TASK

Exploratory Data Analysis on the Automobile Data Set

Visit our website

Introduction

Summary of the data set

Column	Missing Data	Discrete/ Continuous	Notes
symboling	0	Discrete/ Categorical	Integer: ranging from -2 to 3 with -2 being safe and 3 being risky
normalized-losses	41	Continuous	Float: Losses compared to other cars. Fill missing data with average
make	0	Discrete/ Categorical	String. Car makes. 22 makes from Alfa Romero to Volvo
fuel-type	0	Discrete/ Categorical	String: Gas or diesel
aspiration	0	Discrete/ Categorical	String: Std or turbo
num-of-doors	2	Discrete/ Categorical	Integer: 2 or 4 doors. 2 missing data. Fill missing data with average for size
body-style	0	Discrete/ Categorical	String: 'convertible', 'hatchback', 'sedan', 'wagon', 'hardtop'
drive-wheels	0	Discrete/ Categorical	String: 'rwd', 'fwd', '4wd'
engine-location	0	Discrete/ Categorical	String: 'front', 'rear'
wheel-base	0	Continuous	Float – measurement
length	0	Continuous	Float – measurement
width	0	Continuous	Float – measurement
height	0	Continuous	Float – measurement
curb-weight	0	Continuous	Integer – weight measurement
engine-type	0	Discrete/ Categorical	String: dohc, ohc, ohcv, l
num-of-cylinders	0	Discrete/ Categorical	String: two to twelve. Number of engine cylinders
engine-size	0	Continuous	Integer: 61 to 329. Engine size
fuel-system	0	Discrete/ Categorical	String: 'diesel' or 'gas'
bore	4	Continuous	Float: 2.54 to 3.94. Diameter of each cylinder

stroke	4	Continuous	Float: 2.07 to 4.17. Each movement of the piston is called a stroke. Four strokes — down, up, down, up — complete the cycle that creates the power to drive the engine.
compression-ratio	0	Continuous	Float: 7 to 23. The compression ratio is the ratio between the volume of the cylinder and combustion chamber in an internal combustion engine at their maximum and minimum values.
horsepower	2	Continuous	Float: 48 to 288. Engine power
city-mpg	0	Continuous	Integer – miles per gallon in the city
highway-mpg	0	Continuous	Integer – miles per gallon on the highway
price	4	Continuous	Float – price of car

DATA CLEANING

SUMMARY OF THE METHODS AND VISUALISATIONS DONE DURING DATA CLEANING

The dataset consists of 26 columns describing car data.
The column headers are:

['symboling', 'normalized-losses', 'make', 'fuel-type', 'aspiration', 'num-of-doors', 'body-style', 'drive-wheels', 'engine-location','wheel-base', 'length', 'width', 'height', 'curb-weight', 'engine-type', num-of-cylinders', 'engine-size', 'fuel-system', 'bore', 'stroke','compression-ratio', 'horsepower', 'peak-rpm', 'city-mpg','highway-mpg', 'price']

When using the '.info' for a summary of the dataset information, it shows there are no null cells. However, viewing the data using head and tail commands shows that the nulls or NaN have been replaced by a '?'. Therefore the '?' can be replaced using '.replace('?', np.Nan)'. np.Nan is used by numpy as not a number, ie no there's entry.

Now running '.isnull().sum()' on the dataframe reveals the missing values in columns: normalised-losses(41), num-of-doors(2), bore(4), stroke(4), horsepower(2), peak-rpm(2) and price(4).

How the missing data is dealt with is below under the Missing Data paragraph header.

In doing the cleansing and missing data transformations it was noted that the columns with '-' in their title weren't responding as they should using the pandas commands. An amendment was made to their names replacing '-' with '_'. This fixed
the issue. The code for this used:
"automobile_df.columns = automobile_df.columns.str.replace('-', '_')"

Any duplicates are discarded using drop_duplicates()

Each of the columns values can be viewed using the '.unique' method. This gives you an idea of the column contents.

The price column was changed to np.int64

MISSING DATA

ANY MISSING DATA? HOW DID YOU HANDLE IT

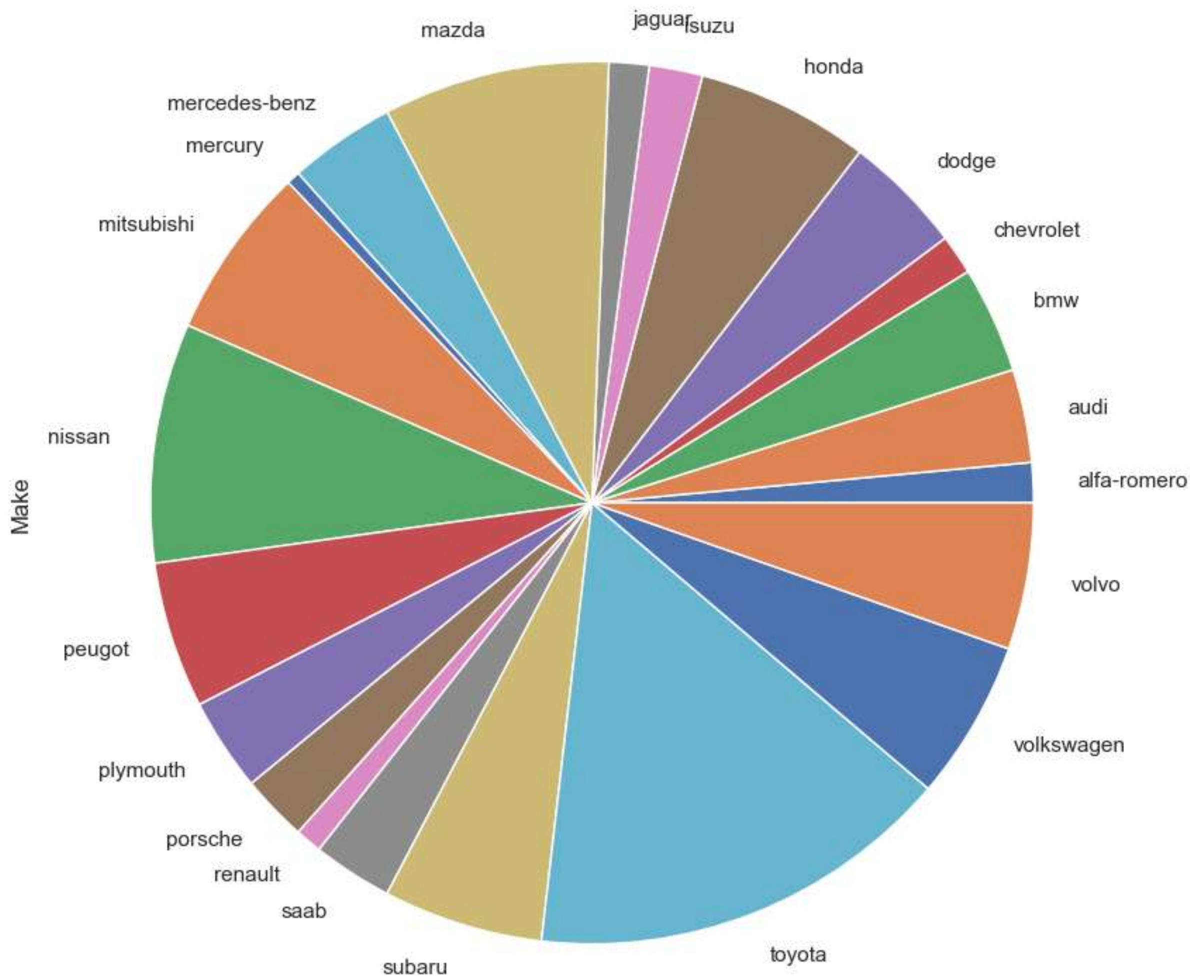
Running '.isnull().sum()' on the dataframe reveals the missing values in columns: normalised-losses(41), num-of-doors(2), bore(4), stroke(4), horsepower(2), peak-rpm(2) and price(4). A missingno.matrix can be used to show exactly where the missing data appears.

