Data Analysis Challenge

INTRODUCTION

Data Analysis Challenge

An analysis of the data provided by Skillhives was performed and this report consists of all the steps taken and the findings.

Aim

The aim is to inspect, clean, transform and analyze data with the goal of discovering useful information to smoothen decision-making process.

Steps

- Data Summary Generated
- Data Quality Testing
- Data Cleansing
- Data Analysis
- Summarizing Conclusions
- Defining Future Scope

SUBMITTED BY-
SHUBHAM THAKUR

HOW TO CONTACT ME

shubhamthakur2021@dbe-du.org
shubhamrgtu@gmail.com

+91 7879677879
+91 8962832989

https://www.linkedin.com/in/shubham-thakur-mbadbe/

https://github.com/Shubham-Thakur-India/Data-Analysis-Challenge

Data Summary(raw_data):

```
In [3]: raw_data.head()
```

Out[3]:

	symboling	normalized-losses	make	fuel-type	aspiration	num-of-doors	body-style	drive-wheels	engine-location	wheel-base	...	engine-size	fuel-system	bore	stroke	compression-ratio	horsepower
0	3	?	alfa-romero	gas	std	two	convertible	rwd	front	88.6	...	130	mpfi	3.47	2.68	9.0	111
1	3	?	alfa-romero	gas	std	two	convertible	rwd	front	88.6	...	130	mpfi	3.47	2.68	9.0	111
2	1	?	alfa-romero	gas	std	two	hatchback	rwd	front	94.5	...	152	mpfi	2.68	3.47	9.0	154
3	2	164	audi	gas	std	four	sedan	fwd	front	99.8	...	109	mpfi	3.19	3.4	10.0	102
4	2	164	audi	gas	std	four	sedan	4wd	front	99.4	...	136	mpfi	3.19	3.4	8.0	115

5 rows × 26 columns

```
In [4]: print('The dataset is a cross sectional data with ',raw_data.shape[0],'rows and ',raw_data.shape[1],'columns.')
        print('Columns are: \n',list(raw_data.columns))

The dataset is a cross sectional data with  205 rows and  26 columns.
Columns 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']
```

Fig.1 Data Test

- Type of Data: Cross Sectional Data
- No. of records (Rows): 205
- Variables (Columns): 26
  - 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

Variable Format will be shown after data cleansing.
Further summary will be provided after data cleansing.

Data Quality Test:

- There is no null values (Empty Cells)(Fig-2)
- Few of the variable which should be in numerical format have string format due to entry "?". This can be deducted from Fig-3.1, 3.2 and 3.3.

```
In [5]: Variable_with_null_variables=[]
        for i in raw_data.isnull().sum():
            if i>0:
                Variable_with_null_variables.append(i)
        Variable_with_null_variables

Out[5]: []
```

Fig. 2 Null Value Test

```
raw_data.describe()
```

	symboling	wheel-base	length	width	height	curb-weight	engine-size	compression-ratio	city-mpg	highway-mpg
count	205.000000	205.000000	205.000000	205.000000	205.000000	205.000000	205.000000	205.000000	205.000000	205.000000
mean	0.834146	98.756585	174.049268	65.907805	53.724878	2555.565854	126.907317	10.142537	25.219512	30.751220
std	1.245307	6.021776	12.337289	2.145204	2.443522	520.680204	41.642693	3.972040	6.542142	6.886443
min	-2.000000	86.600000	141.100000	60.300000	47.800000	1488.000000	61.000000	7.000000	13.000000	16.000000
25%	0.000000	94.500000	166.300000	64.100000	52.000000	2145.000000	97.000000	8.600000	19.000000	25.000000
50%	1.000000	97.000000	173.200000	65.500000	54.100000	2414.000000	120.000000	9.000000	24.000000	30.000000
75%	2.000000	102.400000	183.100000	66.900000	55.500000	2935.000000	141.000000	9.400000	30.000000	34.000000
max	3.000000	120.900000	208.100000	72.300000	59.800000	4066.000000	326.000000	23.000000	49.000000	54.000000

Fig. 3.1 Summary

Out[8]:

	Variable	Type
0	symboling	int
1	normalized-losses	str
2	make	str
3	fuel-type	str
4	aspiration	str
5	num-of-doors	str
6	body-style	str
7	drive-wheels	str
8	engine-location	str
9	wheel-base	float
10	length	float
11	width	float
12	height	float
13	curb-weight	int
14	engine-type	str
15	num-of-cylinders	str
16	engine-size	int
17	fuel-system	str
18	bore	str
19	stroke	str
20	compression-ratio	float
21	horsepower	str
22	peak-rpm	str
23	city-mpg	int
24	highway-mpg	int
25	price	str

Fig. 3.2 Variable Format

```
raw_data.describe(include='all').loc['count':'freq',].T.dropna()
```

	count	unique	top	freq
normalized-losses	205	52	?	41
make	205	22	toyota	32
fuel-type	205	2	gas	185
aspiration	205	2	std	168
num-of-doors	205	3	four	114
body-style	205	5	sedan	96
drive-wheels	205	3	fwd	120
engine-location	205	2	front	202
engine-type	205	7	ohc	148
num-of-cylinders	205	7	four	159
fuel-system	205	8	mpfi	94
bore	205	39	3.62	23
stroke	205	37	3.4	20
horsepower	205	60	68	19
peak-rpm	205	24	5500	37
price	205	187	?	4

