Vathana Him and Andrew Lindberg

Automobile Dataset

This dataset contains information about specific type of vehicles and their specs. Based on the head and tails visualization, this data set is a dirty data set. There are some values that contains improper formatting and some non-standardized values. For example column['normalized-loss'] contains '?', which is not uniform with the rest of the dataset. Subsequently, column['engine-location'] also contains non-uniform formatting of data value. Additionally, when looking at the entire dataframe, columns [num-of-doors],['bore'],['stroke'],['horsepower'], ['peak-rpm'], and ['price'] also have a few missing values with the '?' data point.

In [3]:
```
#libraries
import os #operating system
import numpy as np # linear algebra
import pandas as pd #data processing, CSV file I/O (e.g. pd.read_csv)
import matplotlib.pyplot as plt #standard graphics
import seaborn as sns #fancier graphics
from scipy import stats
from sklearn import preprocessing
from functools import reduce
```

In [4]:
```
L = [1,2,3,4,5]

#lambda-calculus (google this) is another way of defining functions
#f is a function that takes two arguments, x and y,
f = lambda x,y: x+y

#reduce allows us to add the values of L simply
reduce(f,L)
```

Out[4]: 15

In [32]:
```
data = pd.read_csv('/Users/vathanahim/Documents/UNCG-FALL2020/IAF603/DataSet/Automobile_data.csv')
pd.set_option('display.max_columns',26)
pd.set_option('display.max_rows', 400)
```

In [33]:
```
#show the first 5 rows
data.head()
```

Out[33]:

	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
0	3	?	alfa-romero	gas	std	two	convertible	rwd	front[front]	88.6	168.8	64.1	48.8	2548	dohc
1	3	?	alfa-romero	gas	std	two	convertible	rwd	front?	88.6	168.8	64.1	48.8	2548	dohc
2	1	?	alfa-romero	gas	std	two	hatchback	rwd	front?	94.5	171.2	65.5	52.4	2823	ohcv
3	2	164	audi	gas	std	four	sedan	fwd	front,	99.8	176.6	66.2	54.3	2337	ohc
4	2	164	audi	gas	std	four	sedan	4wd	front,front	99.4	176.6	66.4	54.3	2824	ohc

In [34]:
```
#show the last 5 rows
data.tail()
```

Out[34]:

	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
200	-1	95	volvo	gas	std	four	sedan	rwd	front	109.1	188.8	68.9	55.5	2952	ohc
201	-1	95	volvo	gas	turbo	four	sedan	rwd	front	109.1	188.8	68.8	55.5	3049	ohc
202	-1	95	volvo	gas	std	four	sedan	rwd	front	109.1	188.8	68.9	55.5	3012	ohcv
203	-1	95	volvo	diesel	turbo	four	sedan	rwd	front	109.1	188.8	68.9	55.5	3217	ohc
204	-1	95	volvo	gas	turbo	four	sedan	rwd	front	109.1	188.8	68.9	55.5	3062	ohc

In [35]:
```
size = data.size
print(size)
data.shape
```

```
5330
```

Out[35]: (205, 26)

This data frame has a total of 5,330 data points. The dimension of this dataframe is 205 rows by 26 columns.