The normalised-losses missing data can be filled in with the mean for the cells. The mean is obtained by using '.dropna(how='any').mean' on the dataframe. This gives the mean as 164. This is put into the NaN cells via replace. Similarly the bore, stroke, horsepower, peak-rpm and price NaN cells are filled in. The num-of-doors NaN rows are inspected and it is noted that these have a body-style of 'Sedan'. The Sedan cars are usually bigger cars so 4 doors are entered for the num-of-doors for these rows.

DATA STORIES AND VISUALISATIONS

THIS IS THE BULK OF THIS PROJECT. EXTRACT STORIES AND ASSUMPTIONS BASED ON VISUALISATIONS OF THE DATA

Pie Chart of Makes in the Dataset
The dataset contains various makes of cars and the dataset is made of makes in the proportions shown in the pie chart below:



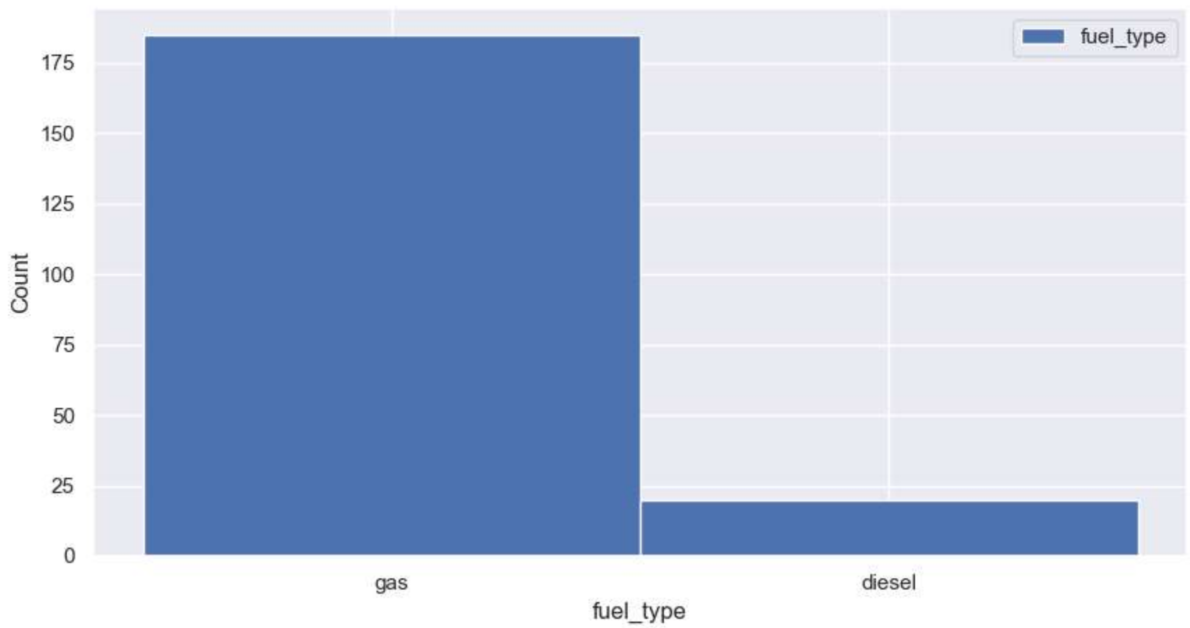
As you can see the Toyota is the make with the most entries in our dataset. Mercury appears to the make with the lowest representation in the data. This plot is a count plot pie from the Matplotlib module.

The full break down is shown in a table below:

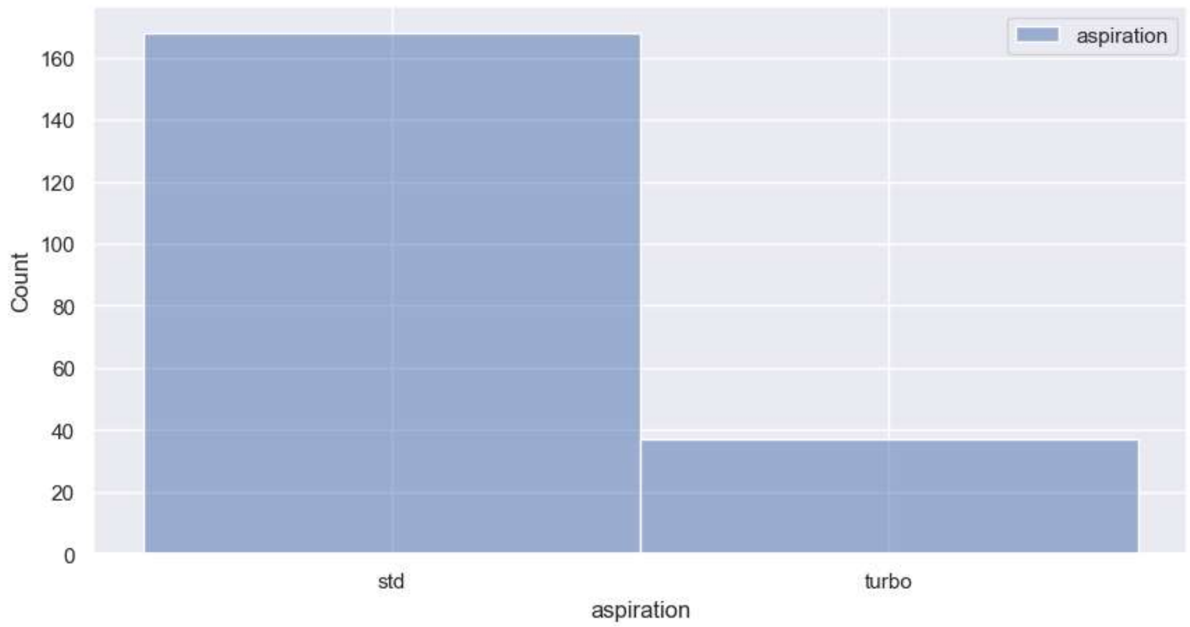
toyota	32
nissan	18
mazda	17
mitsubishi	13
honda	13
volkswagen	12
subaru	12
peugot	11
volvo	11
dodge	9
mercedes-benz	8
bmw	8
audi	7
plymouth	7
saab	6
porsche	5
isuzu	4
jaguar	3
chevrolet	3
alfa-romero	3
renault	2
mercury	1

