```
# Importing the required libraries
import pandas as pd
import numpy as np
import seaborn as sns #visualisation
import matplotlib.pyplot as plt #visualisation
%matplotlib inline
sns.set# Importing the required libraries
import pandas as pd
import numpy as np
import seaborn as sns #visualisation
import matplotlib.pyplot as plt #visualisation
%matplotlib inline
sns.set(color_codes=True)
```

```
# Loading the CSV file into a pandas dataframe.
df = pd.read_csv("CARS.csv")
df.head(5)
```

	Make	Model	Type	Origin	DriveTrain	MSRP	Invoice	EngineSize	Cylinders	Horsepower	MPG_Cit
0	Acura	MDX	SUV	Asia	All	$36,945	$33,337	3.5	6.0	265	1
1	Acura	RSX Type S 2dr	Sedan	Asia	Front	$23,820	$21,761	2.0	4.0	200	2
2	Acura	TSX 4dr	Sedan	Asia	Front	$26,990	$24,647	2.4	4.0	200	2
		TL									

```
# Removing irrelevant features
df = df.drop(['Model','DriveTrain','Invoice', 'Origin', 'Type'], axis=1)
df.head(5)
```

	Make	MSRP	EngineSize	Cylinders	Horsepower	MPG_City	MPG_Highway	Weight	Wheelbase	Length
0	Acura	$36,945	3.5	6.0	265	17	23	4451	106	189
1	Acura	$23,820	2.0	4.0	200	24	31	2778	101	172
2	Acura	$26,990	2.4	4.0	200	22	29	3230	105	183
3	Acura	$33,195	3.2	6.0	270	20	28	3575	108	186
4	Acura	$43,755	3.5	6.0	225	18	24	3880	115	197

```
# To identify the type of data
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 428 entries, 0 to 427
Data columns (total 10 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   Make         428 non-null    object
 1   MSRP         428 non-null    object
 2   EngineSize   428 non-null    float64
 3   Cylinders    426 non-null    float64
 4   Horsepower   428 non-null    int64
 5   MPG_City     428 non-null    int64
 6   MPG_Highway  428 non-null    int64
 7   Weight       428 non-null    int64
 8   Wheelbase    428 non-null    int64
 9   Length       428 non-null    int64
dtypes: float64(2), int64(6), object(2)
memory usage: 33.6+ KB
```

```
# Getting the number of instances and features
df.shape
```

```
(428, 10)
```

```
# Getting the dimensions of the data frame
df.ndim
```

```
2
```

```
df = df.drop_duplicates(subset='MSRP', keep='first')
df.count()
```

```
Make           410
MSRP           410
EngineSize     410
Cylinders      408
Horsepower     410
MPG_City       410
MPG_Highway    410
Weight         410
Wheelbase      410
Length         410
dtype: int64
```

```
# Finding the null values
print(df.isnull().sum())
```

```
Make           0
MSRP           0
EngineSize     0
Cylinders      2
Horsepower     0
MPG_City       0
MPG_Highway    0
Weight         0
Wheelbase      0
Length         0
dtype: int64
```

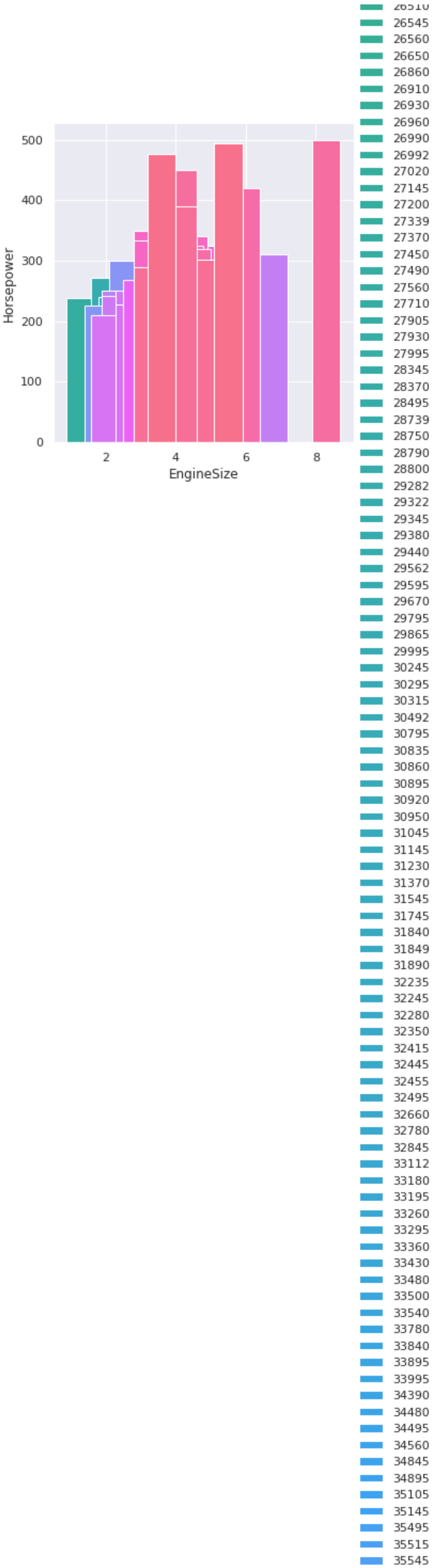
```
df.Horsepower.max()
```

```
500
```

```
# Printing the null value rows
df[240:242]
```

