Q.1) Download the car dataset from the following link-

https://www.kaggle.com/CooperUnion/cardataset

Import required libraries, read data, display car companies with car numbers, clean data and display sample data for 2 variables.

Solution:

```
#Importing Libraries
import pandas as pd
#Reading data
df = pd.read_csv('C:/TYBSC/car_data.csv') # Importing the data set
df.sample(5)
print(df.sample(5))
print(df.shape)
#Print car companies with car numbers
print(df['Make'].value_counts())
new_df = df[df['Make']=='Volkswagen']
print(new_df.shape) # Viewing the new dataset shape
#Data Cleaning
print(new_df.isnull().sum())
new_df = new_df.dropna()
print(new_df.isnull().sum())
new_df.shape
new_df.isnull().sum()
print(new_df.sample(2))
#Sample data for 2 variables..
new_df = new_df[['Engine HP','MSRP']]
print(new_df.sample(5))
```

Output:

runfile('C:/TYBSC/LinearReg1.py', wdir='C:/TYBSC')

	Make	Model	Year	...	city mpg	Popularity	MSRP
10785	Chevrolet	Trax	2017	...	24	1385	22500
1991	Pontiac	Bonneville	2005	...	15	210	35585
1772	Subaru	B9 Tribeca	2007	...	16	640	34495
9387	GMC	Sierra 1500	2017	...	16	549	52455

7584 Toyota Pickup 1995 ... 17 2031 2553

[5 rows x 16 columns]
(11914, 16)

Chevrolet 1123
Ford 881
Volkswagen 809
Toyota 746
Dodge 626
Nissan 558
GMC 515
Honda 449
Mazda 423
Cadillac 397
Mercedes-Benz 353
Suzuki 351
BMW 334
Infiniti 330
Audi 328
Hyundai 303
Volvo 281
Subaru 256
Acura 252
Kia 231
Mitsubishi 213
Lexus 202
Buick 196
Chrysler 187
Pontiac 186
Lincoln 164
Oldsmobile 150
Land Rover 143
Porsche 136
Saab 111
Aston Martin 93
Plymouth 82
Bentley 74
Ferrari 69
FIAT 62
Scion 60
Maserati 58
Lamborghini 52
Rolls-Royce 31
Lotus 29
Tesla 18
HUMMER 17
Maybach 16
Alfa Romeo 5
McLaren 5
Spyker 3
Genesis 3
Bugatti 3

Name: Make, dtype: int64
(809, 16)
Make 0
Model 0
Year 0
Engine Fuel Type 0
Engine HP 0
Engine Cylinders 4
Transmission Type 0
Driven_Wheels 0
Number of Doors 0
Market Category 224
Vehicle Size 0
Vehicle Style 0
highway MPG 0
city mpg 0
Popularity 0
MSRP 0

dtype: int64
Make 0
Model 0
Year 0
Engine Fuel Type 0
Engine HP 0
Engine Cylinders 0
Transmission Type 0
Driven_Wheels 0
Number of Doors 0
Market Category 0
Vehicle Size 0
Vehicle Style 0
highway MPG 0
city mpg 0
Popularity 0
MSRP 0
dtype: int64

	Make	Model	Year	...	city mpg	Popularity	MSRP
1864	Volkswagen	Beetle Convertible	2016	...	23	873	32670
2959	Volkswagen	Corrado	1992	...	16	873	2000

[2 rows x 16 columns]

	Engine HP	MSRP
6079	140.0	25795
1915	170.0	21795
5332	200.0	24770
5459	170.0	19595
6031	210.0	30875

Q.2)) Download the car dataset from the following link-

https://www.kaggle.com/CooperUnion/cardataset

Import required libraries, read data and split test and train data.