Name: make, dtype: int64

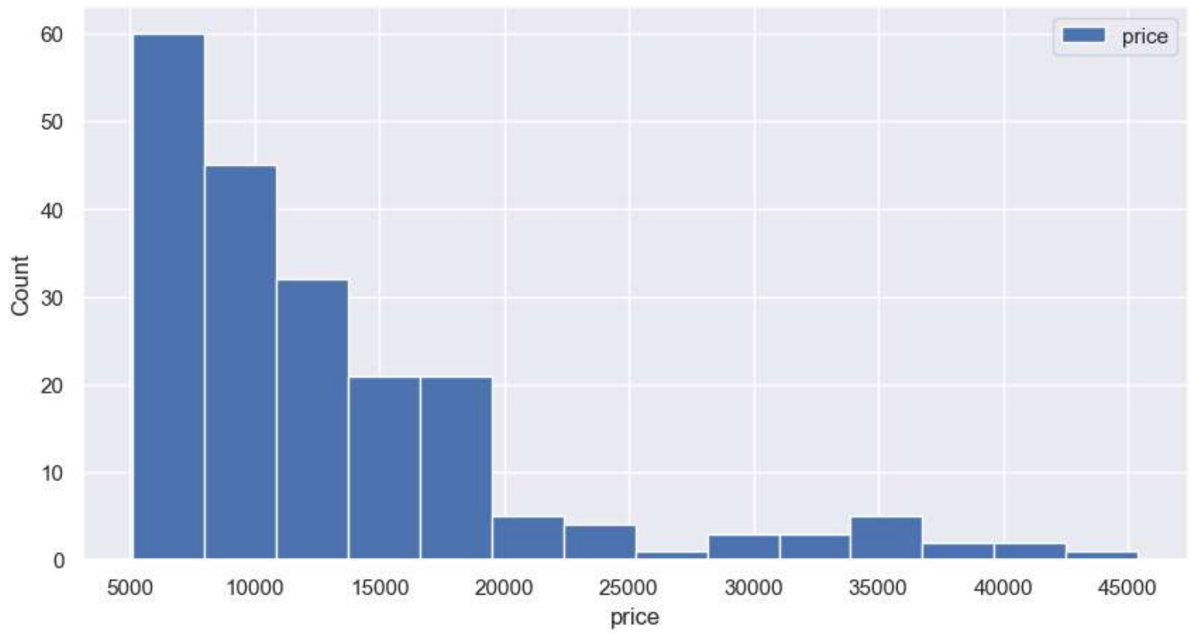
Fuel Type in the Dataset



Aspiration in the Dataset



Price Range in the Dataset



The above bar chart shows that most of the data is around the $5000 to $20000 range. We have less data for cars above $25000.

The lowest priced car was a Sabura at $5118 and the highest price car was Mercedes-Benz at $45400. These were found using the nsmallest and nlargest. The mininum price and maximum price can also be found by using min and max. The mean price for a car in the dataset was $13212.75

Highest Priced Car Makes:

	make	price
74	mercedes-benz	45400
16	bmw	41315
73	mercedes-benz	40960
128	porsche	37028
17	bmw	36880

Lowest Priced Car Makes:

	make	price
138	subaru	5118
18	chevrolet	5151
50	mazda	5195
150	toyota	5348
76	mitsubishi	5389

Grouping by Make and Averaging the Data

This table summarises average attributes by make:

	symboling	wheel_base	engine_size	city_mpg	highway_mpg	price
make						
alfa-romero	2.333333	90.566667	137.333333	20.333333	26.666667	15498.333333
audi	1.285714	102.271429	130.714286	18.857143	24.142857	17235.714286
bmw	0.375000	103.162500	166.875000	19.375000	25.375000	26118.750000
chevrolet	1.000000	92.466667	80.333333	41.000000	46.333333	6007.000000
dodge	1.000000	95.011111	102.666667	28.000000	34.111111	7875.444444
honda	0.615385	94.330769	99.307692	30.384615	35.461538	8184.692308
isuzu	0.750000	94.825000	102.500000	31.000000	36.000000	11205.750000
jaguar	0.000000	109.333333	280.666667	14.333333	18.333333	34600.000000
mazda	1.117647	97.017647	103.000000	25.705882	31.941176	10652.882353
mercedes-benz	0.000000	110.925000	226.500000	18.500000	21.000000	33647.000000
mercury	1.000000	102.700000	140.000000	19.000000	24.000000	16503.000000
mitsubishi	1.846154	95.353846	118.307692	24.923077	31.153846	9239.769231