	Make	MSRP	EngineSize	Cylinders	Horsepower	MPG_City	MPG_Highway	Weight	W
247	Mazda	$25,700	1.3	NaN	197	18	25	3053	
248	Mazda	$27,200	1.3	NaN	238	18	24	3029	

```
sns.FacetGrid(df, hue='MSRP', height=5).map(plt.bar, 'EngineSize', 'Horsepower').add_legend()
```



500
400
300
200
100
0
Horsepower
2
4
6
8
EngineSize
26545
26560
26650
26860
26910
26930
26960
26990
26992
27020
27145
27200
27339
27370
27450
27490
27560
27710
27905
27930
27995
28345
28370
28495
28739
28750
28790
28800
29282
29322
29345
29380
29440
29562
29595
29670
29795
29865
29995
30245
30295
30315
30492
30795
30835
30860
30895
30920
30950
31045
31145
31230
31370
31545
31745
31840
31849
31890
32235
32245
32280
32350
32415
32445
32455
32495
32660
32780
32845
33112
33180
33195
33260
33295
33360
33430
33480
33500
33540
33780
33840
33895
33995
34390
34480
34495
34560
34845
34895
35105
35145
35495
35515
35545

```
# Filling the rows with the mean of the column
val = df['Cylinders'].mean()
df['Cylinders'][247] = round(val)
val = df['Cylinders'].mean()
df['Cylinders'][248]= round(val)
```

```
<ipython-input-19-d9e0250049dd>:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-co
  df['Cylinders'][248]= round(val)
```

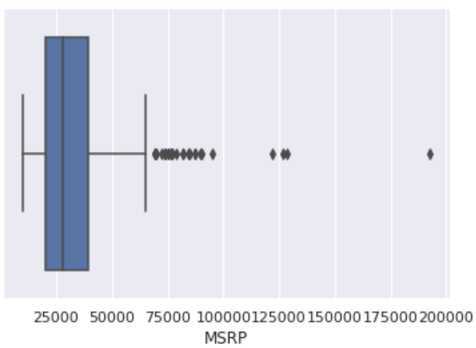
75000

```
# Removing the formatting
df['MSRP'] = [x.replace('$', '') for x in df['MSRP']]
df['MSRP'] = [x.replace(',', '') for x in df['MSRP']]
df['MSRP']=pd.to_numeric(df['MSRP'],errors='coerce')
```

84165

```
sns.boxplot(x=df['MSRP'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f90de5d4df0>
```



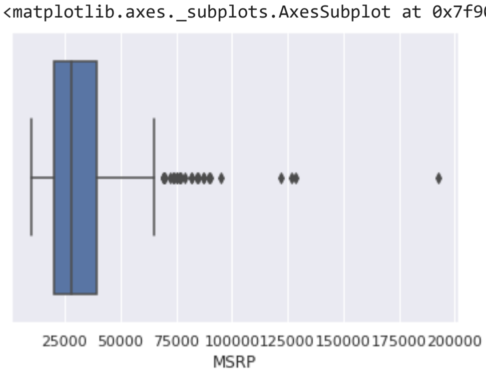

```
Q1 = df.quantile(0.25)
Q3 = df.quantile(0.75)
```

```
IQR = Q3 - Q1
print(IQR)
```

```
MSRP           19086.50
EngineSize         1.55
Cylinders          2.00
Horsepower        85.00
MPG_City           4.00
MPG_Highway        5.00
Weight           872.25
Wheelbase          9.00
Length            16.00
dtype: float64
```

```
sns.boxplot(x=df['MSRP'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f90de44ebe0>
```