Fig. 3.3 Summary ("?" in data)

Data Cleansing:

- Column "normalized-losses" is dropped because it has 41/205 entries as "?". Which can not be ignored. (Fig. 4)
- Rows containing entries as "?" will be removed. (12 Entries have been removed)
- Format of following columns are corrected:
    - bore
    - stroke
    - horsepower
    - peak-rpm
    - price

```
(raw_data=='?').sum()
```

```
symboling           0
normalized-losses    41
make                 0
fuel-type            0
aspiration           0
num-of-doors         2
body-style           0
drive-wheels         0
engine-location      0
wheel-base           0
length               0
width                0
height               0
curb-weight          0
engine-type          0
num-of-cylinders     0
engine-size          0
fuel-system          0
bore                 4
stroke               4
compression-ratio    0
horsepower           2
peak-rpm             2
city-mpg             0
highway-mpg          0
price                4
dtype: int64
```

Fig. 4 Null Value Test

Variable Formats and Count after cleaning data:

	Variable	Type_raw	Type_cleaned
0	symboling	int	int
1	normalized-losses	str	variable_dropped
2	make	str	str
3	fuel-type	str	str
4	aspiration	str	str
5	num-of-doors	str	int
6	body-style	str	str
7	drive-wheels	str	str
8	engine-location	str	str
9	wheel-base	float	float
10	length	float	float
11	width	float	float
12	height	float	float
13	curb-weight	int	int
14	engine-type	str	str
15	num-of-cylinders	str	str
16	engine-size	int	int
17	fuel-system	str	str
18	bore	str	float
19	stroke	str	float
20	compression-ratio	float	float
21	horsepower	str	float
22	peak-rpm	str	float
23	city-mpg	int	int
24	highway-mpg	int	int
25	price	str	float

Fig. 5 Data variable Formats

```
symboling           193
make                193
fuel-type           193
aspiration          193
num-of-doors        193
body-style          193
drive-wheels        193
engine-location     193
wheel-base          193
length              193
width               193
height              193
curb-weight         193
engine-type         193
num-of-cylinders    193
engine-size         193
fuel-system         193
bore                193
stroke              193
compression-ratio   193
horsepower          193
peak-rpm            193
city-mpg            193
highway-mpg         193
price               193
Name: count, dtype: object
```

Fig. 5 Data variable counts

Data Analysis:

General description:

	count	mean	std	min	25%	50%	75%	max
wheel-base	193.0	98.923834	6.152409	86.60	94.50	97.00	102.40	120.90
length	193.0	174.326425	12.478593	141.10	166.30	173.20	184.60	208.10
width	193.0	65.893782	2.137795	60.30	64.10	65.40	66.90	72.00
height	193.0	53.869948	2.394770	47.80	52.00	54.10	55.70	59.80
curb-weight	193.0	2561.507772	526.700026	1488.00	2145.00	2414.00	2952.00	4066.00
engine-size	193.0	128.124352	41.590452	61.00	98.00	120.00	146.00	326.00
bore	193.0	3.330622	0.272385	2.54	3.15	3.31	3.59	3.94
stroke	193.0	3.248860	0.315421	2.07	3.11	3.29	3.41	4.17
compression-ratio	193.0	10.143627	3.977491	7.00	8.50	9.00	9.40	23.00
horsepower	193.0	103.481865	37.960107	48.00	70.00	95.00	116.00	262.00
peak-rpm	193.0	5099.740933	468.694369	4150.00	4800.00	5100.00	5500.00	6600.00
city-mpg	193.0	25.326425	6.387828	13.00	19.00	25.00	30.00	49.00
highway-mpg	193.0	30.787565	6.816910	16.00	25.00	30.00	34.00	54.00
price	193.0	13285.025907	8089.082886	5118.00	7738.00	10245.00	16515.00	45400.00

Fig. 6 Quantitative Variables

	count	unique	top	freq
symboling	193	6	0	63
make	193	21	toyota	32
fuel-type	193	2	gas	174
aspiration	193	2	std	158
num-of-doors	193	2	four	112
body-style	193	5	sedan	92
drive-wheels	193	3	fwd	114
engine-location	193	2	front	190
engine-type	193	5	ohc	141
num-of-cylinders	193	6	four	153
fuel-system	193	7	mpfi	88