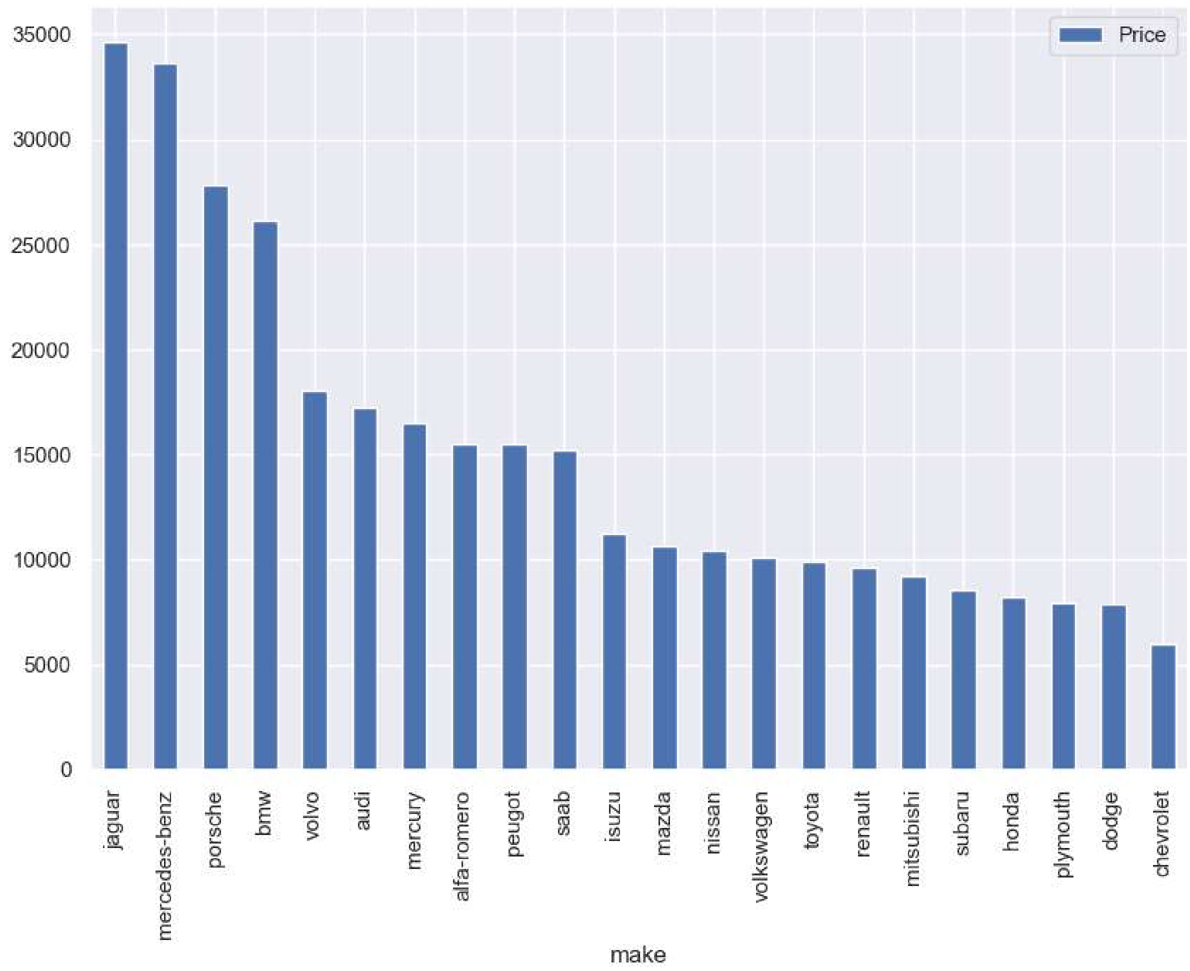
	symboling	wheel_base	engine_size	city_mpg	highway_mpg	price
make						
nissan	1.000000	95.722222	127.888889	27.000000	32.944444	10415.666667
peugot	0.000000	110.200000	135.818182	22.454545	26.636364	15489.090909
plymouth	1.000000	95.385714	106.285714	28.142857	34.142857	7963.428571
porsche	2.600000	92.280000	187.200000	17.400000	26.000000	27819.400000
renault	1.000000	96.100000	132.000000	23.000000	31.000000	9595.000000
saab	2.500000	99.100000	121.000000	20.333333	27.333333	15223.333333
subaru	0.500000	96.175000	107.083333	26.333333	30.750000	8541.250000
toyota	0.562500	98.103125	118.812500	27.500000	32.906250	9885.812500
volkswagen	1.666667	97.608333	107.250000	28.583333	34.916667	10077.500000
volvo	-1.272727	106.481818	142.272727	21.181818	25.818182	18063.181818

The Makes and their Average Prices

This table shows the makes with the lowest average prices. Chevrolet is the cheapest car make.

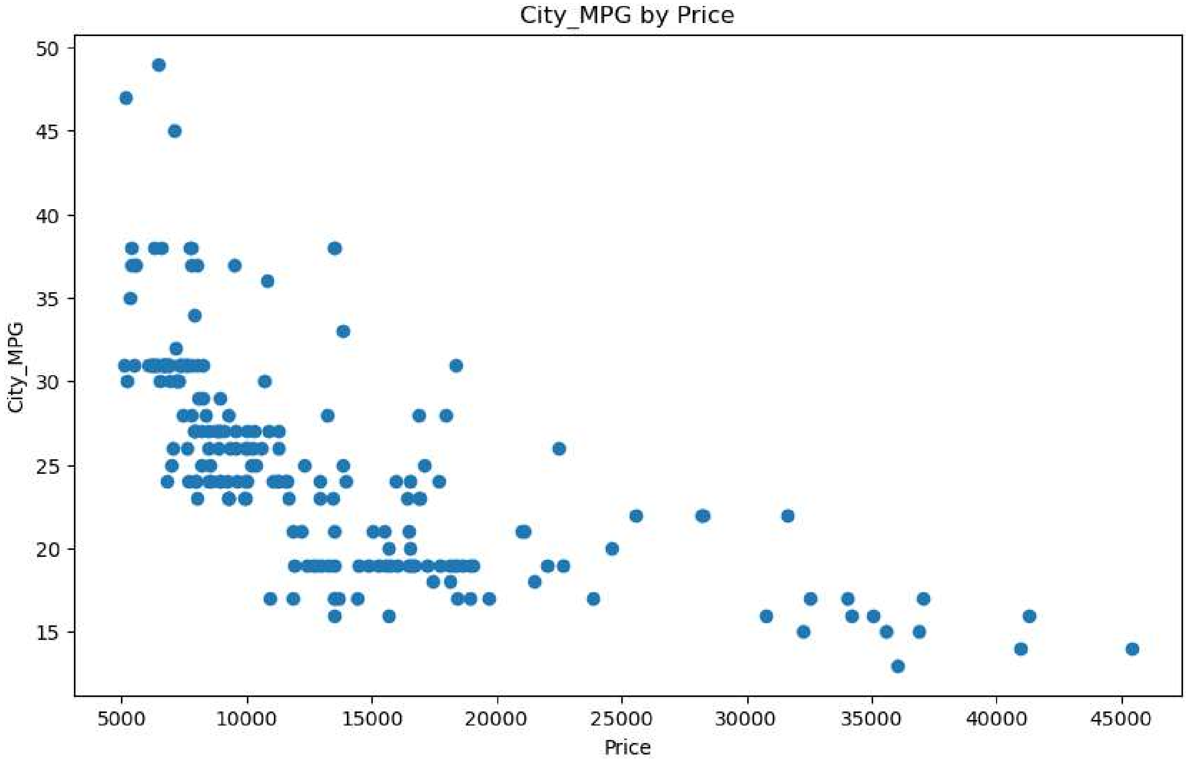
	symboling	wheel_base	engine_size	city_mpg	highway_mpg	price
make						
chevrolet	1.000000	92.466667	80.333333	41.000000	46.333333	6007.000000
dodge	1.000000	95.011111	102.666667	28.000000	34.111111	7875.444444
plymouth	1.000000	95.385714	106.285714	28.142857	34.142857	7963.428571
honda	0.615385	94.330769	99.307692	30.384615	35.461538	8184.692308
subaru	0.500000	96.175000	107.083333	26.333333	30.750000	8541.250000

