	4.2e+04	2.3e+03	6.8e+03	1.3e+04	1.7e+04
Mini	6.6e+03	7.6e+03	1.3e+04	9.2e+03	8.3e+03	1.2e+04	1e+04	1.9e+04
Mitsubishi	1.9e+03	1.8e+03	5.3e+03	3e+03	2e+03	1.8e+03	3.1e+03	7.1e+03
Nissan	5.9e+03	4.4e+03	8.9e+03	1e+04	2.2e+03	2e+03	5.1e+03	9.5e+03
Opel	2.7e+03	4.4e+03	4.4e+03	3.4e+03	2.1e+03	3.3e+03	4.5e+03	7.5e+03
Peugeot	3.2e+03	5.1e+03	4.6e+03	6.8e+03	2e+03	3.4e+03	3.2e+03	1.2e+04
Porsche	5.1e+04		3.9e+04	7.8e+04	1.4e+02		5.6e+04	2.7e+04
Renault	2.9e+03	3.1e+03	4.1e+03	6e+03	1.7e+03	3e+03	2.3e+03	9.3e+03
Rover	2.8e+03	4.5e+02	3.1e+03	7.4e+02	1.1e+03	2.2e+03	1.2e+03	1.1e+04
Saab	1.4e+03		5.8e+03	2.4e+03	2.4e+03	3.8e+03	3.1e+03	1.1e+04
Seat	2.8e+03	7.1e+03		5.6e+03	3e+03	6.9e+03	5.8e+03	8.7e+03
Skoda	4.7e+03	6.6e+03	7.7e+03	3.5e+03	4.4e+03	7e+03	5.3e+03	1.4e+04
Smart	3.2e+03	1.6e+03	4.7e+03	3.9e+03	3.2e+03	1.7e+03	2.8e+03	
Subaru	2.6e+03	3.6e+03	8.9e+03	1e+04	1.8e+03	3.3e+03	7.6e+03	4.7e+03
Suzuki	3.6e+03	1.8e+03	2.3e+03	3.3e+03	3.2e+03	1.6e+03	4.5e+03	6.7e+03
Toyota	5.5e+03	6.3e+03	5.8e+03	5.3e+03	3.7e+03	5.7e+03	4.8e+03	1e+04
Trabant	2.7e+03		5.8e+03	3.3e+03	2e+03	1.6e+03	2.9e+04	1.4e+04
Volkswagen	1.3e+05	9.2e+03	6.5e+03	1.5e+04	9.3e+03	5.8e+03	2.1e+04	1.6e+04
Volvo	1.4e+04		1.1e+04	6e+03	3.9e+03	9.1e+03	5.1e+03	1.5e+04