In [36]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 205 entries, 0 to 204
Data columns (total 26 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   symboling          205 non-null    int64
 1   normalized-losses  205 non-null    object
 2   make               205 non-null    object
 3   fuel-type          205 non-null    object
 4   aspiration         205 non-null    object
 5   num-of-doors       205 non-null    object
 6   body-style         205 non-null    object
 7   drive-wheels       205 non-null    object
 8   engine-location    205 non-null    object
 9   wheel-base         205 non-null    float64
 10  length             205 non-null    float64
 11  width              205 non-null    float64
 12  height             205 non-null    float64
 13  curb-weight        205 non-null    int64
 14  engine-type        205 non-null    object
 15  num-of-cylinders   205 non-null    object
 16  engine-size        205 non-null    int64
 17  fuel-system        205 non-null    object
 18  bore               205 non-null    object
 19  stroke             205 non-null    object
 20  compression-ratio  205 non-null    float64
 21  horsepower         205 non-null    object
 22  peak-rpm           205 non-null    object
 23  city-mpg           205 non-null    int64
 24  highway-mpg        205 non-null    int64
 25  price              205 non-null    object
dtypes: float64(5), int64(5), object(16)
memory usage: 41.8+ KB
```

In [37]:
```
data.isnull().sum()
```

Out[37]:
```
symboling            0
normalized-losses    0
make                 0
fuel-type            0
aspiration           0
num-of-doors         0
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                 0
stroke               0
compression-ratio    0
horsepower           0
peak-rpm             0
city-mpg             0
highway-mpg          0
price                0
dtype: int64
```

There are no null values in this specific dataset; however, review of the data head and tail show question marks (?) to indicate a missing information.

In [38]:
```
c = ['?']
data.isin(c).sum()
```

Out[38]:
```
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
```

In [39]:
```
sum(data.isin(c).sum())
```

Out[39]: 59

The table above shows the missing values per column. The most egregious offender is normalized-losses with 41 missing values, and a total of 59 missing values in the data set, indicated by '?'.

In [40]:
```
#Descriptive Statistics
data.describe().T
```

Out[40]:

	count	mean	std	min	25%	50%	75%	max
symboling	205.0	0.834146	1.245307	-2.0	0.0	1.0	2.0	3.0
wheel-base	205.0	98.756585	6.021776	86.6	94.5	97.0	102.4	120.9
length	205.0	174.049268	12.337289	141.1	166.3	173.2	183.1	208.1
width	205.0	65.907805	2.145204	60.3	64.1	65.5	66.9	72.3
height	205.0	53.724878	2.443522	47.8	52.0	54.1	55.5	59.8
curb-weight	205.0	2555.565854	520.680204	1488.0	2145.0	2414.0	2935.0	4066.0
engine-size	205.0	126.907317	41.642693	61.0	97.0	120.0	141.0	326.0
compression-ratio	205.0	10.142537	3.972040	7.0	8.6	9.0	9.4	23.0
city-mpg	205.0	25.219512	6.542142	13.0	19.0	24.0	30.0	49.0
highway-mpg	205.0	30.751220	6.886443	16.0	25.0	30.0	34.0	54.0

In [41]:
```
data['symboling'].value_counts()
```

Out[41]:
```
 0    67
 1    54
 2    32
 3    27
-1    22
-2     3
Name: symboling, dtype: int64
```

In [42]:
```
data['make'].value_counts()
```

Out[42]:
```
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
plymouth          7
audi              7
saab              6
porsche           5
isuzu             4
jaguar            3
chevrolet         3
alfa-romero       3
renault           2
mercury           1
Name: make, dtype: int64
```

In [43]:
```
data['fuel-type'].value_counts()
```

Out[43]:
```
gas       185
diesel     20
Name: fuel-type, dtype: int64
```

In [44]:
```
data['aspiration'].value_counts()
```

Out[44]:
```
std      168
turbo     37
Name: aspiration, dtype: int64
```

In [45]:
```
data['num-of-doors'].value_counts()
```

Out[45]:
```
four    114
two      89
?         2
Name: num-of-doors, dtype: int64
```

In [46]:
```
data.loc[data['num-of-doors'] == '?']
```

Out[46]:

	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
27	1	148	dodge	gas	turbo	?	sedan	fwd	front	93.7	157.3	63.8	50.6	2191	ohc
63	0	?	mazda	diesel	std	?	sedan	fwd	front	98.8	177.8	66.5	55.5	2443	ohc

In [47]:
```
is_sedan = data['body-style'] == 'sedan'
data_sedan = data[is_sedan]
data_sedan['num-of-doors'].value_counts()
```

Out[47]:
```
four    79
two     15
?        2
Name: num-of-doors, dtype: int64
```

In [48]:
```
data['body-style'].value_counts()
```

Out[48]:
```
sedan          96
hatchback      70
wagon          25
hardtop         8
convertible     6
Name: body-style, dtype: int64
```

In [49]:
```
data['drive-wheels'].value_counts()
```

Out[49]:
```
fwd    120
rwd     76
4wd      9
Name: drive-wheels, dtype: int64
```

In [50]:
```
data['engine-location'].value_counts()
```

Out[50]:
```
front               179
front,                5
front[location]       5
front[engine]         4
font                  4
rear[end]             3
front?                2
front,front           1
front[front]          1
front,                1
Name: engine-location, dtype: int64
```

In [51]:
```
data['engine-type'].value_counts()
```

Out[51]:
```
ohc      148
ohcf      15
ohcv      13
dohc      12
l         12
rotor      4
dohcv      1
Name: engine-type, dtype: int64
```

In [52]:
```
data['num-of-cylinders'].value_counts()
```

Out[52]:
```
four      134
for        25
six        24
five       11
eight       5
two         4
twelve      1
three       1
Name: num-of-cylinders, dtype: int64
```

In [53]:
```
data['fuel-system'].value_counts()
```

Out[53]:
```
mpfi    94
2bbl    66
idi     20
1bbl    11
spdi     9
4bbl     3
mfi      1
spfi     1
Name: fuel-system, dtype: int64
```

In [54]:
```
data.loc[data['bore'] == '?']
```

Out[54]:

	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
55	3	150	mazda	gas	std	two	hatchback	rwd	front	95.3	169.0	65.7	49.6	2380	rotor
56	3	150	mazda	gas	std	two	hatchback	rwd	front	95.3	169.0	65.7	49.6	2380	rotor
57	3	150	mazda	gas	std	two	hatchback	rwd	front	95.3	169.0	65.7	49.6	2385	rotor
58	3	150	mazda	gas	std	two	hatchback	rwd	front	95.3	169.0	65.7	49.6	2500	rotor

Bore is a measure of the diameter of the engine cylinder, the stroke is the a measure of the distance the traveled by a piston from top to bottom or vice versa, and a rotary engine doesn't have cylinders or pistons; instead a triangular rotor is used instead of pistons.

In [55]:
```
data.loc[data['horsepower'] == '?']
```

Out[55]:

	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
130	0	?	renault	gas	std	four	wagon	fwd	font	96.1	181.5	66.5	55.2	2579	ohc
131	2	?	renault	gas	std	two	hatchback	fwd	font	96.1	176.8	66.6	50.5	2460	ohc

In [56]:
```
data.loc[data['make'] == 'renault']
```

Out[56]:

	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
130	0	?	renault	gas	std	four	wagon	fwd	font	96.1	181.5	66.5	55.2	2579	ohc
131	2	?	renault	gas	std	two	hatchback	fwd	font	96.1	176.8	66.6	50.5	2460	ohc

In [57]:
```
data.loc[data['price'] == '?']
```

Out[57]:

	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
9	0	?	audi	gas	turbo	two	hatchback	4wd	front,	99.5	178.2	67.9	52.0	3053	ohc
44	1	?	isuzu	gas	std	two	sedan	fwd	front	94.5	155.9	63.6	52.0	1874	ohc
45	0	?	isuzu	gas	std	four	sedan	fwd	front	94.5	155.9	63.6	52.0	1909	ohc
129	1	?	porsche	gas	std	two	hatchback	rwd	font	98.4	175.7	72.3	50.5	3366	dohc

In [58]:
```
data.loc[data['make'] == 'audi']
```

Out[58]:

	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
3	2	164	audi	gas	std	four	sedan	fwd	front,	99.8	176.6	66.2	54.3	2337	ohc
4	2	164	audi	gas	std	four	sedan	4wd	front,front	99.4	176.6	66.4	54.3	2824	ohc
5	2	?	audi	gas	std	two	sedan	fwd	front,	99.8	177.3	66.3	53.1	2507	ohc
6	1	158	audi	gas	std	four	sedan	fwd	front,	105.8	192.7	71.4	55.7	2844	ohc
7	1	?	audi	gas	std	four	wagon	fwd	front,	105.8	192.7	71.4	55.7	2954	ohc
8	1	158	audi	gas	turbo	four	sedan	fwd	front,	105.8	192.7	71.4	55.9	3086	ohc
9	0	?	audi	gas	turbo	two	hatchback	4wd	front,	99.5	178.2	67.9	52.0	3053	ohc

In [59]:
```
data.loc[data['make'] == 'isuzu']
```

Out[59]:

	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
43	0	?	isuzu	gas	std	four	sedan	rwd	front	94.3	170.7	61.8	53.5	2337	ohc
44	1	?	isuzu	gas	std	two	sedan	fwd	front	94.5	155.9	63.6	52.0	1874	ohc
45	0	?	isuzu	gas	std	four	sedan	fwd	front	94.5	155.9	63.6	52.0	1909	ohc
46	2	?	isuzu	gas	std	two	hatchback	rwd	front	96.0	172.6	65.2	51.4	2734	ohc

In [60]:
```
data.loc[data['make'] == 'porsche']
```

Out[60]:

	symboling	normalized-losses	make	fuel-type	aspiration	num-of-doors	body-style	drive-wheels	engine-location	wheel-base	length	width	height	curb-weight
125	3	186	porsche	gas	std	two	hatchback	rwd	front	94.5	168.9	68.3	50.2	2778
126	3	?	porsche	gas	std	two	hardtop	rwd	rear[end]	89.5	168.9	65.0	51.6	2756
127	3	?	porsche	gas	std	two	hardtop	rwd	rear[end]	89.5	168.9	65.0	51.6	2756
128	3	?	porsche	gas	std	two	convertible	rwd	rear[end]	89.5	168.9	65.0	51.6	2800
129	1	?	porsche	gas	std	two	hatchback	rwd	font	98.4	175.7	72.3	50.5	3366

Pt 2. Data Cleaning

For "normalized-losses", "num-of-doors", "bore", "stroke", "horsepower", "peak-rpm", and "price" there are missing values because it has "?" in its place.

"Normalized-losses," "bore," "stroke," "horsepower," "peak-rpm," and "price" data need to first be converted to numeric values.

1. For number-of-doors, there are only two missing values. If it's possible to find the year these models were produced, brand, price, and engine size could be used to make an educated guess on number of doors. Without that information, we can reference the 'body-style' column, which should offer some insight as to the 'type' of car. In this data set, there are 79 four-door sedans, 15 two-door sedans, and two missing pieces of data. Since four-door is the mode for the entire data set and for sedans, replace the missing values with four-doors.
2. Normalized-losses is a factor relative to the average loss payment per insured vehicle per year. These values are normalized for all autos within a particular size classification such as SUV, compact car, sports car, etc. There should be some relationship between insurance assigned risk factors (symboling), number of doors, and horsepower.
    - According to the Insurance Institute for Highway Safety (IIHS), *Larger, heavier vehicles generally afford more protection than smaller, lighter ones.* As such, vehicles with a higher curbweight may have lower normalized losses because passengers are less likely to need medical attention. Alternatively, a new Ford F-150, America's best-selling vehicle for 37 years, has a starting MSRP of nearly 45% more than a base model Honda Civic (30,000vs20,650). If the majority of losses come from damage to vehicles, then a lower curbweight may have lower normalized losses.
    - Given that loss data is likely to be observational, and the high possibility of confounding variables to influence losses; the best course of action would be to perform hypothesis testing for the correlation of different variables to normalized losses, and substitute the missing data with some sort of linear interpolation.
3. 'Bore' and 'stroke' are missing on vehicles with rotary engines, and this data should be deleted without replacement.
4. The missing 'horsepower' and 'peak-rpm' data are renault vehicles with an engine size of 132, four cylinders, and mpg of 23/31 (city/hwy) - one wagon and one hatchback. According to the United States Department of Energy, Renault made an 18I 4DR wagon in 1985 with an engine that has four cylinders, 132 cubic inch displacement, and 23/31 mpg (city/hwy). The Douvrin engine in that car, has a shorter piston stroke (3.5 vs. 3.9 in the dataset) and a different compression ratio (8.8 vs 8.7). The Douvrin's horsepower is somewhere between 108-135. The smaller version of the same engine has a peak-rpm of 6,000.
    - Hypothesis testing for a relationship between horsepower and naturally aspirated (std), gas engine sizes is likely to show a relationship between engine size and horsepower. If there is sufficient evidence, linear interpolation could be used for this data. otherwise, review the distribution of the data and substitute the median or mean.
    - Peak-rpm data needs to be reviewed for distribution shape. If normal, substitue for the mean, if skewed, use the median.
5. Brand data can often be used as a reference for price, especially when considering that some of the missing price data is for luxury imported vehicles such as Audi and Porsche. If the median price for each of these brands is in the upper quartile of prices, then the median price for the brand should be substituted for the missing data. Check to ensure that Isuzu is less of an outlier, and if it is, use the median price for these missing data points.

For "engine-location" the formatting of the enginge location should be either "front" or "rear" and will replace the incorrect formatting in the data frame.

For "engine-size" the outliers, which is considered to be greater than or less than Q1-1.5IQR and Q3+1.5IQR of its distribution will be replaced with the mean value.

Based on the analysis above we can determine that columns "normalized-losses", "engine-location", and "engine-size" contains dirty data because each has missing values, typos, and outliers respectively, and will targeted for cleaning.

In [61]:
```python
targeted_column_list = ["normalized-losses","engine-location","engine-size"]
df_set = data[targeted_column_list]
df_set
```

Out[61]:

	normalized-losses	engine-location	engine-size
0	?	front[front]	130
1	?	front?	130
2	?	front?	152
3	164	front,	109
4	164	front,front	136
5	?	front,	136
6	158	front,	136
7	?	front,	136
8	158	front,	131
9	?	front,	131
10	192	front[engine]	108
11	192	front[engine]	108
12	188	front[engine]	164
13	188	front[engine]	164
14	?	front[location]	164
15	?	front[location]	209
16	?	front[location]	209
17	?	front[location]	209
18	121	front[location]	61
19	98	front	90
20	81	front	90
21	118	front	90
22	118	front	90
23	118	front	98
24	148	front	90
25	148	front	90
26	148	front	90
27	148	front	98
28	110	front	122
29	145	front	156
30	137	front	92
31	137	front	92
32	101	front	79
33	101	front	92
34	101	front	92
35	110	front	92
36	78	front	92
37	106	front	110
38	106	front	110
39	85	front	110
40	85	front	110
41	85	front	110
42	107	front	110
43	?	front	111
44	?	front	90
45	?	front	90
46	?	front	119
47	145	front	258
48	?	front	258
49	?	front	326
50	104	front	91
51	104	front	91
52	104	front	91
53	113	front	91
54	113	front	91
55	150	front	70
56	150	front	70
57	150	front	70
58	150	front	80
59	129	front	122
60	115	front	122
61	129	front	122
62	115	front	122
63	?	front	122
64	115	front	122
65	118	front	140
66	?	front	134
67	93	front	183
68	93	front	183
69	93	front	183
70	93	front	183
71	?	front	234
72	142	front	234
73	?	front	308
74	?	front	304
75	?	front	140
76	161	front	92
77	161	front	92
78	161	front	92
79	161	front	98
80	153	front	110
81	153	front	122
82	?	front	156
83	?	front	156
84	?	front	156
85	125	front	122
86	125	front	122
87	125	front	110
88	137	front	110
89	128	front	97
90	128	front	103
91	128	front	97
92	122	front	97
93	103	front	97
94	128	front	97
95	128	front	97
96	122	front	97
97	103	front	97
98	168	front	97
99	106	front	120
100	106	front	120
101	128	front	181
102	108	front	181
103	108	front	181
104	194	front	181
105	194	front	181
106	231	front	181
107	161	front	120
108	161	front	152
109	?	front	120
110	?	front	152
111	161	front	120
112	161	front	152
113	?	front	120
114	?	front	152
115	161	front	120
116	161	front	152
117	161	front	134
118	119	front	90
119	119	front	98
120	154	front	90
121	154	front	90
122	154	front	98
123	74	front	122
124	?	front	156
125	186	front	151
126	?	rear[end]	194
127	?	rear[end]	194
128	?	rear[end]	194
129	?	font	203
130	?	font	132
131	?	font	132
132	150	font	121
133	104	front	121
134	150	front	121
135	104	front	121
136	150	front	121
137	104	front	121
138	83	front	97
139	83	front	108
140	83	front	108
141	102	front	108
142	102	front	108
143	102	front	108
144	102	front	108
145	102	front	108
146	89	front	108
147	89	front	108
148	85	front	108
149	85	front	108
150	87	front	92
151	87	front	92
152	74	front	92
153	77	front	92
154	81	front	92
155	91	front	92
156	91	front	98
157	91	front	98
158	91	front	110
159	91	front	110
160	91	front	98
161	91	front	98
162	91	front	98
163	168	front	98
164	168	front	98
165	168	front	98
166	168	front	98
167	134	front	146
168	134	front	146
169	134	front	146
170	134	front	146
171	134	front	146
172	134	front	146
173	65	front	122
174	65	front	110
175	65	front	122
176	65	front	122
177	65	front	122
178	197	front	171
179	197	front	171
180	90	front	171
181	?	front	161
182	122	front	97
183	122	front	109
184	94	front	97
185	94	front	109
186	94	front	109
187	94	front	97
188	94	front	109
189	?	front	109
190	256	front	109
191	?	front	136
192	?	front	97
193	?	front	109
194	103	front	141
195	74	front	141
196	103	front	141
197	74	front	141
198	103	front	130
199	74	front	130
200	95	front	141
201	95	front	141
202	95	front	173
203	95	front	145
204	95	front	141

In [62]:
```python
summary = df_set["engine-size"].describe()
summary
```

Out[62]:
```
count    205.000000
mean     126.907317
std       41.642693
min       61.000000
25%       97.000000
50%      120.000000
75%      141.000000
max      326.000000
Name: engine-size, dtype: float64
```

In [63]:
```python
#find Interquartile range so we can use to calculate for outliers
IQR = summary[6]-summary[4]
lowerQoutliers = summary[4] - (1.5*IQR)
upperQoutliers = summary[6] + (1.5*IQR)
print("IQR: "+str(IQR), "lower outlier: "+str(lowerQoutliers), "upper outlier: "+str(upperQoutliers))
```