Fig. 7 Categorical Variables

```
## Checking variable details of Categorical Variables:

categorical_variables=list(cleaned_data.describe(include='all').loc['count':'freq',].T.dropna().T.columns)
quantitave_variables=list(cleaned_data.describe().columns)
quantitave_variables

for i in categorical_variables:
    print(i,list(set(cleaned_data[i])))
```

```
symboling ['-2', '3', '2', '0', '-1', '1']
make ['bmw', 'isuzu', 'mercury', 'volvo', 'audi', 'peugot', 'saab', 'volkswagen', 'dodge', 'porsche', 'nissan', 'subaru', 'h
onda', 'plymouth', 'alfa-romero', 'mazda', 'jaguar', 'toyota', 'mercedes-benz', 'chevrolet', 'mitsubishi']
fuel-type ['gas', 'diesel']
aspiration ['turbo', 'std']
num-of-doors ['two', 'four']
body-style ['sedan', 'wagon', 'hardtop', 'hatchback', 'convertible']
drive-wheels ['rwd', '4wd', 'fwd']
engine-location ['front', 'rear']
engine-type ['dohc', 'ohcf', 'ohc', 'ohcv', 'l']
num-of-cylinders ['three', 'eight', 'six', 'five', 'four', 'twelve']
fuel-system ['spdi', 'spfi', 'mpfi', '2bbl', 'idi', 'mfi', '1bbl']
```

Fig. 7 Categorical Variable categories

Observation:

- Top occurrences and frequency are shown in Fig.7.
- Gas fueled vehicle are vastly available (90%). Diesel type are hardly there(10%).
- Most vehicle have aspiration='std'.(81.86%), while only 18.14% are 'turbo'.
- Almost all vehicles have engine in front (98.44%).
- Engine type 'ehc' is vastly available (73%). Breakup of rest needs analysis.
- Vehicle with 4 cylinders is vastly available (73%). Breakup of rest needs analysis.
- Rest categories needs further analysis.

Quantitative Data analysis:

- Pairplot is made to check the relationship overview. It is best for intial check.
- Once any pattern is seen, We will explore the category further.

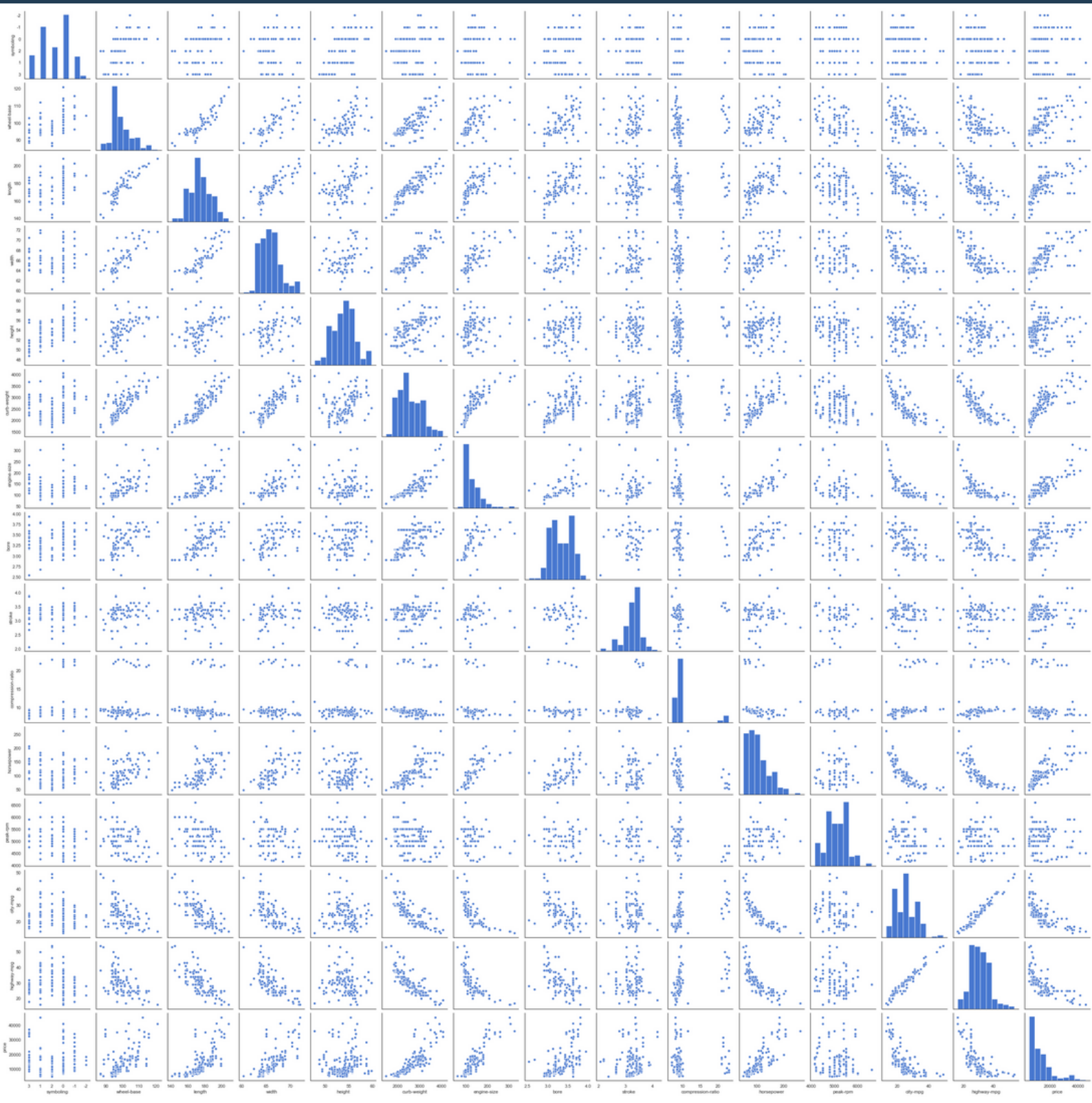

Fig. 8 Pairplot

Major Observation from Pairplot:

