LINE CHARTS

Line charts are pretty much useful and very common in practice. Line charts are the way to depict relations b/w two continuous variables and hence they are bivariate i.e you need at least 2 variables to plot a line chart.

Advantages :

1. Easy to interpret the trends over time and also helps to extrapolate the data.
2. A-line break can also depict the presence of missing data.
3. We can also estimate the correlation value by seeing the steepness in the line.
   Disadvantages:
4. Though line charts are easy to interpret but plotting two line charts over the same figure can make it difficult to compare the results.

In [1]:
```
import pandas as pd
import numpy as np
```

In [2]:
```
df = pd.read_csv(r'E:\Downloads\auto-mpg.csv', na_values = ['?'])
df.head()
```

Out[2]:

	mpg	cylinders	displacement	horsepower	weight	acceleration	model year	origin	car name
0	18.0	8	307.0	130.0	3504	12.0	70	1	chevrolet chevelle malibu
1	15.0	8	350.0	165.0	3693	11.5	70	1	buick skylark 320
2	18.0	8	318.0	150.0	3436	11.0	70	1	plymouth satellite
3	16.0	8	304.0	150.0	3433	12.0	70	1	amc rebel sst
4	17.0	8	302.0	140.0	3449	10.5	70	1	ford torino

In [3]:
```
df.isna().sum()
```

Out[3]:
```
mpg             0
cylinders       0
displacement    0
horsepower      6
weight          0
acceleration    0
model year      0
origin          0
car name        0
dtype: int64
```

In [4]:
```
df['horsepower'].fillna(df['horsepower'].median(), inplace = True)
```

In [5]:
```
df.sort_values(by = 'mpg', inplace = True)
```

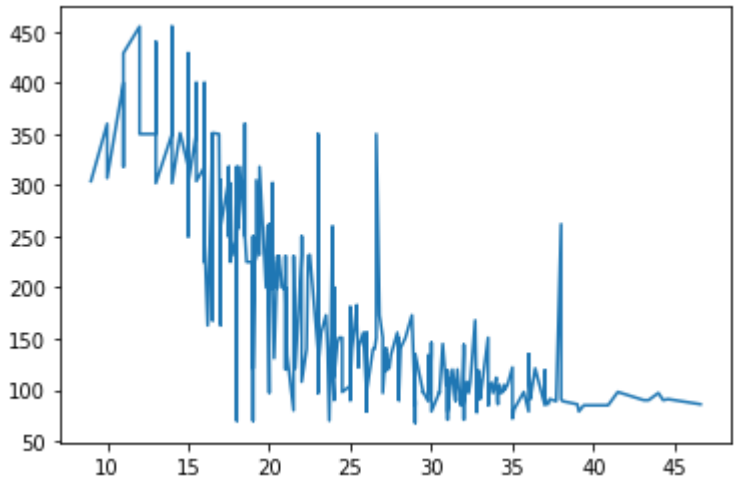
In [6]:
```
df.head()
```

Out[6]:

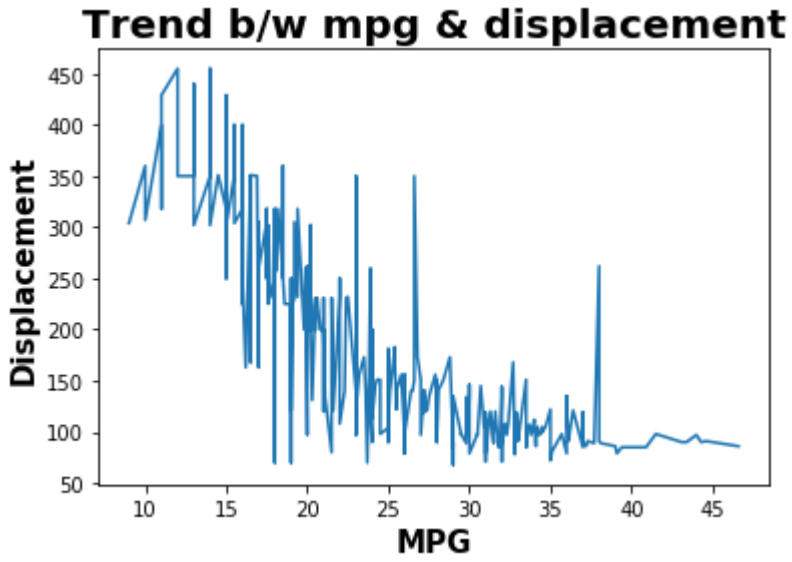
	mpg	cylinders	displacement	horsepower	weight	acceleration	model year	origin	car name
28	9.0	8	304.0	193.0	4732	18.5	70	1	hi 1200d
25	10.0	8	360.0	215.0	4615	14.0	70	1	ford f250
26	10.0	8	307.0	200.0	4376	15.0	70	1	chevy c20
103	11.0	8	400.0	150.0	4997	14.0	73	1	chevrolet impala
124	11.0	8	350.0	180.0	3664	11.0	73	1	oldsmobile omega