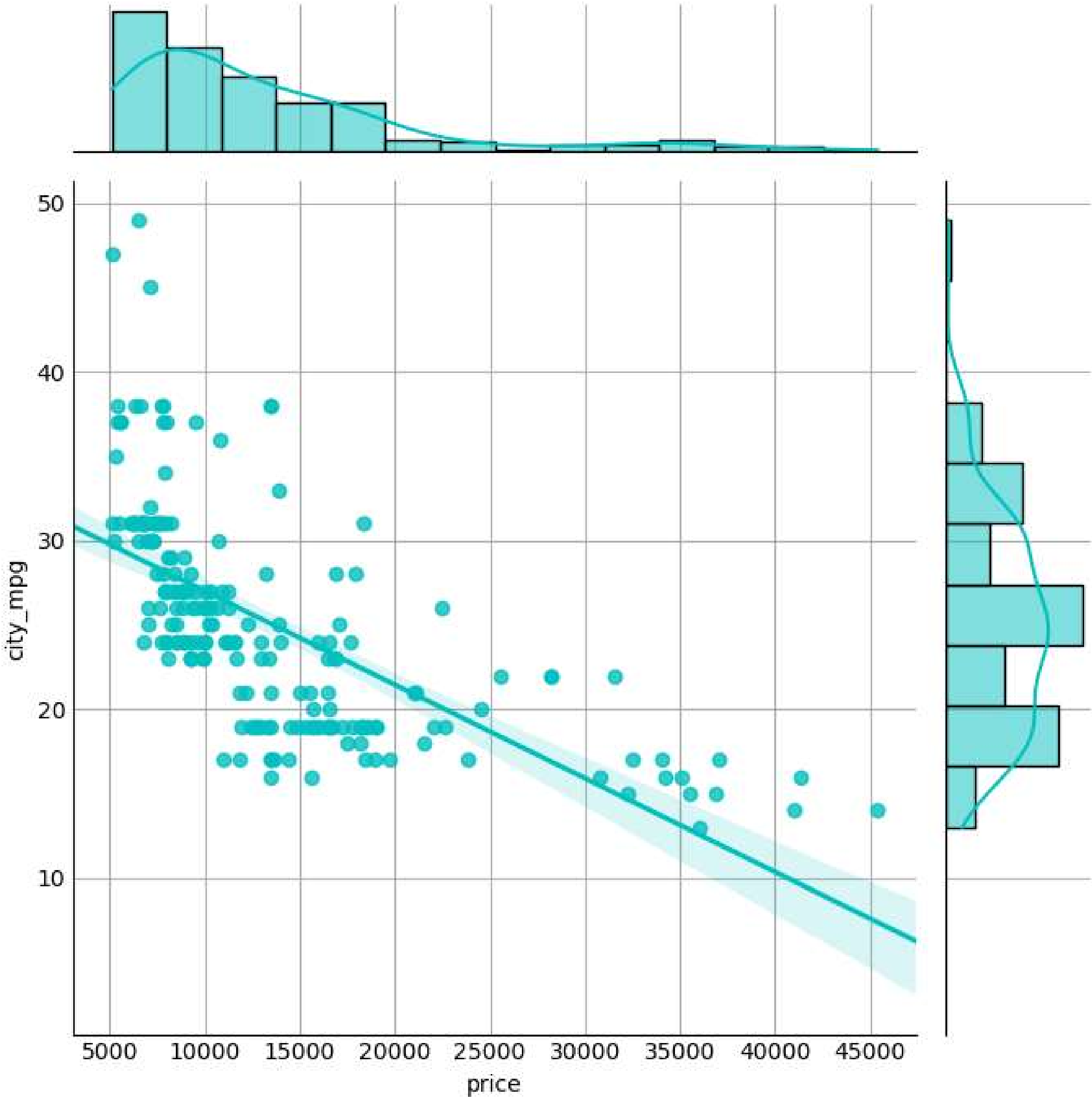
The bar plot below shows the average cost of the different makes of cars.


35000
30000
25000
20000
15000
10000
5000
0
Price
jaguar
mercedes-benz
porsche
bmw
volvo
audi
mercury
alfa-romero
peugot
saab
isuzu
mazda
nissan
volkswagen
toyota
renault
mitsubishi
subaru
honda
plymouth
dodge
chevrolet
make

Miles per Gallon in the City by Price

This plot is a scatter plot in Matplotlib of Price by Miles per Gallon in the City.