- There seems to be a positive linear relationship between 'city-mpg' and 'highway-mpg'.
- Most vehicle have compression ratio less than 10.
- 'Wheelbase' seems to have linear positive relationship with 'length','wheelbase','curb-weight','width'.
- 'Wheelbase' seems to have linear positive relationship with 'length' and 'width'.
- 'Wheelbase' seems to have negative relationship with 'city-mpg' and 'highway-mpg'.
- 'Price' seems to have negative relationship with 'city-mpg' and 'highway-mpg'.

#lets plot heatmap for better understanding

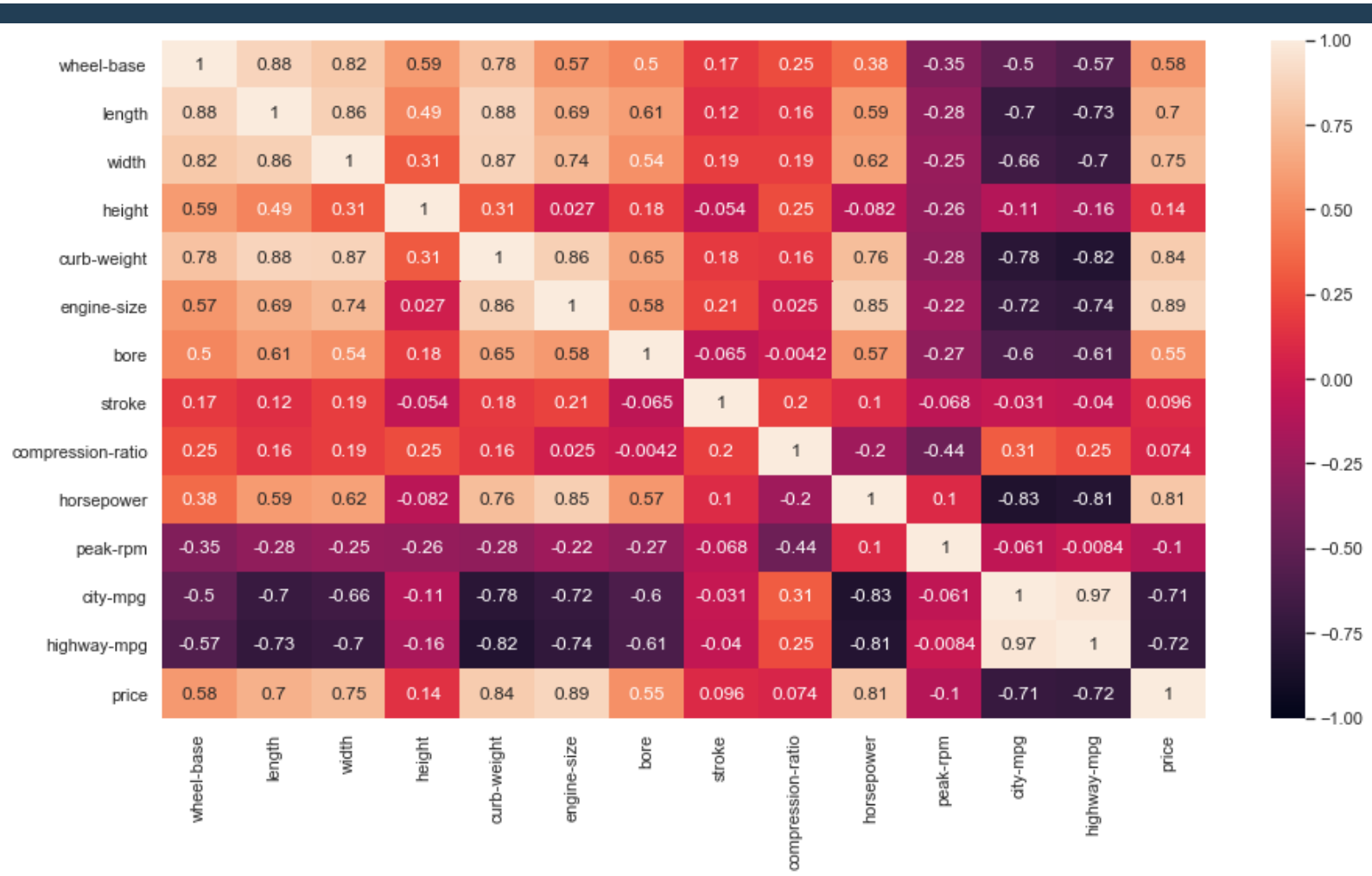

Fig. 9 Correlation Heatmap

Major Observation from Correlation Heatmap:

MPG

- The vehicle with more city-mpg will have more highway mpg and vice versa. Lets call it mpg together.
- Incresing any one parameter among length, width, curb-weight, engine-size, bore and horsepower will greatly reduce MPG.
- Higher the wheelbase, lower the MPG.
- Higher the compression ratio, higher the MPG.
- Higher MPG vehicles are cheaper.