In [7]:
```
import matplotlib.pyplot as plt
plt.style.use('default')
```

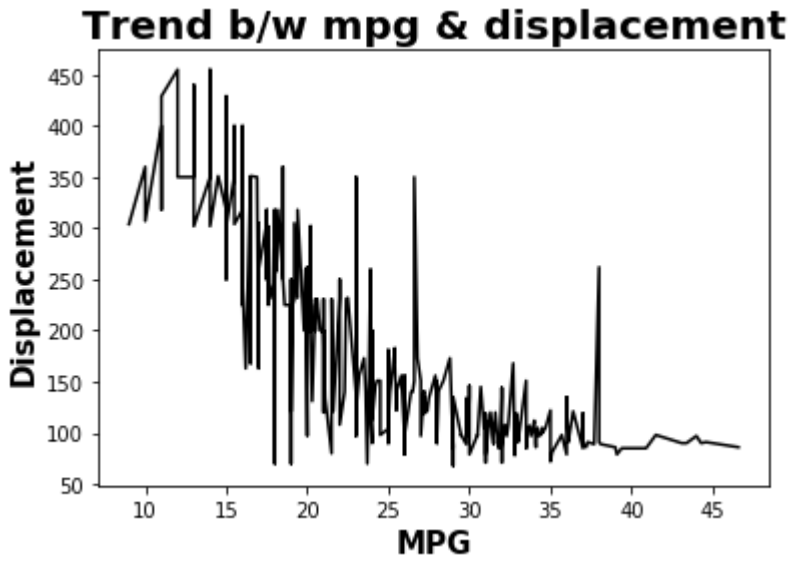
In [8]:
```
plt.plot(df['mpg'], df['displacement'])
plt.show()
```



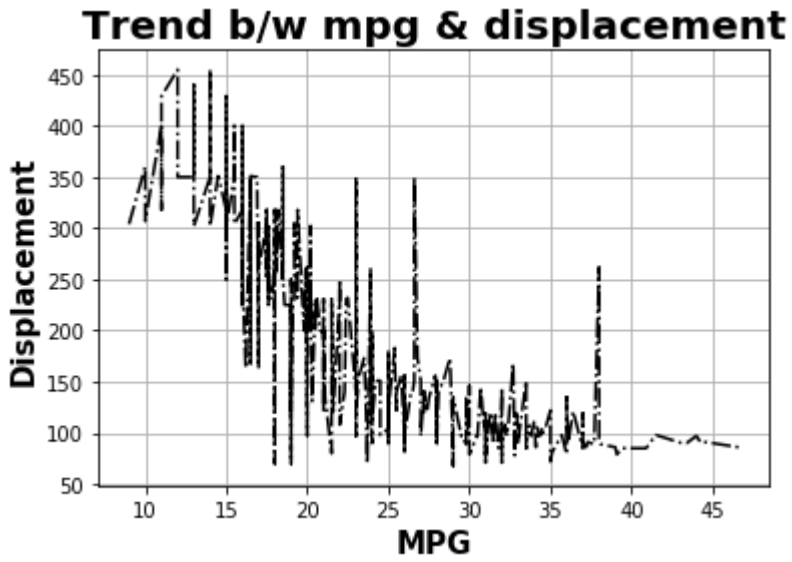
In [9]:
```
plt.plot(df['mpg'], df['displacement'])
plt.xlabel('MPG', size = 15, fontweight = 'bold')
plt.ylabel('Displacement', size = 15, fontweight = 'bold')
plt.title('Trend b/w mpg & displacement', size = 20, fontweight = 'bold')
plt.show()
```