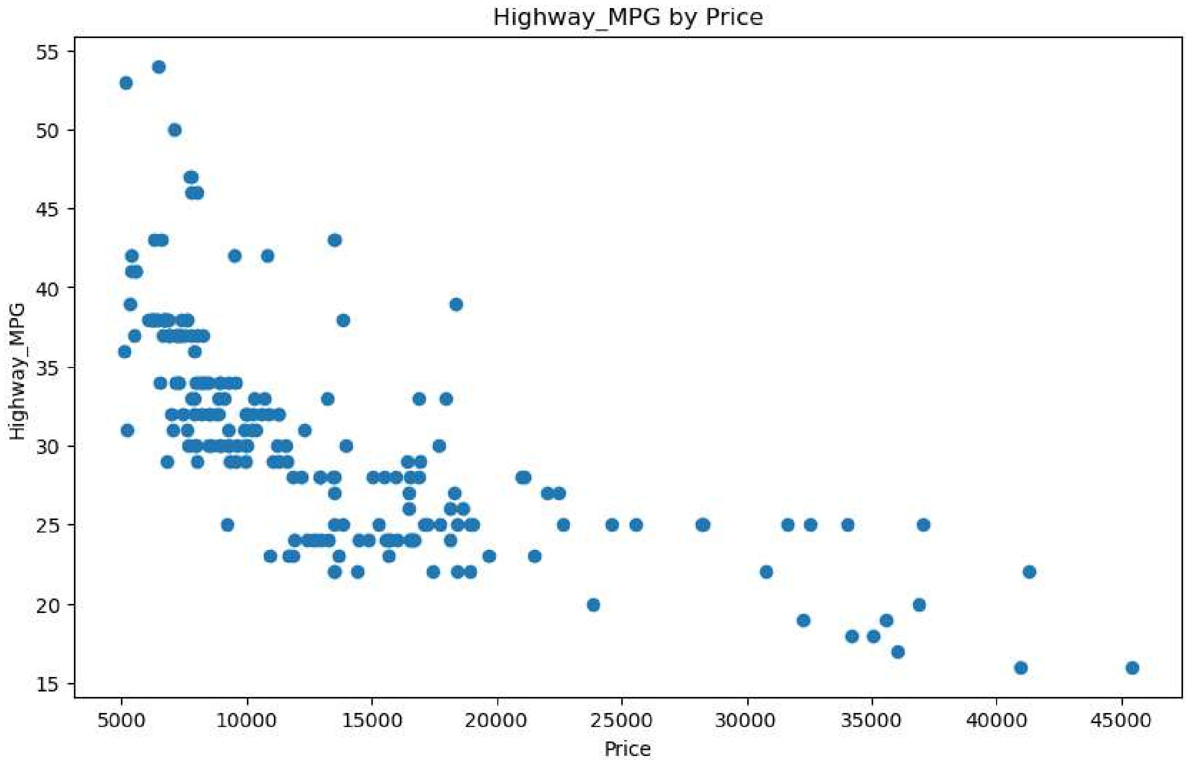


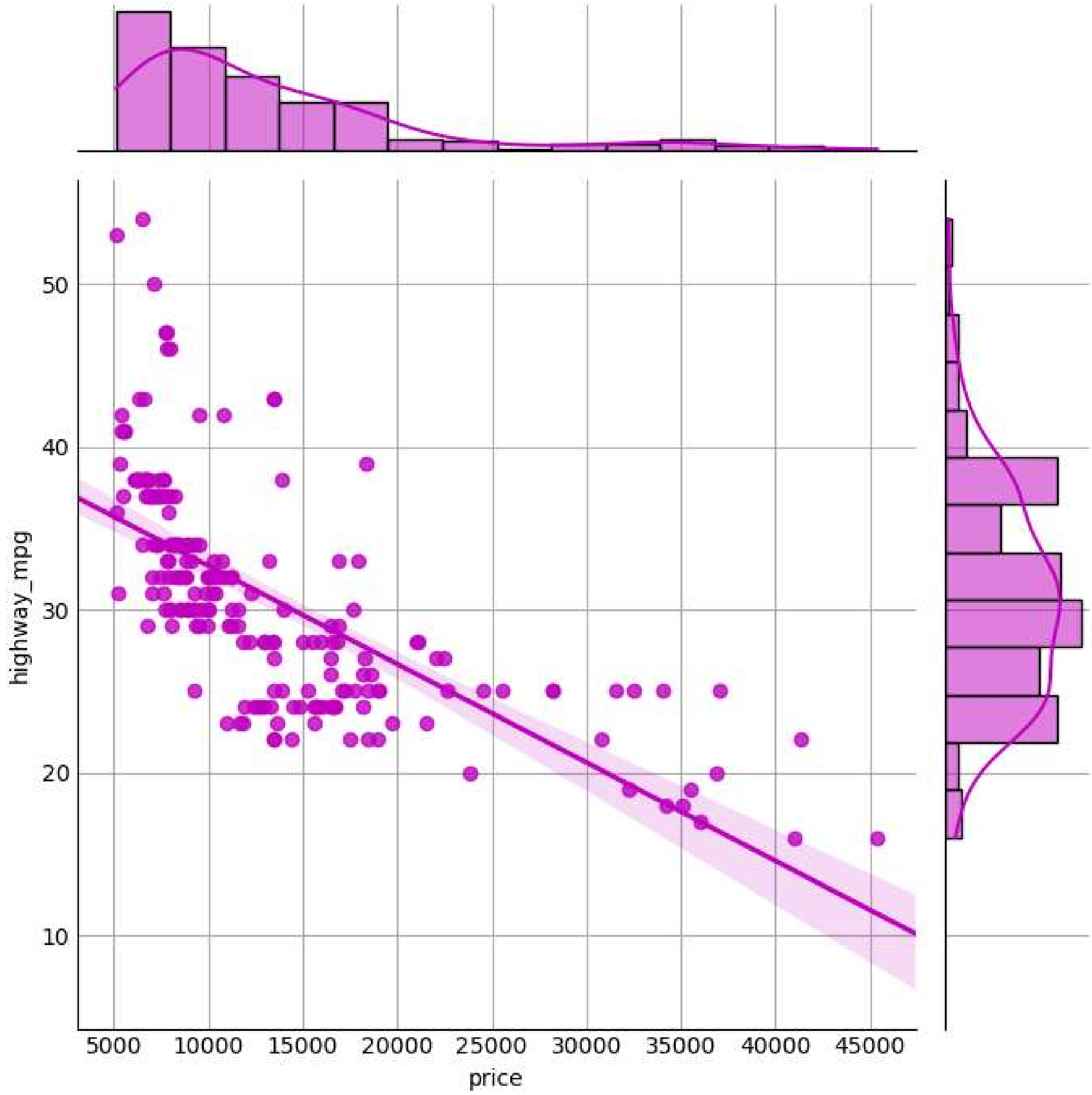
The above plot is a scatter plot in Seaborn showing price by City Miles per Gallon.

The above scatterplots show that as the price goes up the City_MPG goes down. Therefore a more expensive car will take more fuel and be more expensive to run. Most of the data we have is clustered at the lower price cars.

Miles per Gallon in the Highway by Price

This plot is a scatter plot in Matplotlib of Price by Miles per Gallon in the Highway.