Price-

- Price increases with increase in 'length','wheelbase','curb-weight','width','engine-size', 'bore' and 'horsepower'.
- Costly vehicle give less average.
- rpm,stroke and comression ratio doesn't affect price much.

Frequency Charts:

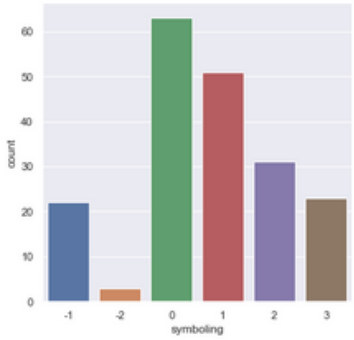

Fig. 10 Symbolling

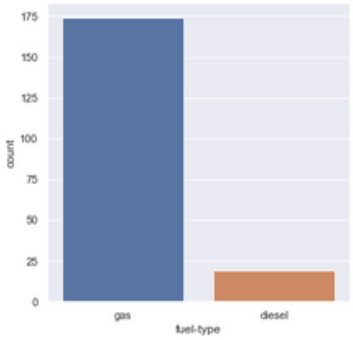

Fig.11 fuel-type

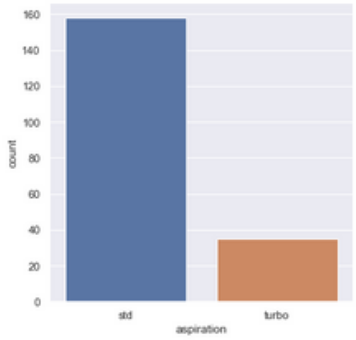

Fig.12 aspiration

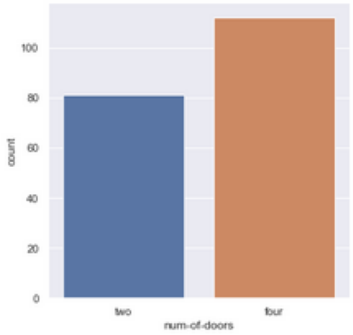

Fig. 13 num-of-doors

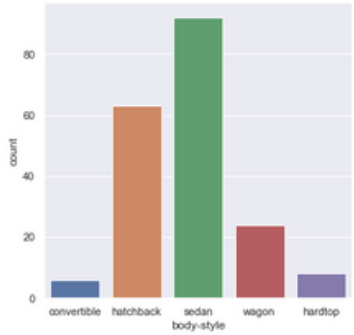

Fig.14 body-style

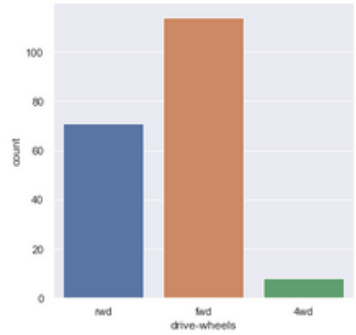

Fig.15 drive-wheels

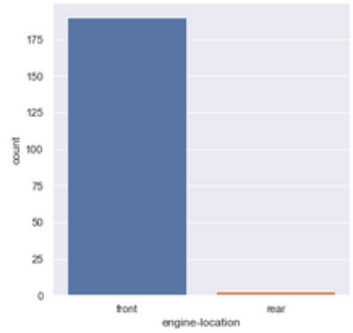

Fig.16 engine-location

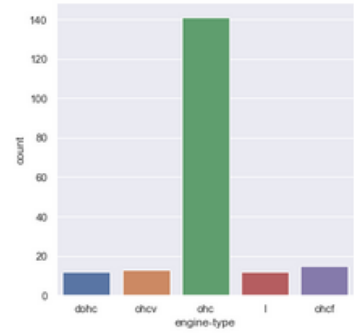

Fig.17 engine-type

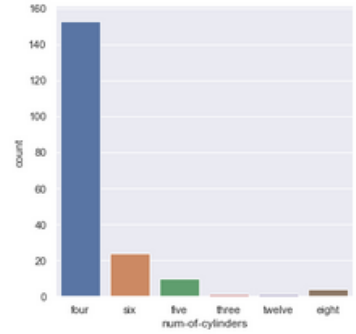

Fig.18 num-of-cylinders

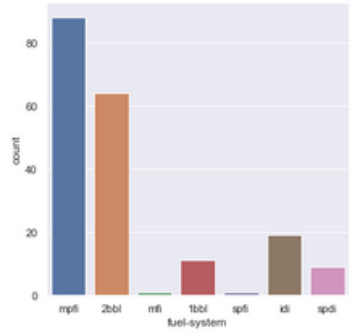

Fig. 19 fuel-system

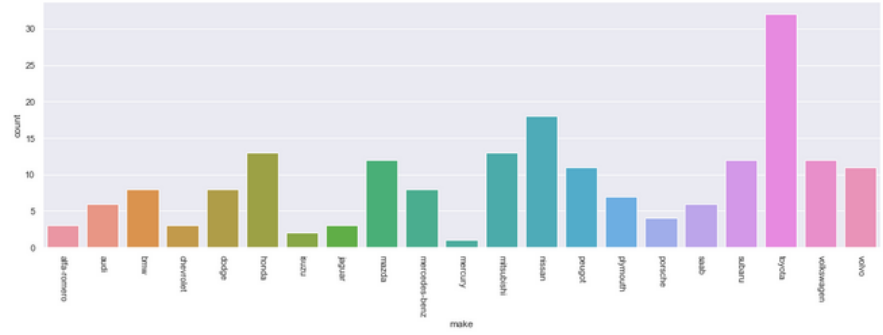

Fig.20 Make

Major Observation from Frequency charts:

- Most vehicle are of symbol = 0 followed by symbol = 1, While the safest vehicle with symbol=-2 are least.
  - We can say that safety in most of the vehicle is average.
- Vehicle with gas type fuel is prevalent.
- Aspiration type std is prevalent.
- Almost all vehicles have an engine in front.
- Ohc engine types are prevalent while rest categories are nearly equal.
- Four-cylinder Vehicles are prominent. while 12 & 3 cylinder vehicles are barely there.
- Most vehicles are fwd wheels driven followed by bwd. 4wd wheels driven vehicle are rarely available.
- Mpfi vehicle is mostly available with 2bbl following suit. Rest lacks behind by a lot with mfi & spfi almost nonexistent.
- The most common maker is Toyota with Nissan in the second position. Mercury vehicles are hardly there.
- Four-door vehicles are more than two-door vehicles.

We will not be doing in depth analysis for 'fuel-type', 'aspiration', 'engine-location', 'num-of-cylinders' , 'engine-type' as the number of observation are too less for few categories in this variables and this might result into biased conclusion. We will check their effect on price in general.