```
IQR: 44.0 lower outlier: 31.0 upper outlier: 207.0
```

In [64]:
```python
def cleaning(df_set):
    #replace missing values of ? with mean for column normalized-losses
    mean = 0
    total = 0
    count = 0
    for i, row in df_set.iterrows():
        if row["normalized-losses"] != '?':
            total = total+int(i)
            count = count + 1
            mean = total/count

    for i, row in df_set.iterrows():
        if row["normalized-losses"] == '?':
            df_set.at[i,"normalized-losses"] = round(mean)

    #replace outliers of column "engine-size" with the mean anything greater than 75% of distribution f
or this column is outliers
    mean = 0
    total = 0
    count = 0
    for i, row in df_set.iterrows():
        if row["engine-size"] != '?':
            total = total+int(i)
            count = count + 1
            mean = total/count
    for i, row in df_set.iterrows():
        if int(row["engine-size"]) > upperQoutliers or int(row["engine-size"]) < lowerQoutliers:
            df_set.at[i,"engine-size"] = round(mean)

    #cleans up the formatting of the string in column "engine-location"
    unwanted_char = []
    L = np.array(df_set["engine-location"]).astype(np.str)
    conc = lambda x,y: x+y
    total_char = set(reduce(conc,L))

    #find targeted character that tells from what index we should remove
    for i in total_char:
        if(i=='['or i =="," or i=="?"):
            unwanted_char.append(i)

    for i, row in df_set.iterrows():
        current_str = str(row["engine-location"])
        for j in unwanted_char:
            if j in current_str:
                char_index = current_str.find(j)
                current_str = current_str[:char_index]
                df_set.at[i,"engine-location"] = current_str
    return df_set

cleaning(df_set)
```

Out[64]:

	normalized-losses	engine-location	engine-size
0	107	front	130
1	107	front	130
2	107	front	152
3	164	front	109
4	164	front	136
5	107	front	136
6	158	front	136
7	107	front	136
8	158	front	131
9	107	front	131
10	192	front	108
11	192	front	108
12	188	front	164
13	188	front	164
14	107	front	164
15	107	front	102
16	107	front	102
17	107	front	102
18	121	front	61
19	98	front	90
20	81	front	90
21	118	front	90
22	118	front	90
23	118	front	98
24	148	front	90
25	148	front	90
26	148	front	90
27	148	front	98
28	110	front	122
29	145	front	156
30	137	front	92
31	137	front	92
32	101	front	79
33	101	front	92
34	101	front	92
35	110	front	92
36	78	front	92
37	106	front	110
38	106	front	110
39	85	front	110
40	85	front	110
41	85	front	110
42	107	front	110
43	107	front	111
44	107	front	90
45	107	front	90
46	107	front	119
47	145	front	102
48	107	front	102
49	107	front	102
50	104	front	91
51	104	front	91
52	104	front	91
53	113	front	91
54	113	front	91
55	150	front	70
56	150	front	70
57	150	front	70
58	150	front	80
59	129	front	122
60	115	front	122
61	129	front	122
62	115	front	122
63	107	front	122
64	115	front	122
65	118	front	140
66	107	front	134
67	93	front	183
68	93	front	183
69	93	front	183
70	93	front	183
71	107	front	102
72	142	front	102
73	107	front	102
74	107	front	102
75	107	front	140
76	161	front	92
77	161	front	92
78	161	front	92
79	161	front	98
80	153	front	110
81	153	front	122
82	107	front	156
83	107	front	156
84	107	front	156
85	125	front	122
86	125	front	122
87	125	front	110
88	137	front	110
89	128	front	97
90	128	front	103
91	128	front	97
92	122	front	97
93	103	front	97
94	128	front	97
95	128	front	97
96	122	front	97
97	103	front	97
98	168	front	97
99	106	front	120
100	106	front	120
101	128	front	181
102	108	front	181
103	108	front	181
104	194	front	181
105	194	front	181
106	231	front	181
107	161	front	120
108	161	front	152
109	107	front	120
110	107	front	152
111	161	front	120
112	161	front	152
113	107	front	120
114	107	front	152
115	161	front	120
116	161	front	152
117	161	front	134
118	119	front	90
119	119	front	98
120	154	front	90
121	154	front	90
122	154	front	98
123	74	front	122
124	107	front	156
125	186	front	151
126	107	rear	194
127	107	rear	194
128	107	rear	194
129	107	font	203
130	107	font	132
131	107	font	132
132	150	font	121
133	104	front	121
134	150	front	121
135	104	front	121
136	150	front	121
137	104	front	121
138	83	front	97
139	83	front	108
140	83	front	108
141	102	front	108
142	102	front	108
143	102	front	108
144	102	front	108
145	102	front	108
146	89	front	108
147	89	front	108
148	85	front	108
149	85	front	108
150	87	front	92
151	87	front	92
152	74	front	92
153	77	front	92
154	81	front	92
155	91	front	92
156	91	front	98
157	91	front	98
158	91	front	110
159	91	front	110
160	91	front	98
161	91	front	98
162	91	front	98
163	168	front	98
164	168	front	98
165	168	front	98
166	168	front	98
167	134	front	146
168	134	front	146
169	134	front	146
170	134	front	146
171	134	front	146
172	134	front	146
173	65	front	122
174	65	front	110
175	65	front	122
176	65	front	122
177	65	front	122
178	197	front	171
179	197	front	171
180	90	front	171
181	107	front	161
182	122	front	97
183	122	front	109
184	94	front	97
185	94	front	109
186	94	front	109
187	94	front	97
188	94	front	109
189	107	front	109
190	256	front	109
191	107	front	136
192	107	front	97
193	107	front	109
194	103	front	141
195	74	front	141
196	103	front	141
197	74	front	141
198	103	front	130
199	74	front	130
200	95	front	141
201	95	front	141
202	95	front	173
203	95	front	145
204	95	front	141

In [66]:
```
df2 = data
cleaning(df2)