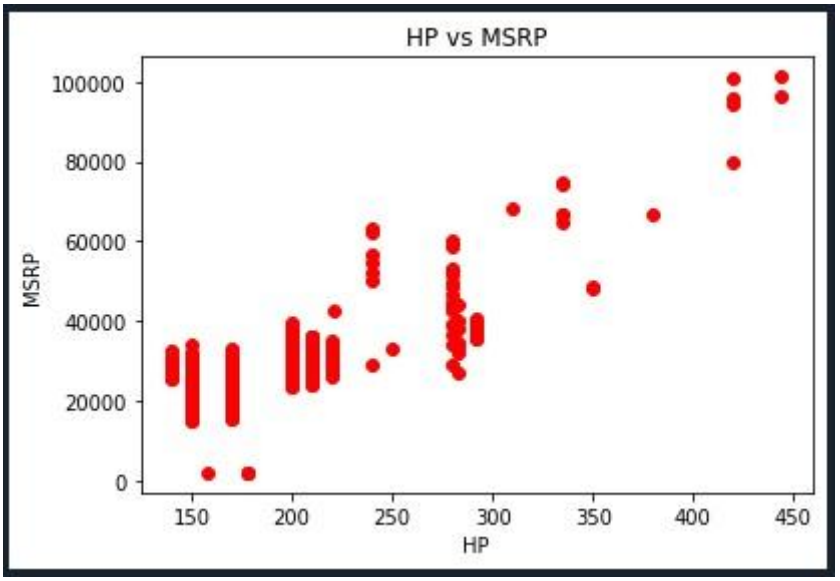
Solution:

```
#Importing libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
#Read Data
df = pd.read_csv('C:/TYBSC/car_data.csv') # Importing the data set
df.sample(5) #previewing dataset randomly
print(df.sample(5))

new_df = df[df['Make']=='Volkswagen'] # in this new data set we only take 'Volkswagen' Cars

#print(new_df.isnull().sum()) # Is there any Null or Empty cell presents
new_df = new_df.dropna() # Deleting the rows which have Empty cells
#print(new_df.isnull().sum()) # Is there any Null or Empty cell presents
new_df.isnull().sum() #Is there any Null or Empty cell presents
print(new_df.sample(2)) # Checking the random dataset sample
new_df = new_df[['Engine HP','MSRP']] # We only take the 'Engine HP' and 'MSRP' columns
print(new_df.sample(5))
#Split Train and Test dataset
X = np.array(new_df[['Engine HP']]) # Storing into X the 'Engine HP' as np.array
y = np.array(new_df[['MSRP']]) # Storing into y the 'MSRP' as np.array
plt.scatter(X,y,color="red") # Plot a graph X vs y
plt.title('HP vs MSRP')
plt.xlabel('HP')
plt.ylabel('MSRP')
plt.show()
```

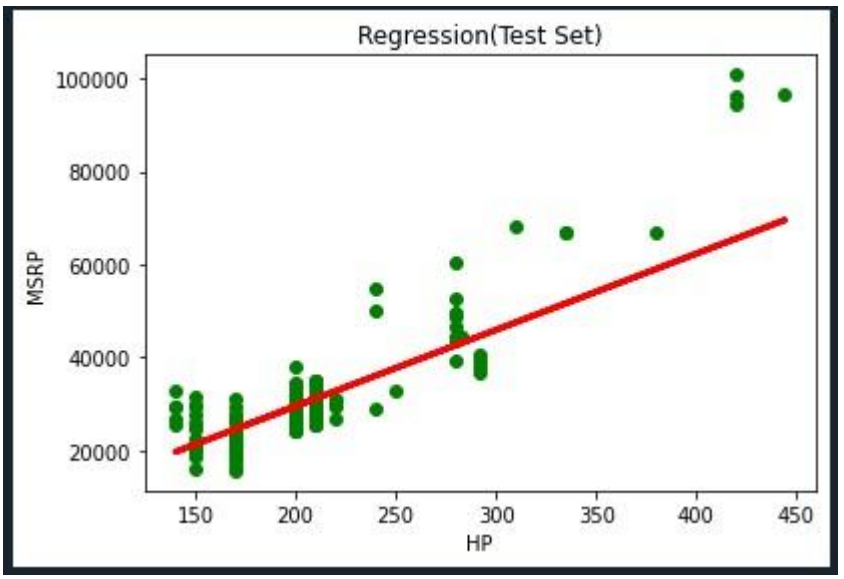
```
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size = 0.25,random_state=15)
regressor = LinearRegression()
regressor.fit(X_train,y_train)

plt.scatter(X_test,y_test,color="green") # Plot a graph with X_test vs y_test
plt.plot(X_train,regressor.predict(X_train),color="red",linewidth=3)
plt.title('Regression(Test Set)')
plt.xlabel('HP')
plt.ylabel('MSRP')
plt.show()

plt.scatter(X_train,y_train,color="blue")  # Plot a graph with X_train vs y_train
plt.plot(X_train,regressor.predict(X_train),color="red",linewidth=3)
plt.title('Regression(training Set)')
plt.xlabel('HP')
plt.ylabel('MSRP')
plt.show()
```