Variable Analysis: Boxplots

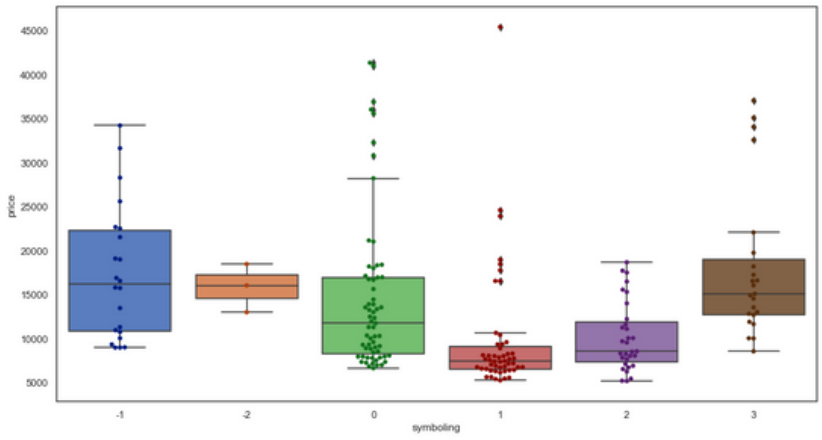

Fig. 21
Boxplot with data points
Price- Symboling

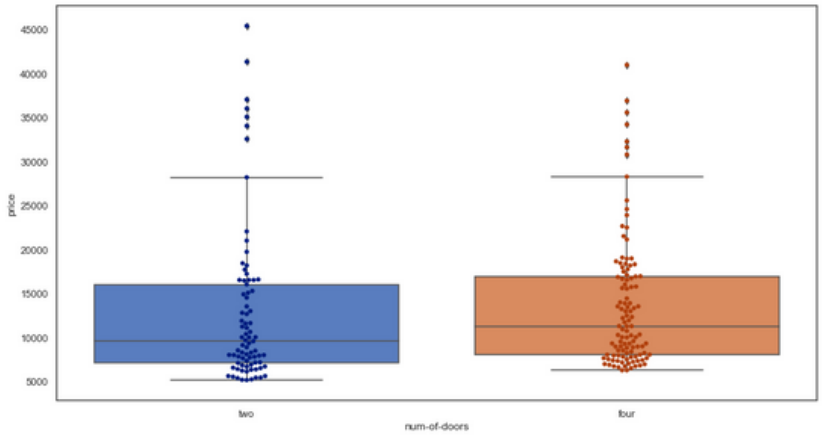

Fig. 22
Boxplot with data points
Price- num of doors

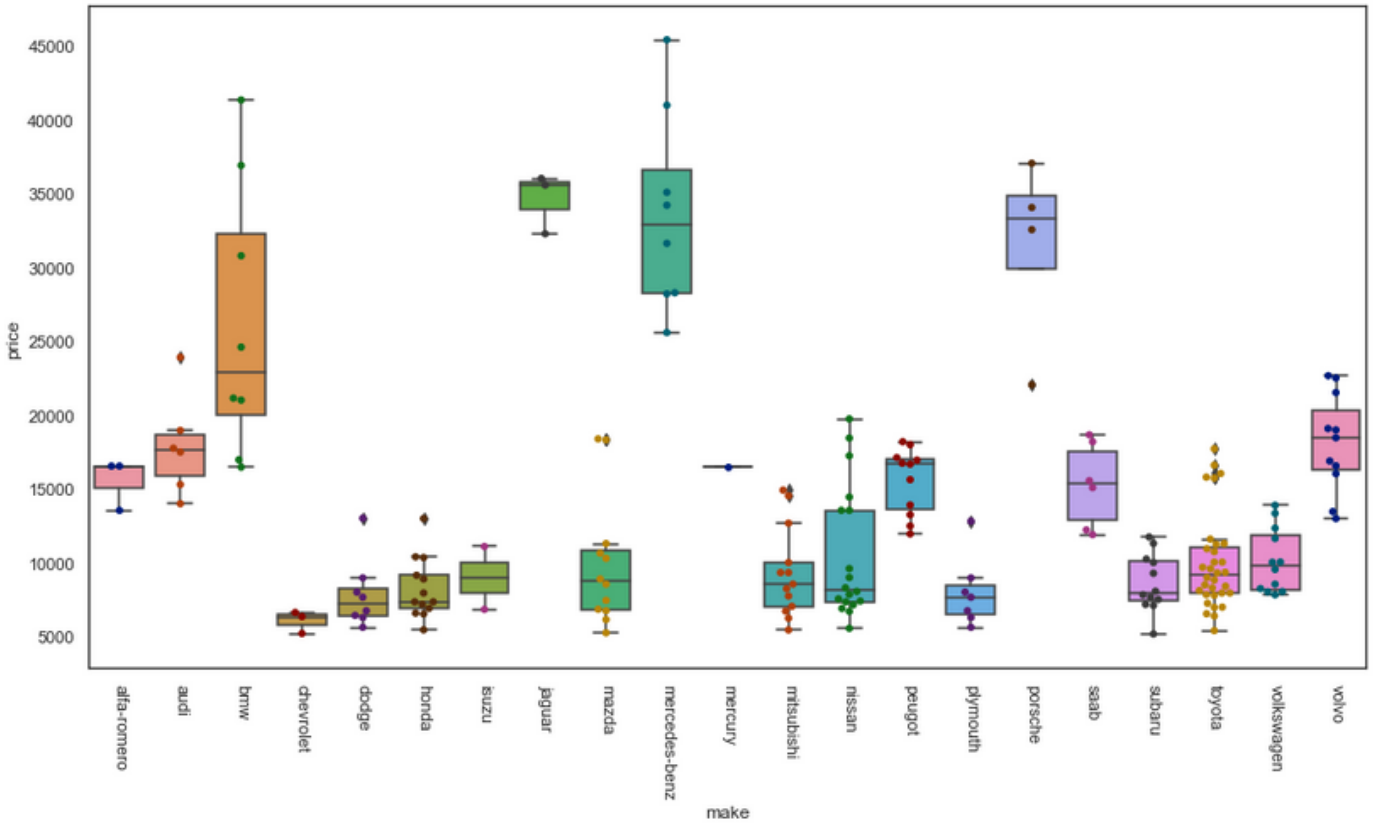