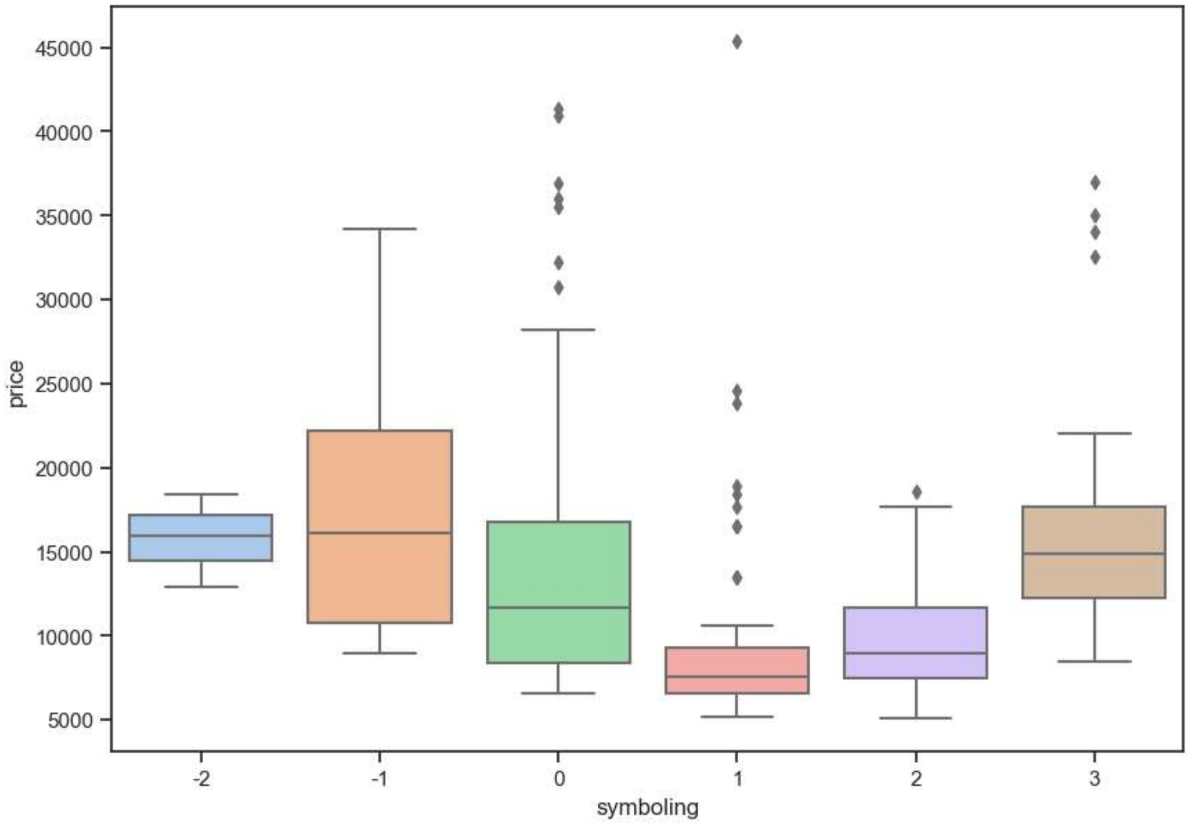
The above plot is a scatter plot in Seaborn showing price by Highway Miles per Gallon.

The above scatter plots show that as the price goes up the Highway_MPG goes down.
Therefore a more expensive car will take more fuel and be more expensive to run.
Most of the data we have is clustered at the lower price cars.

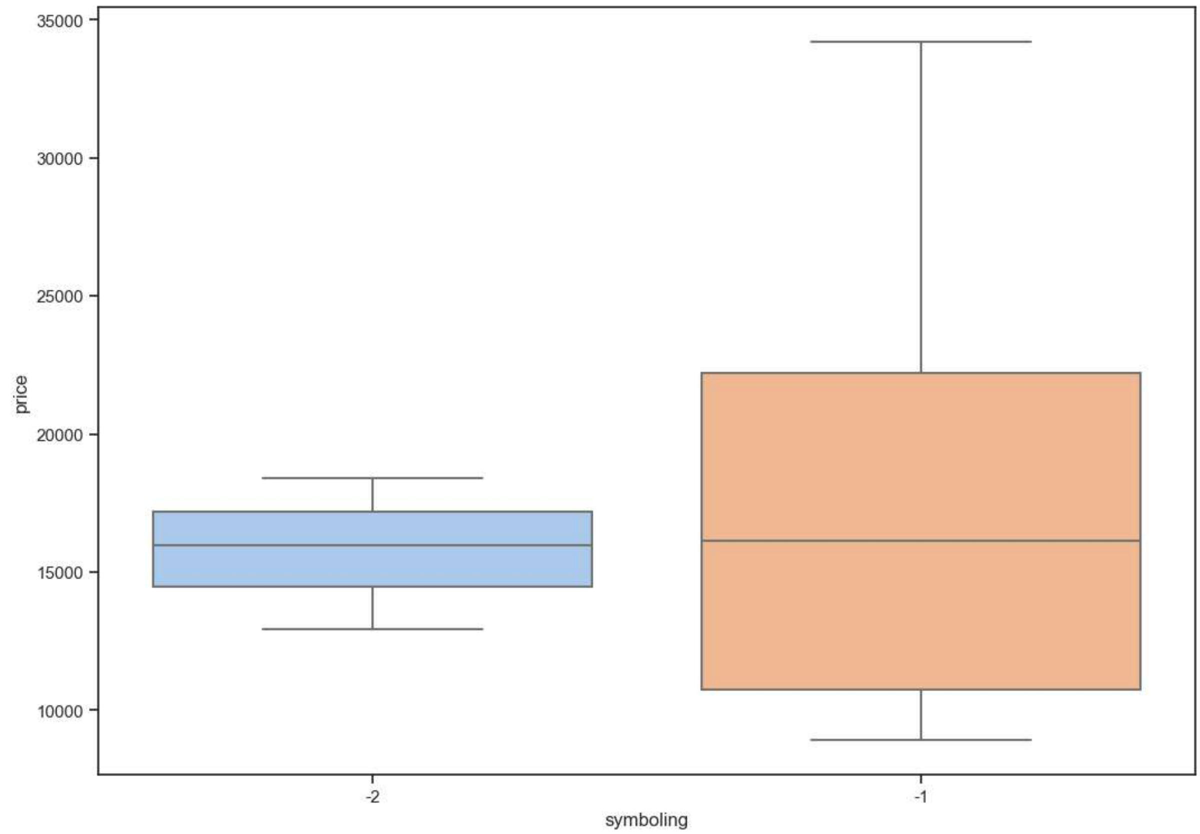
The Price of Safety

Symboling: -2 is the safest cars and 3 are the riskiest cars.

The below box plot shows the symboling against price.