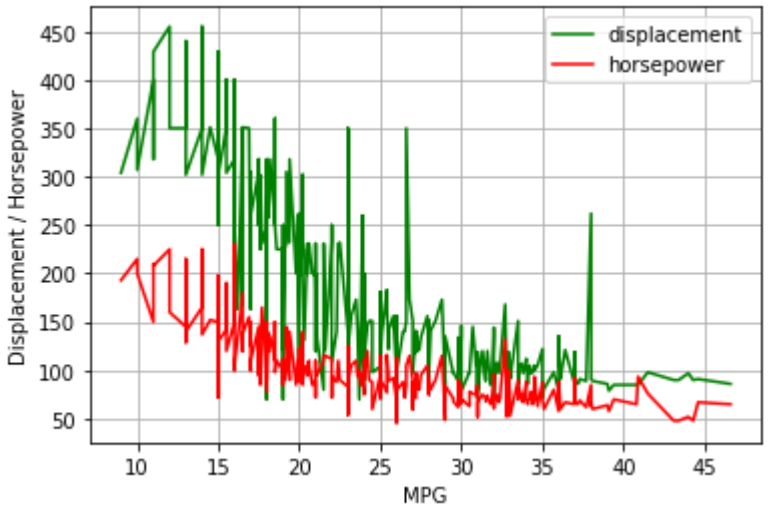
In [10]:
```
plt.plot(df['mpg'], df['displacement'], color = 'k')
plt.xlabel('MPG', size = 15, fontweight = 'bold')
plt.ylabel('Displacement', size = 15, fontweight = 'bold')
plt.title('Trend b/w mpg & displacement', size = 20, fontweight = 'bold')
#plt.grid()
plt.show()
```



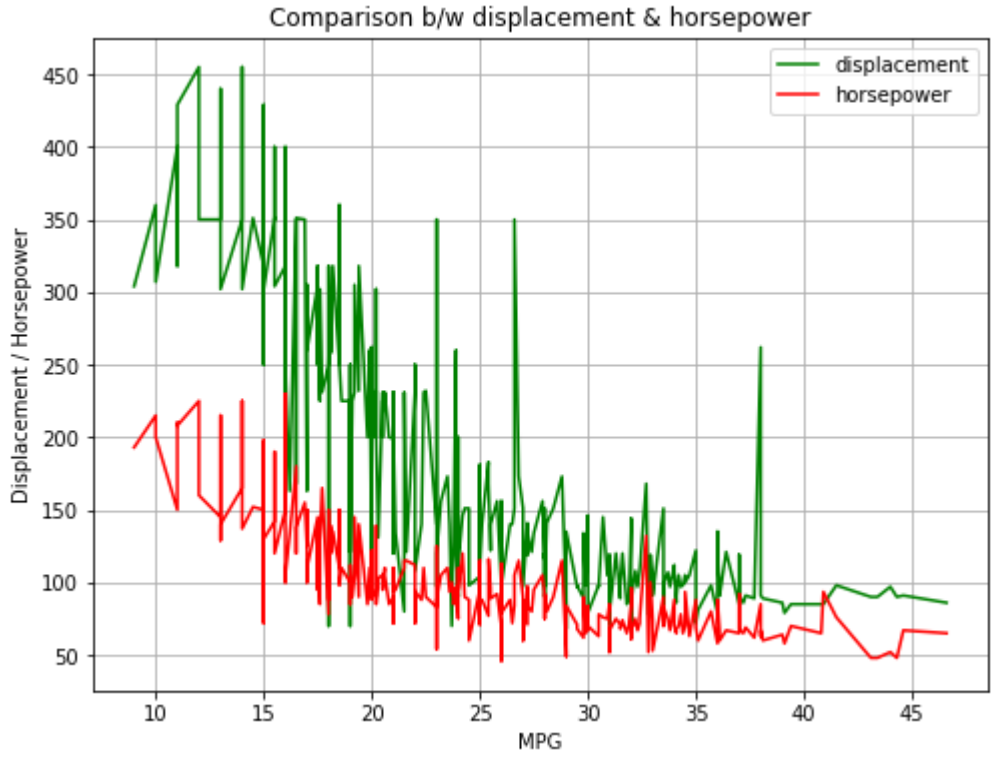
In [11]:
```
plt.plot(df['mpg'], df['displacement'], color = 'k',
        linestyle = '-.')
plt.xlabel('MPG', size = 15, fontweight = 'bold')
plt.ylabel('Displacement', size = 15, fontweight = 'bold')
plt.title('Trend b/w mpg & displacement', size = 20, fontweight = 'bold')
plt.grid()
plt.show()
```