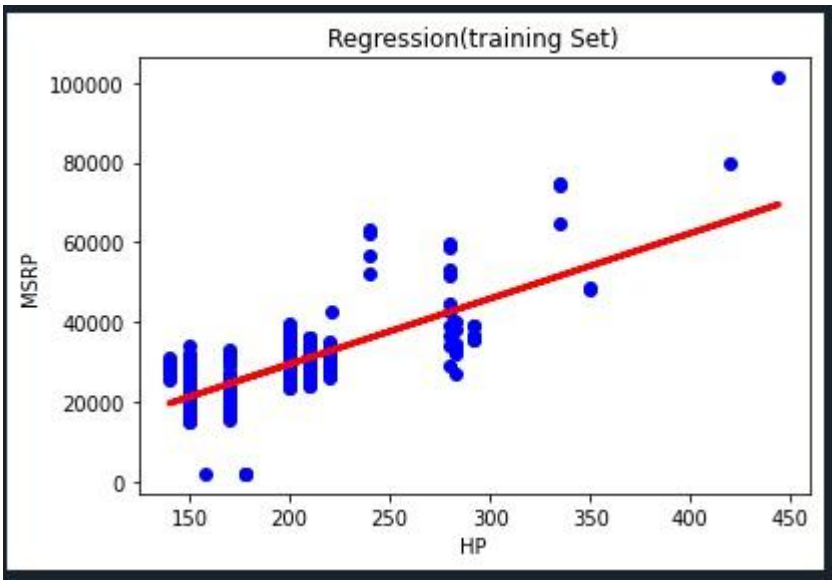
Output:

```
runfile('C:/TYBSC/LinearReg1.py', wdir='C:/TYBSC')
         Make     Model  Year  ... city mpg  Popularity   MSRP
7341     Subaru   Outback  2016  ...       25         640  25295
44         BMW  2 Series  2016  ...       23        3916  34850
4080    Hyundai     Equus  2015  ...       15        1439  61500
10350       GMC   Terrain  2016  ...       22         549  23975
9977   Cadillac       SRX  2016  ...       17        1624  48920

[5 rows x 16 columns]
          Make Model  Year  ... city mpg  Popularity   MSRP
5466  Volkswagen  Golf  2017  ...       25         873  20995
5675  Volkswagen   GTI  2013  ...       21         873  28795

[2 rows x 16 columns]
      Engine HP   MSRP
1841      210.0  29895
10530     280.0  39300
5325      200.0  25375
1835      210.0  30995
6059      140.0  28390
```


HP vs MSRP
MSRP
HP
100000
80000
60000
40000
20000
0
150
200
250
300
350
400
450

OBJ


Regression(Test Set)
MSRP
HP
100000
80000
60000
40000
20000
150
200
250
300
350
400
450


Regression(training Set)
MSRP
HP
100000
80000
60000
40000
20000
0
150
200
250
300
350
400
450

OBJ

*****