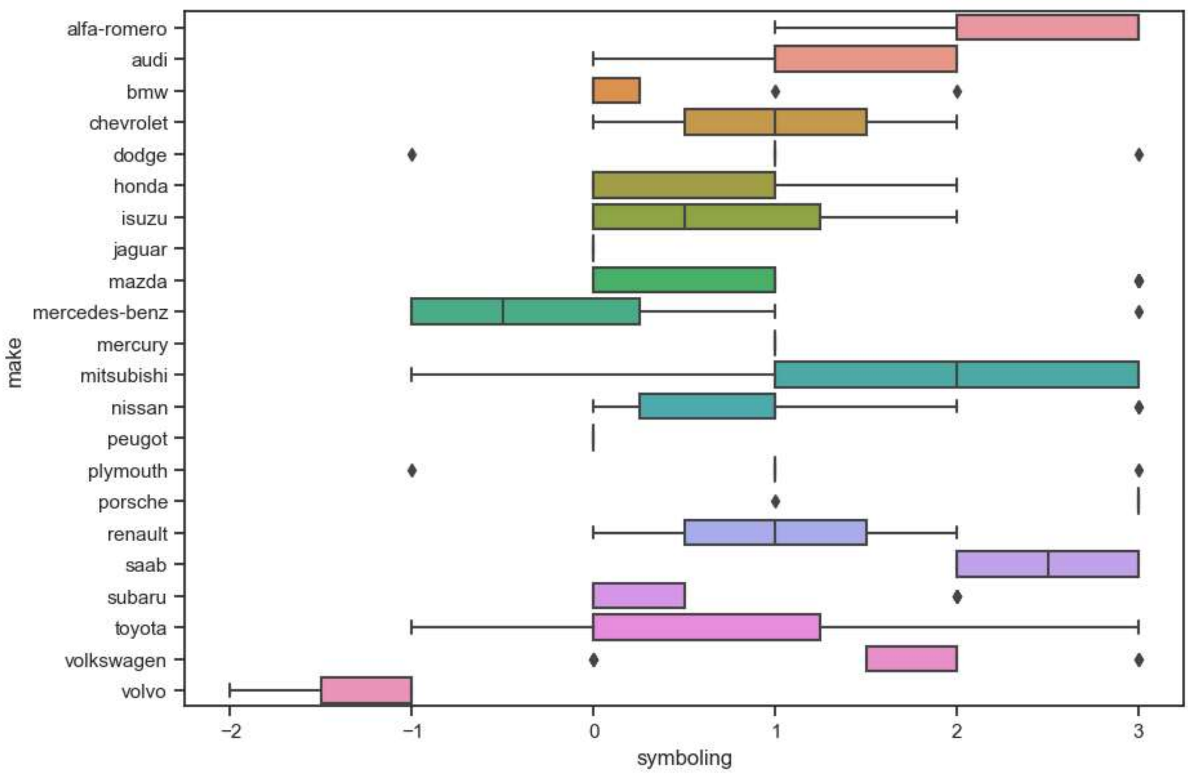


The below box plot shows the symboling against price filtered to the safest symbols of -2 and -1.



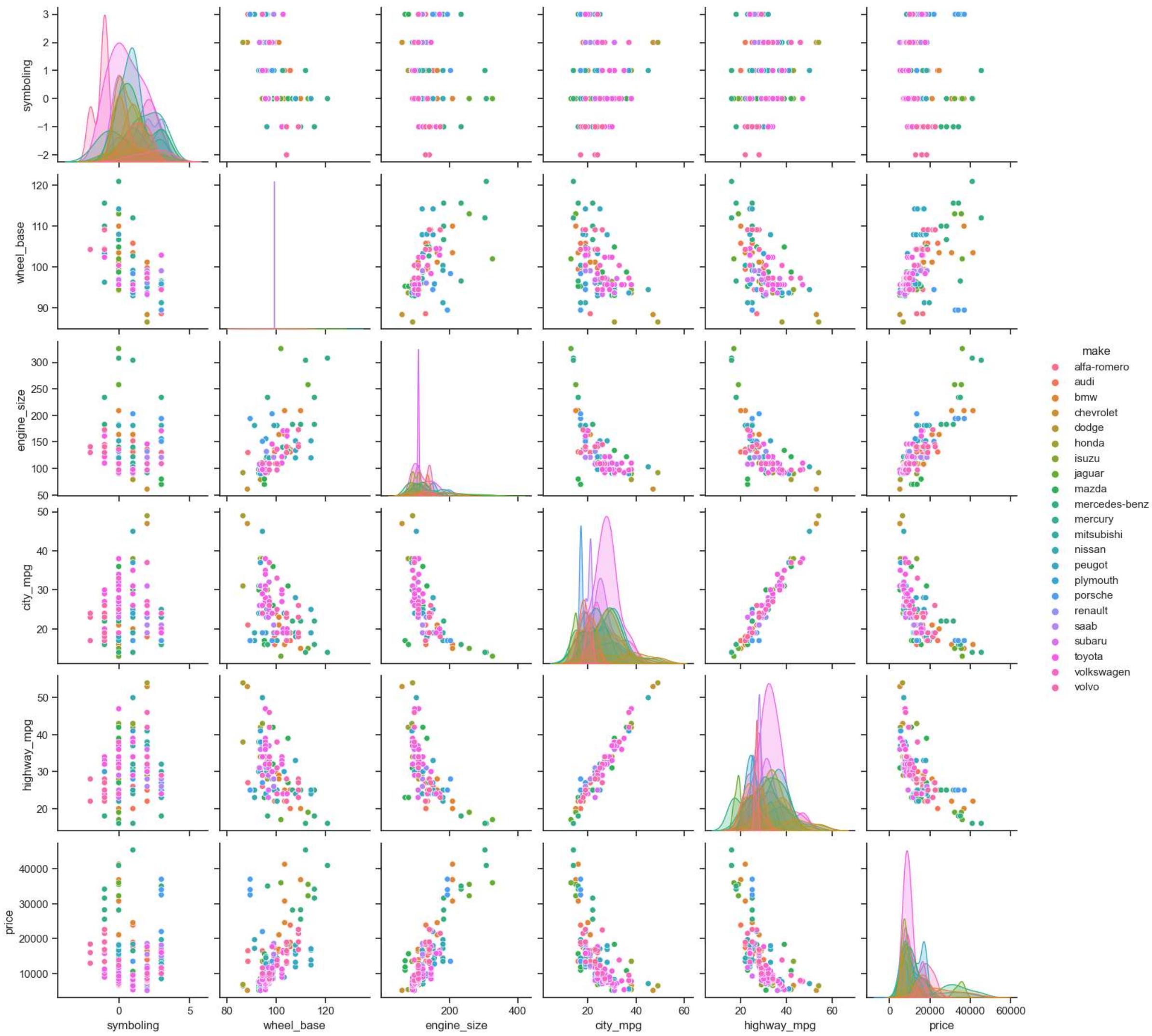
The above box plots shows that most of the safest cars can be bought for less than 17000.

The below box plot shows the makes and their symbolising category.



The above box plot, where -2 is the safest car and 3 is the riskiest, shows that Volvo and Mercedes-Benz have the safest cars. Alfa-Romero, Mitsubishi and Saab have the riskiest cars.

Scatter Plot Matrix for a wider look at the data:



The above scatter plot matrix suggests a relationship between engine size and highway mpg and engine size and highway mpg. As the engine size goes up the mpg, for city and highway, goes down.

ENSURE THIS DOCUMENT IS NEAT AND ADD IT TO YOUR PORTFOLIO

THIS REPORT WAS WRITTEN BY : Michael Sullivan