Fig. 23 Boxplot with data points: Price-Make

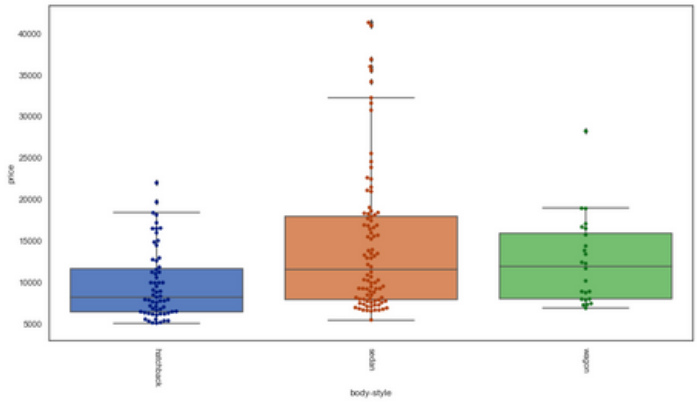

Fig. 24 Boxplot with data points:
Price-'body-style' (excluding convertible and hardtop)

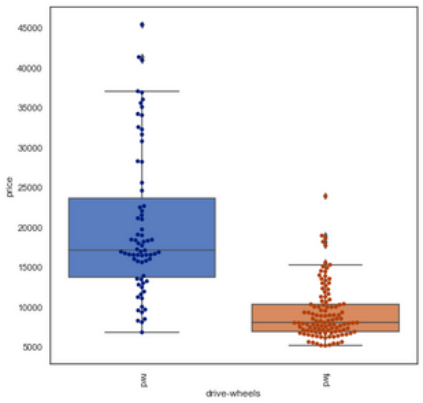

Fig. 25 Boxplot with data points:
Price-'drive-wheels' (excluding 4wd)