df2 = df2.replace('?', np.NaN)
df2
```

Out[66]:

	symboling	normalized-losses	make	fuel-type	aspiration	num-of-doors	body-style	drive-wheels	engine-location	wheel-base	length	width	height	curb-weight	e
0	3	107	alfa-romero	gas	std	two	convertible	rwd	front	88.6	168.8	64.1	48.8	2548	
1	3	107	alfa-romero	gas	std	two	convertible	rwd	front	88.6	168.8	64.1	48.8	2548	
2	1	107	alfa-romero	gas	std	two	hatchback	rwd	front	94.5	171.2	65.5	52.4	2823	
3	2	164	audi	gas	std	four	sedan	fwd	front	99.8	176.6	66.2	54.3	2337	
4	2	164	audi	gas	std	four	sedan	4wd	front	99.4	176.6	66.4	54.3	2824	
5	2	107	audi	gas	std	two	sedan	fwd	front	99.8	177.3	66.3	53.1	2507	
6	1	158	audi	gas	std	four	sedan	fwd	front	105.8	192.7	71.4	55.7	2844	
7	1	107	audi	gas	std	four	wagon	fwd	front	105.8	192.7	71.4	55.7	2954	
8	1	158	audi	gas	turbo	four	sedan	fwd	front	105.8	192.7	71.4	55.9	3086	
9	0	107	audi	gas	turbo	two	hatchback	4wd	front	99.5	178.2	67.9	52.0	3053	
10	2	192	bmw	gas	std	two	sedan	rwd	front	101.2	176.8	64.8	54.3	2395	
11	0	192	bmw	gas	std	four	sedan	rwd	front	101.2	176.8	64.8	54.3	2395	
12	0	188	bmw	gas	std	two	sedan	rwd	front	101.2	176.8	64.8	54.3	2710	
13	0	188	bmw	gas	std	four	sedan	rwd	front	101.2	176.8	64.8	54.3	2765	
14	1	107	bmw	gas	std	four	sedan	rwd	front	103.5	189.0	66.9	55.7	3055	
15	0	107	bmw	gas	std	four	sedan	rwd	front	103.5	189.0	66.9	55.7	3230	
16	0	107	bmw	gas	std	two	sedan	rwd	front	103.5	193.8	67.9	53.7	3380	
17	0	107	bmw	gas	std	four	sedan	rwd	front	110.0	197.0	70.9	56.3	3505	
18	2	121	chevrolet	gas	std	two	hatchback	fwd	front	88.4	141.1	60.3	53.2	1488	
19	1	98	chevrolet	gas	std	two	hatchback	fwd	front	94.5	155.9	63.6	52.0	1874	
20	0	81	chevrolet	gas	std	four	sedan	fwd	front	94.5	158.8	63.6	52.0	1909	
21	1	118	dodge	gas	std	two	hatchback	fwd	front	93.7	157.3	63.8	50.8	1876	
22	1	118	dodge	gas	std	two	hatchback	fwd	front	93.7	157.3	63.8	50.8	1876	
23	1	118	dodge	gas	turbo	two	hatchback	fwd	front	93.7	157.3	63.8	50.8	2128	
24	1	148	dodge	gas	std	four	hatchback	fwd	front	93.7	157.3	63.8	50.6	1967	
25	1	148	dodge	gas	std	four	sedan	fwd	front	93.7	157.3	63.8	50.6	1989	
26	1	148	dodge	gas	std	four	sedan	fwd	front	93.7	157.3	63.8	50.6	1989	
27	1	148	dodge	gas	turbo	NaN	sedan	fwd	front	93.7	157.3	63.8	50.6	2191	
28	-1	110	dodge	gas	std	four	wagon	fwd	front	103.3	174.6	64.6	59.8	2535	
29	3	145	dodge	gas	turbo	two	hatchback	fwd	front	95.9	173.2	66.3	50.2	2811	
30	2	137	honda	gas	std	two	hatchback	fwd	front	86.6	144.6	63.9	50.8	1713	
31	2	137	honda	gas	std	two	hatchback	fwd	front	86.6	144.6	63.9	50.8	1819	
32	1	101	honda	gas	std	two	hatchback	fwd	front	93.7	150.0	64.0	52.6	1837	
33	1	101	honda	gas	std	two	hatchback	fwd	front	93.7	150.0	64.0	52.6	1940	
34	1	101	honda	gas	std	two	hatchback	fwd	front	93.7	150.0	64.0	52.6	1956	
35	0	110	honda	gas	std	four	sedan	fwd	front	96.5	163.4	64.0	54.5	2010	
36	0	78	honda	gas	std	four	wagon	fwd	front	96.5	157.1	63.9	58.3	2024	
37	0	106	honda	gas	std	two	hatchback	fwd	front	96.5	167.5	65.2	53.3	2236	
38	0	106	honda	gas	std	two	hatchback	fwd	front	96.5	167.5	65.2	53.3	2289	
39	0	85	honda	gas	std	four	sedan	fwd	front	96.5	175.4	65.2	54.1	2304	
40	0	85	honda	gas	std	four	sedan	fwd	front	96.5	175.4	62.5	54.1	2372	
41	0	85	honda	gas	std	four	sedan	fwd	front	96.5	175.4	65.2	54.1	2465	
42	1	107	honda	gas	std	two	sedan	fwd	front	96.5	169.1	66.0	51.0	2293	
43	0	107	isuzu	gas	std	four	sedan	rwd	front	94.3	170.7	61.8	53.5	2337	
44	1	107	isuzu	gas	std	two	sedan	fwd	front	94.5	155.9	63.6	52.0	1874	
45	0	107	isuzu	gas	std	four	sedan	fwd	front	94.5	155.9	63.6	52.0	1909	
46	2	107	isuzu	gas	std	two	hatchback	rwd	front	96.0	172.6	65.2	51.4	2734	
47	0	145	jaguar	gas	std	four	sedan	rwd	front	113.0	199.6	69.6	52.8	4066	
48	0	107	jaguar	gas	std	four	sedan	rwd	front	113.0	199.6	69.6	52.8	4066	
49	0	107	jaguar	gas	std	two	sedan	rwd	front	102.0	191.7	70.6	47.8	3950	
50	1	104	mazda	gas	std	two	hatchback	fwd	front	93.1	159.1	64.2	54.1	1890	
51	1	104	mazda	gas	std	two	hatchback	fwd	front	93.1	159.1	64.2	54.1	1900	
52	1	104	mazda	gas	std	two	hatchback	fwd	front	93.1	159.1	64.2	54.1	1905	
53	1	113	mazda	gas	std	four	sedan	fwd	front	93.1	166.8	64.2	54.1	1945	
54	1	113	mazda	gas	std	four	sedan	fwd	front	93.1	166.8	64.2	54.1	1950	
55	3	150	mazda	gas	std	two	hatchback	rwd	front	95.3	169.0	65.7	49.6	2380	
56	3	150	mazda	gas	std	two	hatchback	rwd	front	95.3	169.0	65.7	49.6	2380	
57	3	150	mazda	gas	std	two	hatchback	rwd	front	95.3	169.0	65.7	49.6	2385	
58	3	150	mazda	gas	std	two	hatchback	rwd	front	95.3	169.0	65.7	49.6	2500	
59	1	129	mazda	gas	std	two	hatchback	fwd	front	98.8	177.8	66.5	53.7	2385	
60	0	115	mazda	gas	std	four	sedan	fwd	front	98.8	177.8	66.5	55.5	2410	
61	1	129	mazda	gas	std	two	hatchback	fwd	front	98.8	177.8	66.5	53.7	2385	
62	0	115	mazda	gas	std	four	sedan	fwd	front	98.8	177.8	66.5	55.5	2410	
63	0	107	mazda	diesel	std	NaN	sedan	fwd	front	98.8	177.8	66.5	55.5	2443	
64	0	115	mazda	gas	std	four	hatchback	fwd	front	98.8	177.8	66.5	55.5	2425	
65	0	118	mazda	gas	std	four	sedan	rwd	front	104.9	175.0	66.1	54.4	2670	
66	0	107	mazda	diesel	std	four	sedan	rwd	front	104.9	175.0	66.1	54.4	2700	
67	-1	93	mercedes-benz	diesel	turbo	four	sedan	rwd	front	110.0	190.9	70.3	56.5	3515	
68	-1	93	mercedes-benz	diesel	turbo	four	wagon	rwd	front	110.0	190.9	70.3	58.7	3750	
69	0	93	mercedes-benz	diesel	turbo	two	hardtop	rwd	front	106.7	187.5	70.3	54.9	3495	
70	-1	93	mercedes-benz	diesel	turbo	four	sedan	rwd	front	115.6	202.6	71.7	56.3	3770	
71	-1	107	mercedes-benz	gas	std	four	sedan	rwd	front	115.6	202.6	71.7	56.5	3740	
72	3	142	mercedes-benz	gas	std	two	convertible	rwd	front	96.6	180.3	70.5	50.8	3685	
73	0	107	mercedes-benz	gas	std	four	sedan	rwd	front	120.9	208.1	71.7	56.7	3900	
74	1	107	mercedes-benz	gas	std	two	hardtop	rwd	front	112.0	199.2	72.0	55.4	3715	
75	1	107	mercury	gas	turbo	two	hatchback	rwd	front	102.7	178.4	68.0	54.8	2910	