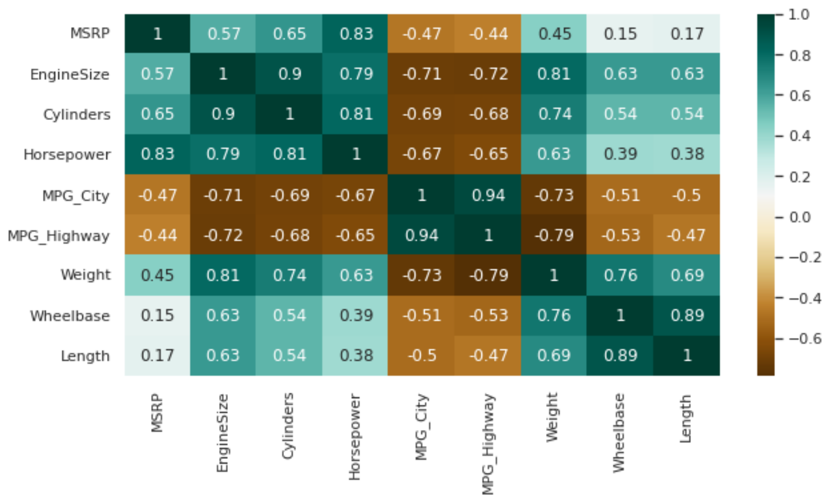


```
df.describe()
```

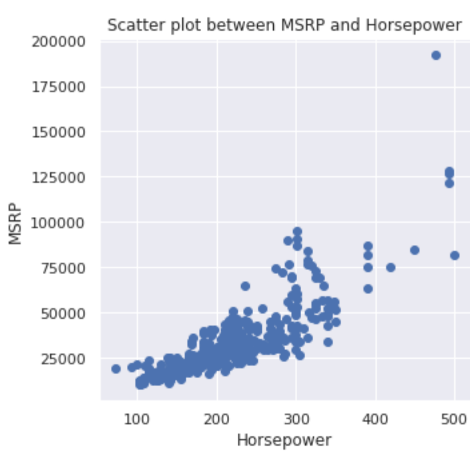
	MSRP	EngineSize	Cylinders	Horsepower	MPG_City	MPG_Highway	Weight	Wheelbas
count	410.000000	410.000000	410.000000	410.000000	410.000000	410.000000	410.000000	410.00000
mean	32919.021951	3.200976	5.824390	216.007317	20.039024	26.792683	3581.221951	108.16829
std	19628.241002	1.115264	1.556873	72.330689	5.280720	5.790004	766.396651	8.36563
min	10280.000000	1.300000	3.000000	73.000000	10.000000	12.000000	1850.000000	89.00000
25%	20324.750000	2.325000	4.000000	165.000000	17.000000	24.000000	3102.000000	103.00000
50%	27807.500000	3.000000	6.000000	210.000000	19.000000	26.000000	3476.000000	107.00000
75%	39411.250000	3.875000	6.000000	250.000000	21.000000	29.000000	3974.250000	112.00000
max	192465.000000	8.300000	12.000000	500.000000	60.000000	66.000000	7190.000000	144.00000

```
# Plotting a heat map
plt.figure(figsize=(10,5))
c= df.corr()
sns.heatmap(c,cmap="BrBG",annot=True)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f90de5d4fa0>
```