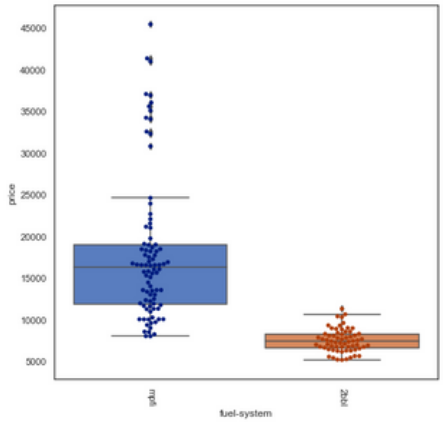

Fig. 26 Boxplot with data points:
Price-'fuel-system' (only 2bbl and mfsi are considered)

Major Observation from box plots:

- Vehicle with symboling=-1 are varied evenly across range- 11000 to 23000. (Fig.21)
- Vehicle with symboling=1 are cheaper than any other vehicles. With few exceptions.
- Number of doors seems to have no effect on price. (Fig.21)
- Sedan has higher range of price. In all categories majority of vehichles are in lower price segment with similar price, less models are available as we increase price. (Fig.24)
- Fwd vehicles drive wheels costs less and have lower price range while rws drive wheels vehicles costs more and have higher price range. (Fig.25)
- mpfi has higher price and have wide range of price while 2bbl is cheap and saturated around 7500 (not even many outlier lie above the upper range and none is costly). (Fig.26)
- mercedes-benz has the costliest vehicles and tends to medium upper range of price. jaguar only tends to higher price range.

Conclusion:

1. Gas fueled vehicle are vastly available (90%).
2. Diesel type is hardly there(10%).
3. Vehicles with std' aspiration are vastly available(81.86%), while only 18.14% are 'turbo'.
4. 'std' type aspiration vehicles are vastly available.
5. Almost all vehicles have an engine in front (98.44%).
6. Engine type 'ehc' is vastly available (73%).
7. A vehicle with 4 cylinders is vastly available (73%).Most vehicles have a compression ratio of less than 10.
8. Increasing 'Wheelbase' Increases 'length','curb-weight','width' of vehicle.
9. Increasing 'Wheelbase' decreases 'city-mpg' and 'highway-mpg'.
10. Costly vehicles have low 'city-mpg' and 'highway-mpg'. Good mpg vehicles are less costly.
11. The vehicle with more city-mpg will have more highway mpg and vice versa. Let's call it mpg together.
12. Increasing any one parameter among length, width, curb-weight, engine-size, bore, and horsepower will greatly reduce MPG.
13. The higher the wheelbase, the lower the MPG.
14. Higher the compression ratio, the higher the MPG.
15. Higher MPG vehicles are cheaper.
16. Price increases with increase in 'length', 'wheelbase', 'curb-weight', 'width', 'engine-size', 'bore' and 'horsepower'.
17. Costly vehicles give less average.
18. rpm, stroke, and compression ratio doesn't affect the price much.
19. Most vehicles are of symbol = 0 followed by symbol = 1, While the safest vehicles with symbol=-2 are less in number.
20. We can say that safety in most of the vehicle is average.
21. A vehicle with gas type fuel is prevalent.
22. Aspiration type std is prevalent.
23. Almost all vehicles have an engine in front.
24. Ohc engine types are prevalent while the rest categories are nearly equal.
25. Four-cylinder Vehicles are prominent. while 12 & 3 cylinder vehicles are barely there.
26. Most vehicles are fwd wheels driven followed by bwd. 4wd wheels driven vehicles are rarely available.
27. Mpfi vehicle is mostly available with 2bbl following suit. Rest lacks behind by a lot with mfi & spfi almost nonexistent.
28. The most common maker is Toyota with Nissan in the second position. Mercury vehicles are hardly there.
29. Four-door vehicles are more than two-door vehicles.
30. Vehicles with symboling=-1 are varied evenly across the range- 11000 to 23000. (Fig.21)Vehicles with symboling=1 are cheaper than any other vehicles. With few exceptions.
31. The number of doors seems to not affect the price. (Fig.21)
32. A sedan has a higher range of price. In all categories majority of vehicles are in the lower price segment with a similar price, fewer models are available as we increase the price. (Fig.24)
33. Fwd vehicles drive wheels cost less and have a lower price range while rws drive wheels vehicles cost more and have a higher price range. (Fig.25)
34. 'mpfi' vehicles have a higher price and have a wide range of price while 2bbl is cheap and saturated around 7500. (Fig.26)
35. Mercedes-Benz has the costliest vehicles and tends to medium upper range of price. jaguar only tends to higher price range.

Future Scope:

Since the aim of the competition was limited to analyze the information available, no modeling was done.

We can perform regression modelling on price and predict price based on other variable. Dropping highly correlated variables would be the step ahead. Removing multicollinearity and Standardizing the data. Creating dummy variables checking effect of each variable and keeping only effective dummy variables. The process goes on.