76	2	161	mitsubishi	gas	std	two	hatchback	fwd	front	93.7	157.3	64.4	50.8	1918	
77	2	161	mitsubishi	gas	std	two	hatchback	fwd	front	93.7	157.3	64.4	50.8	1944	
78	2	161	mitsubishi	gas	std	two	hatchback	fwd	front	93.7	157.3	64.4	50.8	2004	
79	1	161	mitsubishi	gas	turbo	two	hatchback	fwd	front	93.0	157.3	63.8	50.8	2145	
80	3	153	mitsubishi	gas	turbo	two	hatchback	fwd	front	96.3	173.0	65.4	49.4	2370	
81	3	153	mitsubishi	gas	std	two	hatchback	fwd	front	96.3	173.0	65.4	49.4	2328	
82	3	107	mitsubishi	gas	turbo	two	hatchback	fwd	front	95.9	173.2	66.3	50.2	2833	
83	3	107	mitsubishi	gas	turbo	two	hatchback	fwd	front	95.9	173.2	66.3	50.2	2921	
84	3	107	mitsubishi	gas	turbo	two	hatchback	fwd	front	95.9	173.2	66.3	50.2	2926	
85	1	125	mitsubishi	gas	std	four	sedan	fwd	front	96.3	172.4	65.4	51.6	2365	
86	1	125	mitsubishi	gas	std	four	sedan	fwd	front	96.3	172.4	65.4	51.6	2405	
87	1	125	mitsubishi	gas	turbo	four	sedan	fwd	front	96.3	172.4	65.4	51.6	2403	
88	-1	137	mitsubishi	gas	std	four	sedan	fwd	front	96.3	172.4	65.4	51.6	2403	
89	1	128	nissan	gas	std	two	sedan	fwd	front	94.5	165.3	63.8	54.5	1889	
90	1	128	nissan	diesel	std	two	sedan	fwd	front	94.5	165.3	63.8	54.5	2017	
91	1	128	nissan	gas	std	two	sedan	fwd	front	94.5	165.3	63.8	54.5	1918	
92	1	122	nissan	gas	std	four	sedan	fwd	front	94.5	165.3	63.8	54.5	1938	
93	1	103	nissan	gas	std	four	wagon	fwd	front	94.5	170.2	63.8	53.5	2024	
94	1	128	nissan	gas	std	two	sedan	fwd	front	94.5	165.3	63.8	54.5	1951	
95	1	128	nissan	gas	std	two	hatchback	fwd	front	94.5	165.6	63.8	53.3	2028	
96	1	122	nissan	gas	std	four	sedan	fwd	front	94.5	165.3	63.8	54.5	1971	
97	1	103	nissan	gas	std	four	wagon	fwd	front	94.5	170.2	63.8	53.5	2037	
98	2	168	nissan	gas	std	two	hardtop	fwd	front	95.1	162.4	63.8	53.3	2008	
99	0	106	nissan	gas	std	four	hatchback	fwd	front	97.2	173.4	65.2	54.7	2324	
100	0	106	nissan	gas	std	four	sedan	fwd	front	97.2	173.4	65.2	54.7	2302	
101	0	128	nissan	gas	std	four	sedan	fwd	front	100.4	181.7	66.5	55.1	3095	
102	0	108	nissan	gas	std	four	wagon	fwd	front	100.4	184.6	66.5	56.1	3296	
103	0	108	nissan	gas	std	four	sedan	fwd	front	100.4	184.6	66.5	55.1	3060	
104	3	194	nissan	gas	std	two	hatchback	rwd	front	91.3	170.7	67.9	49.7	3071	
105	3	194	nissan	gas	turbo	two	hatchback	rwd	front	91.3	170.7	67.9	49.7	3139	
106	1	231	nissan	gas	std	two	hatchback	rwd	front	99.2	178.5	67.9	49.7	3139	
107	0	161	peugot	gas	std	four	sedan	rwd	front	107.9	186.7	68.4	56.7	3020	
108	0	161	peugot	diesel	turbo	four	sedan	rwd	front	107.9	186.7	68.4	56.7	3197	
109	0	107	peugot	gas	std	four	wagon	rwd	front	114.2	198.9	68.4	58.7	3230	
110	0	107	peugot	diesel	turbo	four	wagon	rwd	front	114.2	198.9	68.4	58.7	3430	
111	0	161	peugot	gas	std	four	sedan	rwd	front	107.9	186.7	68.4	56.7	3075	
112	0	161	peugot	diesel	turbo	four	sedan	rwd	front	107.9	186.7	68.4	56.7	3252	
113	0	107	peugot	gas	std	four	wagon	rwd	front	114.2	198.9	68.4	56.7	3285	
114	0	107	peugot	diesel	turbo	four	wagon	rwd	front	114.2	198.9	68.4	58.7	3485	
115	0	161	peugot	gas	std	four	sedan	rwd	front	107.9	186.7	68.4	56.7	3075	
116	0	161	peugot	diesel	turbo	four	sedan	rwd	front	107.9	186.7	68.4	56.7	3252	
117	0	161	peugot	gas	turbo	four	sedan	rwd	front	108.0	186.7	68.3	56.0	3130	
118	1	119	plymouth	gas	std	two	hatchback	fwd	front	93.7	157.3	63.8	50.8	1918	
119	1	119	plymouth	gas	turbo	two	hatchback	fwd	front	93.7	157.3	63.8	50.8	2128	
120	1	154	plymouth	gas	std	four	hatchback	fwd	front	93.7	157.3	63.8	50.6	1967	
121	1	154	plymouth	gas	std	four	sedan	fwd	front	93.7	167.3	63.8	50.8	1989	
122	1	154	plymouth	gas	std	four	sedan	fwd	front	93.7	167.3	63.8	50.8	2191	
123	-1	74	plymouth	gas	std	four	wagon	fwd	front	103.3	174.6	64.6	59.8	2535	
124	3	107	plymouth	gas	turbo	two	hatchback	rwd	front	95.9	173.2	66.3	50.2	2818	
125	3	186	porsche	gas	std	two	hatchback	rwd	front	94.5	168.9	68.3	50.2	2778	
126	3	107	porsche	gas	std	two	hardtop	rwd	rear	89.5	168.9	65.0	51.6	2756	
127	3	107	porsche	gas	std	two	hardtop	rwd	rear	89.5	168.9	65.0	51.6	2756	
128	3	107	porsche	gas	std	two	convertible	rwd	rear	89.5	168.9	65.0	51.6	2800	
129	1	107	porsche	gas	std	two	hatchback	rwd	font	98.4	175.7	72.3	50.5	3366	
130	0	107	renault	gas	std	four	wagon	fwd	font	96.1	181.5	66.5	55.2	2579	
131	2	107	renault	gas	std	two	hatchback	fwd	font	96.1	176.8	66.6	50.5	2460	
132	3	150	saab	gas	std	two	hatchback	fwd	font	99.1	186.6	66.5	56.1	2658	
133	2	104	saab	gas	std	four	sedan	fwd	front	99.1	186.6	66.5	56.1	2695	
134	3	150	saab	gas	std	two	hatchback	fwd	front	99.1	186.6	66.5	56.1	2707	
135	2	104	saab	gas	std	four	sedan	fwd	front	99.1	186.6	66.5	56.1	2758	
136	3	150	saab	gas	turbo	two	hatchback	fwd	front	99.1	186.6	66.5	56.1	2808	
137	2	104	saab	gas	turbo	four	sedan	fwd	front	99.1	186.6	66.5	56.1	2847	
138	2	83	subaru	gas	std	two	hatchback	fwd	front	93.7	156.9	63.4	53.7	2050	
139	2	83	subaru	gas	std	two	hatchback	fwd	front	93.7	157.9	63.6	53.7	2120	
140	2	83	subaru	gas	std	two	hatchback	4wd	front	93.3	157.3	63.8	55.7	2240	
141	0	102	subaru	gas	std	four	sedan	fwd	front	97.2	172.0	65.4	52.5	2145	
142	0	102	subaru	gas	std	four	sedan	fwd	front	97.2	172.0	65.4	52.5	2190	
143	0	102	subaru	gas	std	four	sedan	fwd	front	97.2	172.0	65.4	52.5	2340	
144	0	102	subaru	gas	std	four	sedan	4wd	front	97.0	172.0	65.4	54.3	2385	
145	0	102	subaru	gas	turbo	four	sedan	4wd	front	97.0	172.0	65.4	54.3	2510	
146	0	89	subaru	gas	std	four	wagon	fwd	front	97.0	173.5	65.4	53.0	2290	
147	0	89	subaru	gas	std	four	wagon	fwd	front	97.0	173.5	65.4	53.0	2455	
148	0	85	subaru	gas	std	four	wagon	4wd	front	96.9	173.6	65.4	54.9	2420	
149	0	85	subaru	gas	turbo	four	wagon	4wd	front	96.9	173.6	65.4	54.9	2650	
150	1	87	toyota	gas	std	two	hatchback	fwd	front	95.7	158.7	63.6	54.5	1985	
151	1	87	toyota	gas	std	two	hatchback	fwd	front	95.7	158.7	63.6	54.5	2040	
152	1	74	toyota	gas	std	four	hatchback	fwd	front	95.7	158.7	63.6	54.5	2015	
153	0	77	toyota	gas	std	four	wagon	fwd	front	95.7	169.7	63.6	59.1	2280	