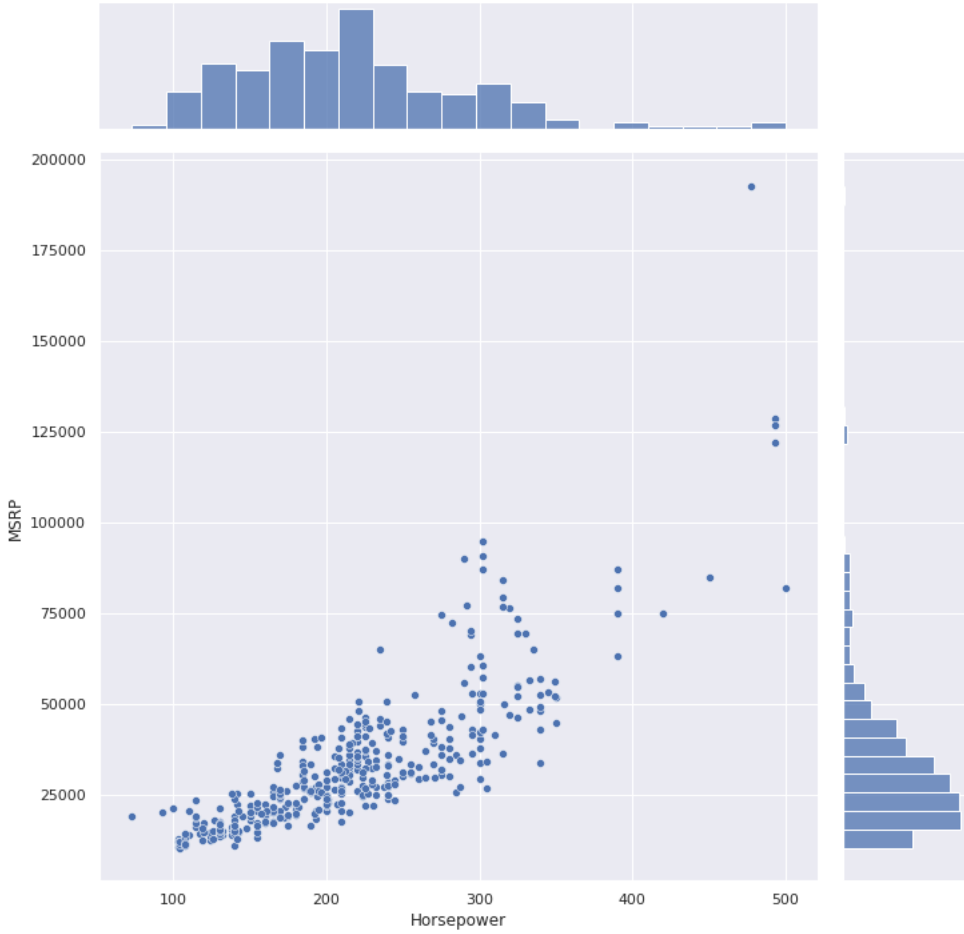



```
# Plotting a scatter plot
fig, ax = plt.subplots(figsize=(5,5))
ax.scatter(df['Horsepower'], df['MSRP'])
plt.title('Scatter plot between MSRP and Horsepower')
ax.set_xlabel('Horsepower')
ax.set_ylabel('MSRP')
plt.show()
```



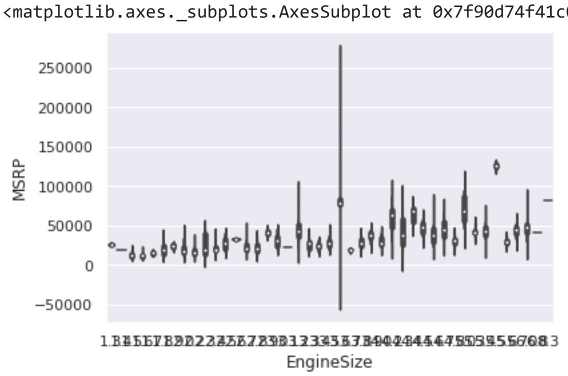

```
sns.jointplot(x='Horsepower', y='MSRP', data=df, height=10)
```

<seaborn.axisgrid.JointGrid at 0x7f90d9b91940>




```
sns.violinplot(x='EngineSize', y='MSRP', data=df, height=6)
```

<matplotlib.axes._subplots.AxesSubplot at 0x7f90d74f41c0>

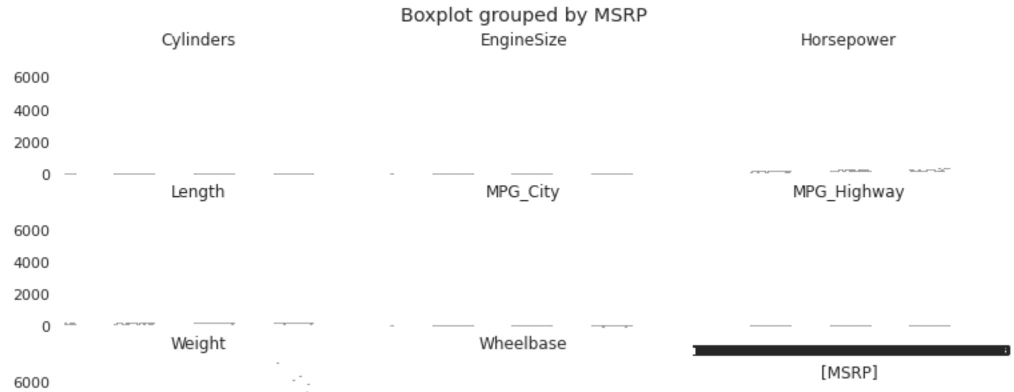



```
sns.pairplot(df, hue='MSRP', height=2)
```

<seaborn.axisgrid.PairGrid at 0x7f90d69b7d30>


```
df.boxplot(by='MSRP', figsize=(12, 6))
```

```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x7f90d6d84670>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f90d6257610>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f90d63053d0>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x7f90d7436430>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f90d74eb3d0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f90d6107520>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x7f90d613c490>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f90d750e670>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7f90d62c8730>]],
      dtype=object)
```



✓ 12s completed at 10:41 PM