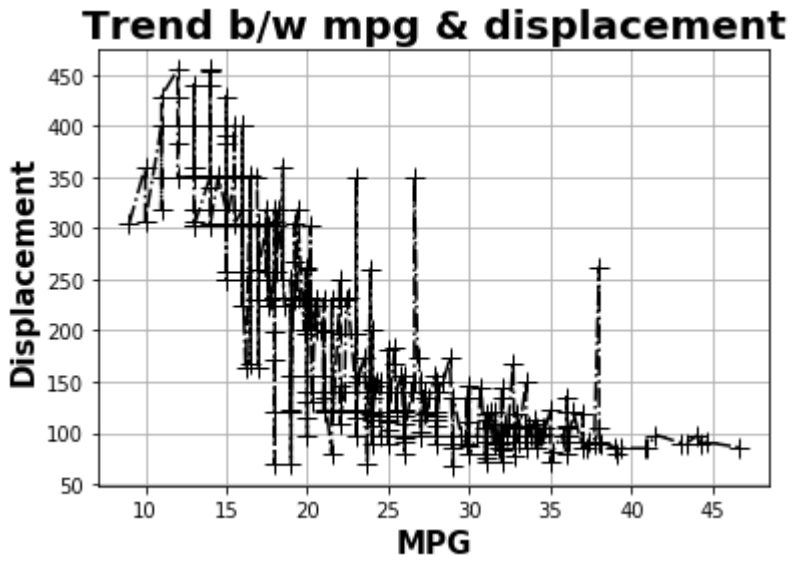
In [12]:
```
plt.plot(df['mpg'], df['displacement'], label = 'displacement', color ='g')
plt.plot(df['mpg'], df['horsepower'], label = 'horsepower', color ='r')
plt.xlabel('MPG')
plt.ylabel('Displacement / Horsepower')
plt.legend()
plt.grid()
plt.show()
```



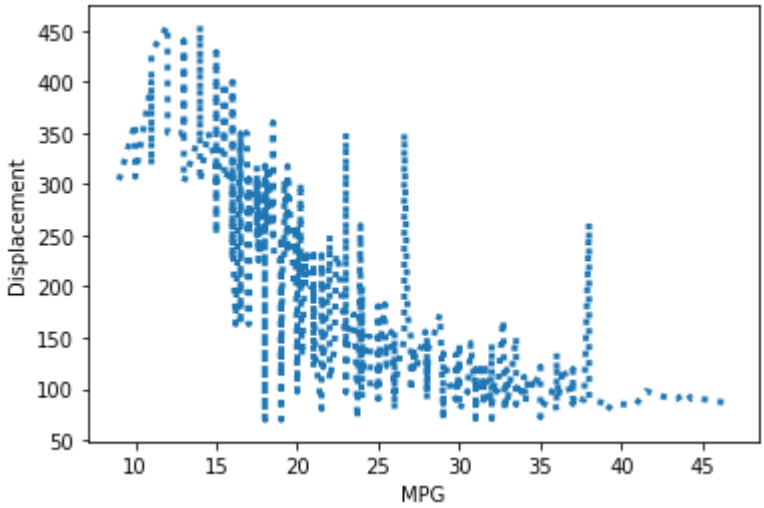
In [13]:
```
plt.figure(figsize = (8,  6))
plt.plot(df['mpg'], df['displacement'], label = 'displacement', color ='g')
plt.plot(df['mpg'], df['horsepower'], label = 'horsepower', color ='r')
plt.xlabel('MPG')
plt.ylabel('Displacement / Horsepower')
plt.legend()
plt.grid()
plt.title('Comparison b/w displacement & horsepower')
plt.savefig(r'E:\Downloads\linechart.png')
```



In [14]:
```
plt.plot(df['mpg'], df['displacement'], color = 'k',
        linestyle = '-.', marker = '+', markersize = 10)
plt.xlabel('MPG', size = 15, fontweight = 'bold')
plt.ylabel('Displacement', size = 15, fontweight = 'bold')
plt.title('Trend b/w mpg & displacement', size = 20, fontweight = 'bold')
plt.grid()
plt.show()
```



In [15]:
```
plt.plot(df['mpg'], df['displacement'], linestyle = ':',
        linewidth = 3)
plt.xlabel('MPG')
plt.ylabel('Displacement')
plt.show()
```



In [ ]:
```

```