154	0	81	toyota	gas	std	four	wagon	4wd	front	95.7	169.7	63.6	59.1	2290	
155	0	91	toyota	gas	std	four	wagon	4wd	front	95.7	169.7	63.6	59.1	3110	
156	0	91	toyota	gas	std	four	sedan	fwd	front	95.7	166.3	64.4	53.0	2081	
157	0	91	toyota	gas	std	four	hatchback	fwd	front	95.7	166.3	64.4	52.8	2109	
158	0	91	toyota	diesel	std	four	sedan	fwd	front	95.7	166.3	64.4	53.0	2275	
159	0	91	toyota	diesel	std	four	hatchback	fwd	front	95.7	166.3	64.4	52.8	2275	
160	0	91	toyota	gas	std	four	sedan	fwd	front	95.7	166.3	64.4	53.0	2094	
161	0	91	toyota	gas	std	four	hatchback	fwd	front	95.7	166.3	64.4	52.8	2122	
162	0	91	toyota	gas	std	four	sedan	fwd	front	95.7	166.3	64.4	52.8	2140	
163	1	168	toyota	gas	std	two	sedan	rwd	front	94.5	168.7	64.0	52.6	2169	
164	1	168	toyota	gas	std	two	hatchback	rwd	front	94.5	168.7	64.0	52.6	2204	
165	1	168	toyota	gas	std	two	sedan	rwd	front	94.5	168.7	64.0	52.6	2265	
166	1	168	toyota	gas	std	two	hatchback	rwd	front	94.5	168.7	64.0	52.6	2300	
167	2	134	toyota	gas	std	two	hardtop	rwd	front	98.4	176.2	65.6	52.0	2540	
168	2	134	toyota	gas	std	two	hardtop	rwd	front	98.4	176.2	65.6	52.0	2536	
169	2	134	toyota	gas	std	two	hatchback	rwd	front	98.4	176.2	65.6	52.0	2551	
170	2	134	toyota	gas	std	two	hardtop	rwd	front	98.4	176.2	65.6	52.0	2679	
171	2	134	toyota	gas	std	two	hatchback	rwd	front	98.4	176.2	65.6	52.0	2714	
172	2	134	toyota	gas	std	two	convertible	rwd	front	98.4	176.2	65.6	53.0	2975	
173	-1	65	toyota	gas	std	four	sedan	fwd	front	102.4	175.6	66.5	54.9	2326	
174	-1	65	toyota	diesel	turbo	four	sedan	fwd	front	102.4	175.6	66.5	54.9	2480	
175	-1	65	toyota	gas	std	four	hatchback	fwd	front	102.4	175.6	66.5	53.9	2414	
176	-1	65	toyota	gas	std	four	sedan	fwd	front	102.4	175.6	66.5	54.9	2414	
177	-1	65	toyota	gas	std	four	hatchback	fwd	front	102.4	175.6	66.5	53.9	2458	
178	3	197	toyota	gas	std	two	hatchback	rwd	front	102.9	183.5	67.7	52.0	2976	
179	3	197	toyota	gas	std	two	hatchback	rwd	front	102.9	183.5	67.7	52.0	3016	
180	-1	90	toyota	gas	std	four	sedan	rwd	front	104.5	187.8	66.5	54.1	3131	
181	-1	107	toyota	gas	std	four	wagon	rwd	front	104.5	187.8	66.5	54.1	3151	
182	2	122	volkswagen	diesel	std	two	sedan	fwd	front	97.3	171.7	65.5	55.7	2261	
183	2	122	volkswagen	gas	std	two	sedan	fwd	front	97.3	171.7	65.5	55.7	2209	
184	2	94	volkswagen	diesel	std	four	sedan	fwd	front	97.3	171.7	65.5	55.7	2264	
185	2	94	volkswagen	gas	std	four	sedan	fwd	front	97.3	171.7	65.5	55.7	2212	
186	2	94	volkswagen	gas	std	four	sedan	fwd	front	97.3	171.7	65.5	55.7	2275	
187	2	94	volkswagen	diesel	turbo	four	sedan	fwd	front	97.3	171.7	65.5	55.7	2319	
188	2	94	volkswagen	gas	std	four	sedan	fwd	front	97.3	171.7	65.5	55.7	2300	
189	3	107	volkswagen	gas	std	two	convertible	fwd	front	94.5	159.3	64.2	55.6	2254	
190	3	256	volkswagen	gas	std	two	hatchback	fwd	front	94.5	165.7	64.0	51.4	2221	
191	0	107	volkswagen	gas	std	four	sedan	fwd	front	100.4	180.2	66.9	55.1	2661	
192	0	107	volkswagen	diesel	turbo	four	sedan	fwd	front	100.4	180.2	66.9	55.1	2579	
193	0	107	volkswagen	gas	std	four	wagon	fwd	front	100.4	183.1	66.9	55.1	2563	
194	-2	103	volvo	gas	std	four	sedan	rwd	front	104.3	188.8	67.2	56.2	2912	
195	-1	74	volvo	gas	std	four	wagon	rwd	front	104.3	188.8	67.2	57.5	3034	
196	-2	103	volvo	gas	std	four	sedan	rwd	front	104.3	188.8	67.2	56.2	2935	
197	-1	74	volvo	gas	std	four	wagon	rwd	front	104.3	188.8	67.2	57.5	3042	
198	-2	103	volvo	gas	turbo	four	sedan	rwd	front	104.3	188.8	67.2	56.2	3045	
199	-1	74	volvo	gas	turbo	four	wagon	rwd	front	104.3	188.8	67.2	57.5	3157	
200	-1	95	volvo	gas	std	four	sedan	rwd	front	109.1	188.8	68.9	55.5	2952	
201	-1	95	volvo	gas	turbo	four	sedan	rwd	front	109.1	188.8	68.8	55.5	3049	
202	-1	95	volvo	gas	std	four	sedan	rwd	front	109.1	188.8	68.9	55.5	3012	
203	-1	95	volvo	diesel	turbo	four	sedan	rwd	front	109.1	188.8	68.9	55.5	3217	
204	-1	95	volvo	gas	turbo	four	sedan	rwd	front	109.1	188.8	68.9	55.5	3062	

In [67]:
```
df2['bore'] = df2['bore'].astype(float)
```

In [68]:
```
df2['stroke'] = df2['stroke'].astype(float)
```

In [69]:
```
df2['horsepower'] = df2['horsepower'].astype(float)
```

In [70]:
```
df2['peak-rpm'] = df2['peak-rpm'].astype(float)
```

In [71]:
```
df2['price'] = df2['price'].astype(float)
```

In [72]:
```
df2.describe().T
```

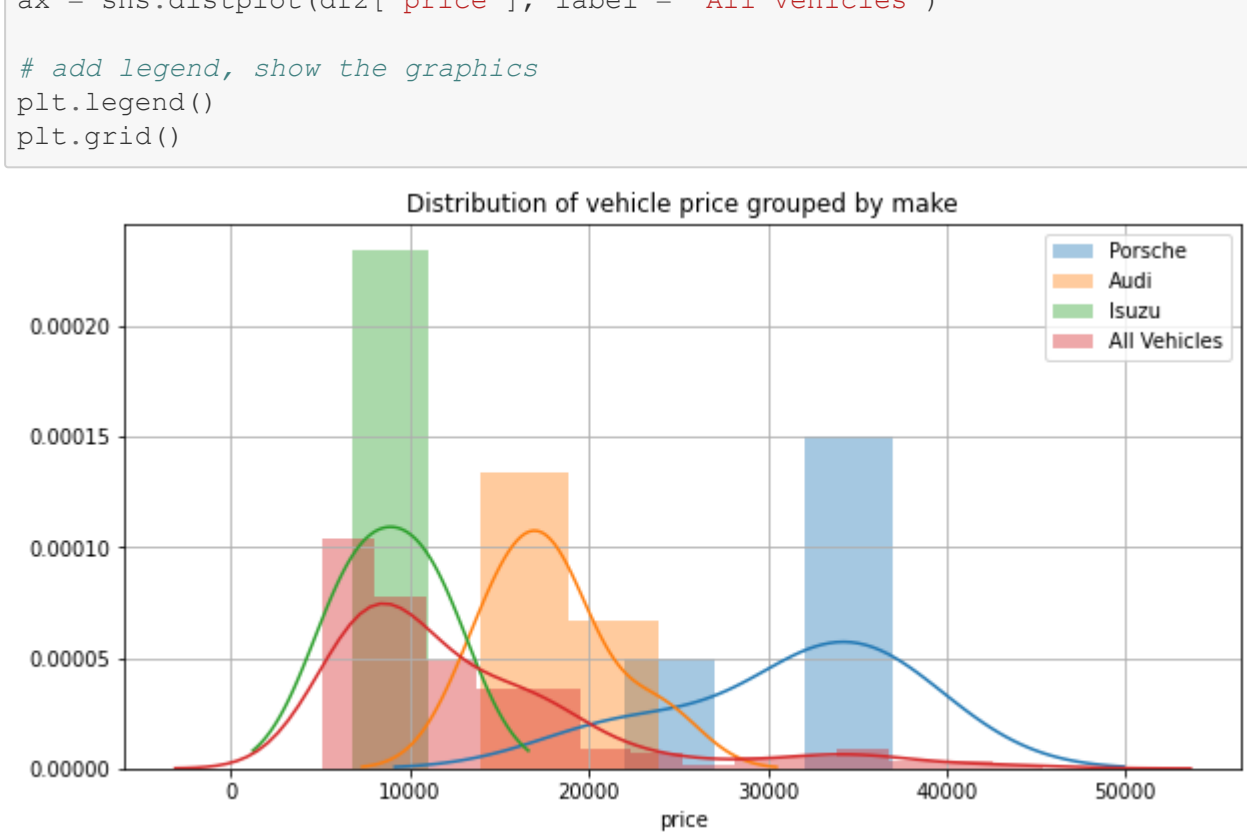
Out[72]:

	count	mean	std	min	25%	50%	75%	max
symboling	205.0	0.834146	1.245307	-2.00	0.00	1.00	2.00	3.00
wheel-base	205.0	98.756585	6.021776	86.60	94.50	97.00	102.40	120.90
length	205.0	174.049268	12.337289	141.10	166.30	173.20	183.10	208.10
width	205.0	65.907805	2.145204	60.30	64.10	65.50	66.90	72.30
height	205.0	53.724878	2.443522	47.80	52.00	54.10	55.50	59.80
curb-weight	205.0	2555.565854	520.680204	1488.00	2145.00	2414.00	2935.00	4066.00
engine-size	205.0	119.448780	28.632108	61.00	97.00	110.00	136.00	203.00
bore	201.0	3.329751	0.273539	2.54	3.15	3.31	3.59	3.94
stroke	201.0	3.255423	0.316717	2.07	3.11	3.29	3.41	4.17
compression-ratio	205.0	10.142537	3.972040	7.00	8.60	9.00	9.40	23.00
horsepower	203.0	104.256158	39.714369	48.00	70.00	95.00	116.00	288.00
peak-rpm	203.0	5125.369458	479.334560	4150.00	4800.00	5200.00	5500.00	6600.00
city-mpg	205.0	25.219512	6.542142	13.00	19.00	24.00	30.00	49.00
highway-mpg	205.0	30.751220	6.886443	16.00	25.00	30.00	34.00	54.00
price	201.0	13207.129353	7947.066342	5118.00	7775.00	10295.00	16500.00	45400.00

In [73]:
```
plt.figure(figsize=(10,5))
plt.title("Distribution Vehicle Peak-RPM")
ax = sns.distplot(df2["peak-rpm"])
plt.grid()
```

Peak-RPM distribution is mostly normal, substitute mean peak-rpm for missing data.

In [74]:
```
plt.figure(figsize=(10,5))
plt.title("Distribution Vehicle Price")
ax = sns.distplot(df2["price"])
plt.grid()
```



Vehicle price distribution is heavily right skewed.

In [75]:
```
# Create a figure, add title
plt.figure(figsize=(10,5))
plt.title("Distribution of vehicle price grouped by make")

# Plot the distribution of Porsche
mask0 = df2['make'] == 'porsche'
ax = sns.distplot(df2["price"][mask0], label = 'Porsche')

# Plot the histogram of Audi
mask1 = df2['make'] == 'audi'
ax = sns.distplot(df2["price"][mask1], label = 'Audi')

mask2 = df2['make'] == 'isuzu'
ax = sns.distplot(df2['price'][mask2], label = 'Isuzu')

#Plot the histogram of All Prices
ax = sns.distplot(df2["price"], label = 'All Vehicles')

# add legend, show the graphics
plt.legend()
plt.grid()
```

In [76]:
```
is_audi = df2['make'] == 'audi'
data_audi = df2[is_audi]
data_audi['price'].describe().T
```

Out[76]:
```
count        6.000000
mean     17859.166667
std       3452.379493
min      13950.000000
25%      15800.000000
50%      17580.000000
75%      18617.500000
max      23875.000000
Name: price, dtype: float64
```

In [77]:
```
data_audi['price'].median()
```

Out[77]:
```
17580.0
```

In [78]:
```
is_porsche = df2['make'] == 'porsche'
data_porsche = df2[is_porsche]
data_porsche['price'].describe().T
```

Out[78]:
```
count        4.000000
mean     31400.500000
std       6528.784343
min      22018.000000
25%      29900.500000
50%      33278.000000
75%      34778.000000
max      37028.000000
Name: price, dtype: float64
```

In [79]:
```
data_porsche['price'].median()
```

Out[79]:
```
33278.0
```

In [80]:
```
is_isuzu = df2['make'] == 'isuzu'
data_isuzu = df2[is_isuzu]
data_isuzu['price'].describe().T
```

Out[80]:
```
count        2.000000
mean      8916.500000
std       3014.396208
min       6785.000000
25%       7850.750000
50%       8916.500000
75%       9982.250000
max      11048.000000
Name: price, dtype: float64
```

In [81]:
```
data_isuzu['price'].median()
```

Out[81]:
```
8916.5
```

Pt 3

In real world scenarios sometimes data does not come in a format that is ideal for doing analysis or model. This is because the data may be skewed or data can be better analyzed if certain features of its variable are combined. For the engine_size, there are certain cars that have extremely larger engine then the median engine_size of the entire data, this will cause a skew distribution, there for we must transform it to standardize its scale. In addition, there may also be imbalanced data within the data set. For example in this data set, in the column aspiration, there are far more cars who has standard aspirators instead of turbo. This will cause our variables of interest to be unevenly represented if we do analyze this column.

A normalization transformation would be best for the data set since it will scale the data appropriately since engine_size and high_way mpg have different measurements.

In [82]:

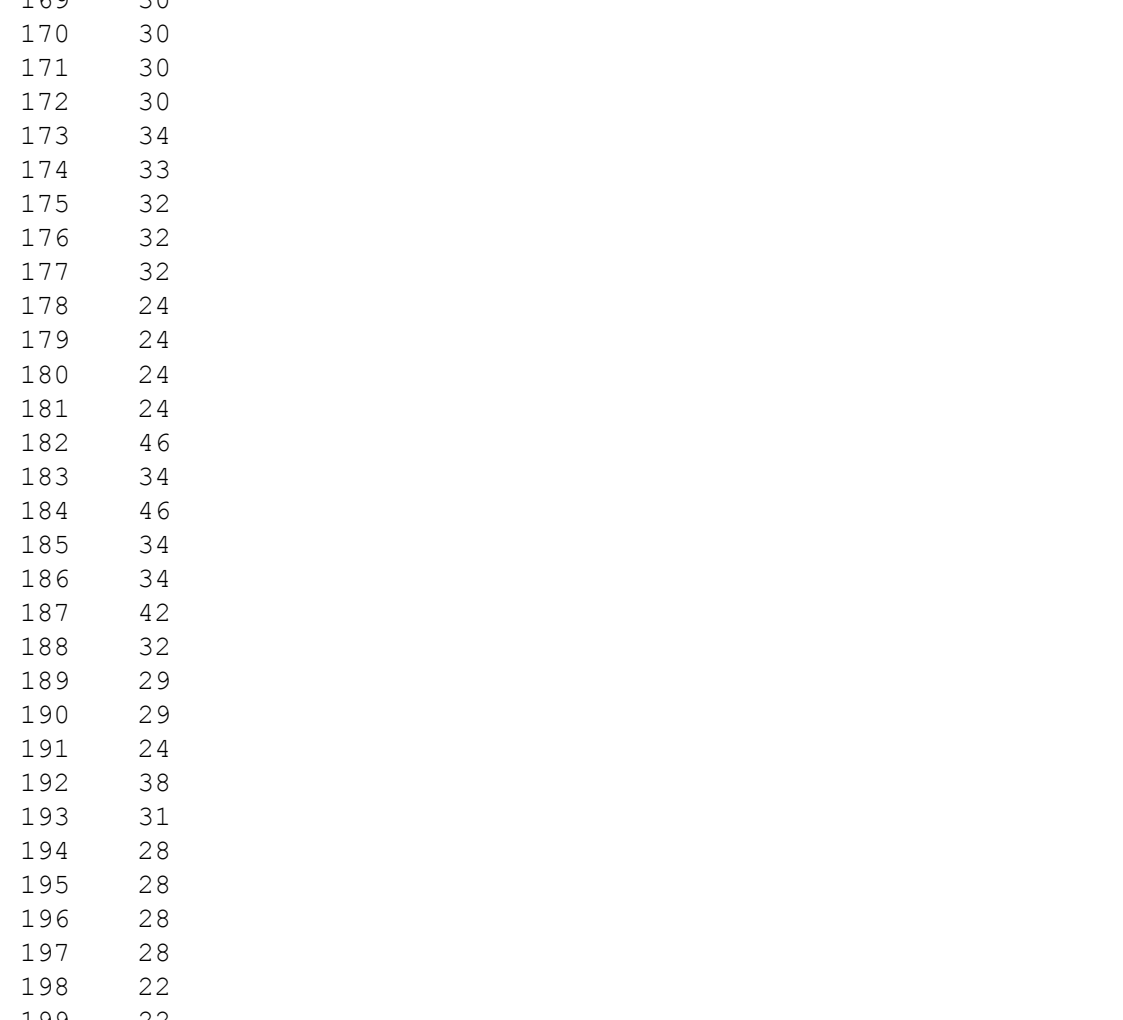
```python
df = df2

engine_size = df['engine-size']
highway_mpg = df['highway-mpg']
data_to_plot = [engine_size, highway_mpg]
data_to_plot
```

Out[82]:

```
[0      130
 1      130
 2      152
 3      109
 4      136
 5      136
 6      136
 7      136
 8      131
 9      131
 10     108
 11     108
 12     164
 13     164
 14     164
 15     102
 16     102
 17     102
 18      61
 19      90
 20      90
 21      90
 22      90
 23      98
 24      90
 25      90
 26      90
 27      98
 28     122
 29     156
 30      92
 31      92
 32      79
 33      92
 34      92
 35      92
 36      92
 37     110
 38     110
 39     110
 40     110
 41     110
 42     110
 43     111
 44      90
 45      90
 46     119
 47     102
 48     102
 49     102
 50      91
 51      91
 52      91
 53      91
 54      91
 55      70
 56      70
 57      70
 58      80
 59     122
 60     122
 61     122
 62     122
 63     122
 64     122
 65     140
 66     134
 67     183
 68     183
 69     183
 70     183
 71     102
 72     102
 73     102
 74     102
 75     140
 76      92
 77      92
 78      92
 79      98
 80     110
 81     122
 82     156
 83     156
 84     156
 85     122
 86     122
 87     110
 88     110
 89      97
 90     103
 91      97
 92      97
 93      97
 94      97
 95      97
 96      97
 97      97
 98      97
 99     120
 100    120
 101    181
 102    181
 103    181
 104    181
 105    181
 106    181
 107    120
 108    152
 109    120
 110    152
 111    120
 112    152
 113    120
 114    152
 115    120
 116    152
 117    134
 118     90
 119     98
 120     90
 121     90
 122     98
 123    122
 124    156
 125    151
 126    194
 127    194
 128    194
 129    203
 130    132
 131    132
 132    121
 133    121
 134    121
 135    121
 136    121
 137    121
 138     97
 139    108
 140    108
 141    108
 142    108
 143    108
 144    108
 145    108
 146    108
 147    108
 148    108
 149    108
 150     92
 151     92
 152     92
 153     92
 154     92
 155     92
 156     98
 157     98
 158    110
 159    110
 160     98
 161     98
 162     98
 163     98
 164     98
 165     98
 166     98
 167    146
 168    146
 169    146
 170    146
 171    146
 172    146
 173    122
 174    110
 175    122
 176    122
 177    122
 178    171
 179    171
 180    171
 181    161
 182     97
 183    109
 184     97
 185    109
 186    109
 187     97
 188    109
 189    109
 190    109
 191    136
 192     97
 193    109
 194    141
 195    141
 196    141
 197    141
 198    130
 199    130
 200    141
 201    141
 202    173
 203    145
 204    141
 Name: engine-size, dtype: int64,
 0      27
 1      27
 2      26
 3      30
 4      22
 5      25
 6      25
 7      25
 8      20
 9      22
 10     29
 11     29
 12     28
 13     28
 14     25
 15     22
 16     22
 17     20
 18     53
 19     43
 20     43
 21     41
 22     38
 23     30
 24     38
 25     38
 26     38
 27     30
 28     30
 29     24
 30     54
 31     38
 32     42
 33     34
 34     34
 35     34
 36     34
 37     33
 38     33
 39     33
 40     33
 41     28
 42     31
 43     29
 44     43
 45     43
 46     29
 47     19
 48     19
 49     17
 50     31
 51     38
 52     38
 53     38
 54     38
 55     23
 56     23
 57     23
 58     23
 59     32
 60     32
 61     32
 62     32
 63     42
 64     32
 65     27
 66     39
 67     25
 68     25
 69     25
 70     25
 71     18
 72     18
 73     16
 74     16
 75     24
 76     41
 77     38
 78     38
 79     30
 80     30
 81     32
 82     24
 83     24
 84     24
 85     32
 86     32
 87     30
 88     30
 89     37
 90     50
 91     37
 92     37
 93     37
 94     37
 95     37
 96     37
 97     37
 98     37
 99     34
 100    34
 101    22
 102    22
 103    25
 104    25
 105    23
 106    25
 107    24
 108    33
 109    24
 110    25
 111    24
 112    33
 113    24
 114    25
 115    24
 116    33
 117    24
 118    41
 119    30
 120    38
 121    38
 122    38
 123    30
 124    24
 125    27
 126    25
 127    25
 128    25
 129    28
 130    31
 131    31
 132    28
 133    28
 134    28
 135    28
 136    26
 137    26
 138    36
 139    31
 140    31
 141    37
 142    33
 143    32
 144    25
 145    29
 146    32
 147    31
 148    29
 149    23
 150    39
 151    38
 152    38
 153    37
 154    32
 155    32
 156    37
 157    37
 158    36
 159    47
 160    47
 161    34
 162    34
 163    34
 164    34
 165    29
 166    29
 167    30
 168    30
 169    30
 170    30
 171    30
 172    30
 173    34
 174    33
 175    32
 176    32
 177    32
 178    24
 179    24
 180    24
 181    24
 182    46
 183    34
 184    46
 185    34
 186    34
 187    42
 188    32
 189    29
 190    29
 191    24
 192    38
 193    31
 194    28
 195    28
 196    28
 197    28
 198    22
 199    22
 200    28
 201    25
 202    23
 203    27
 204    25
 Name: highway-mpg, dtype: int64]
```

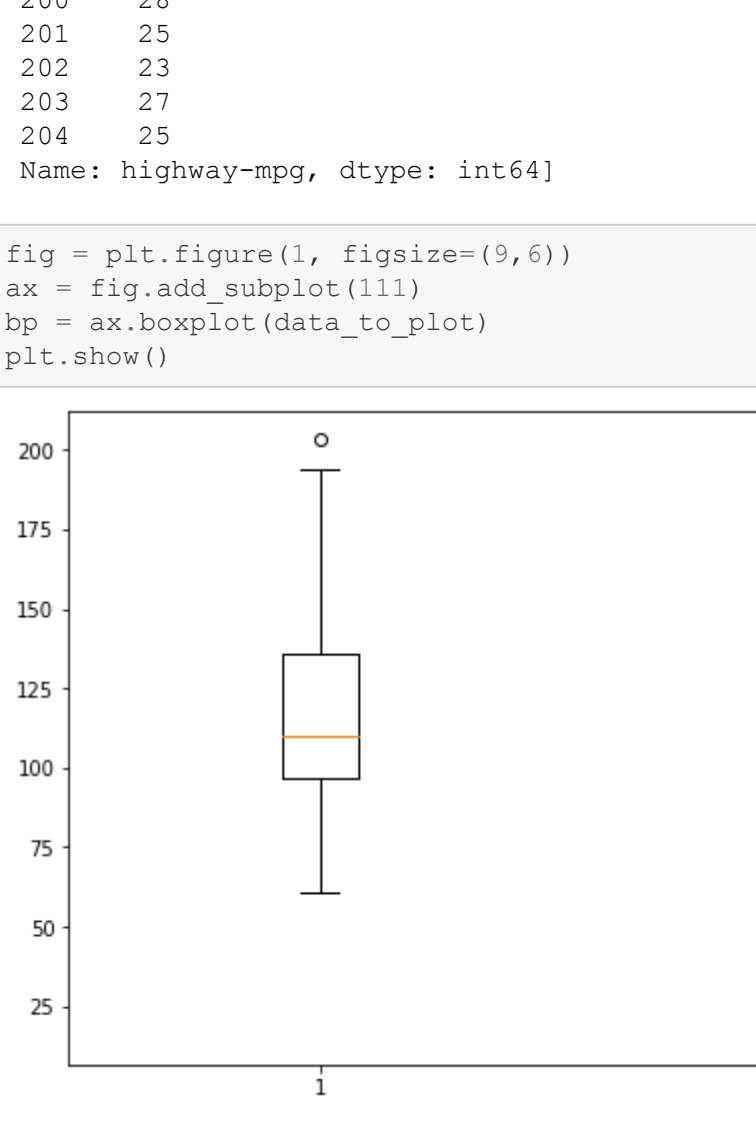
In [83]:

```python
fig = plt.figure(1, figsize=(9,6))
ax = fig.add_subplot(111)
bp = ax.boxplot(data_to_plot)
plt.show()
```



In [84]:

```python
plt.hist(engine_size, alpha=0.5, label='enginesize', bins=30)
plt.hist(highway_mpg, alpha=0.5, label='highwaympg', bins=30)
plt.show()
```



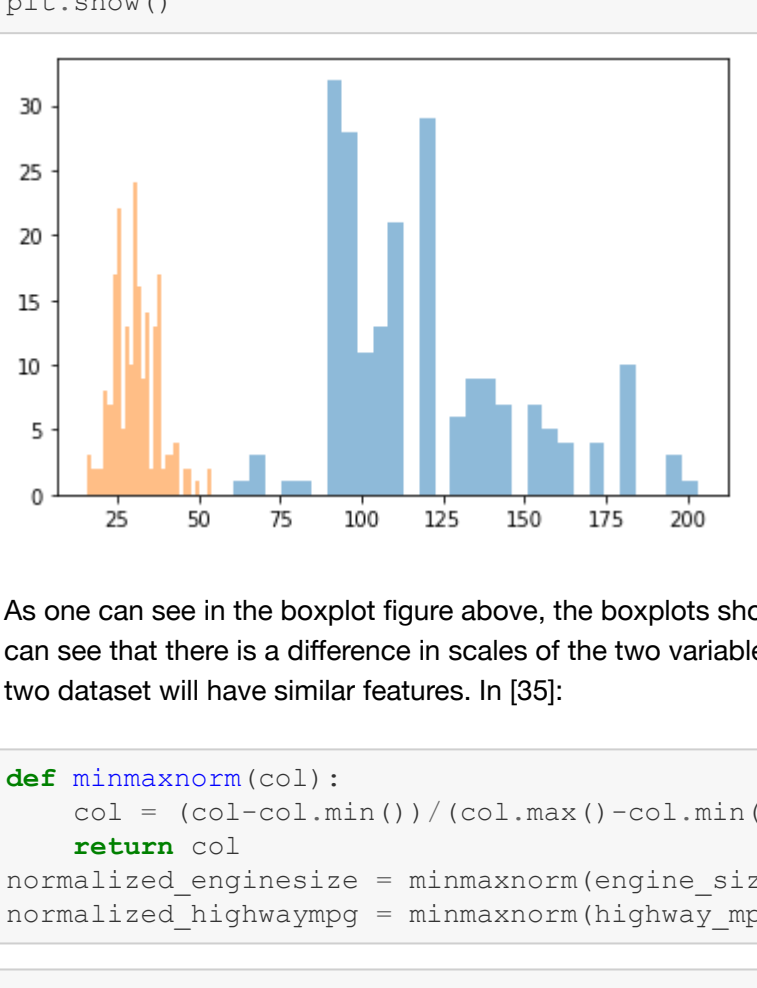
As one can see in the boxplot figure above, the boxplots show outliers in both variables of dataset. Additionally in the histogram figure, we can see that there is a difference in scales of the two variables, thus we need to use minmax normalization to normalize these scales so the two dataset will have similar features. In [35]:

In [85]:

```python
def minmaxnorm(col):
    col = (col-col.min())/(col.max()-col.min())
    return col
normalized_enginesize = minmaxnorm(engine_size)
normalized_highwaympg = minmaxnorm(highway_mpg)
```

In [86]:

```python
plt.hist(normalized_enginesize, alpha=0.5, label='enginesize', bins=30)
plt.hist(normalized_highwaympg, alpha=0.5, label='highwaympg', bins=30)
plt.show()
```



The plot after normalization would be best used to compare the data because it normalized both dataset so each will have similar features as shown above.