```
#Program 1 - Read the dataset.

import pandas as pd
import matplotlib.pyplot as plt
df=pd.read_csv('mtcars.csv')
print(df)
```

```
8              Merc 230  22.8    4  140.8   95  3.92  3.150  22.90   1   0
9              Merc 280  19.2    6  167.6  123  3.92  3.440  18.30   1   0
10            Merc 280C  17.8    6  167.6  123  3.92  3.440  18.90   1   0
11           Merc 450SE  16.4    8  275.8  180  3.07  4.070  17.40   0   0
12           Merc 450SL  17.3    8  275.8  180  3.07  3.730  17.60   0   0
13          Merc 450SLC  15.2    8  275.8  180  3.07  3.780  18.00   0   0
14   Cadillac Fleetwood  10.4    8  472.0  205  2.93  5.250  17.98   0   0
15  Lincoln Continental  10.4    8  460.0  215  3.00  5.424  17.82   0   0
16    Chrysler Imperial  14.7    8  440.0  230  3.23  5.345  17.42   0   0
17             Fiat 128  32.4    4   78.7   66  4.08  2.200  19.47   1   1
18          Honda Civic  30.4    4   75.7   52  4.93  1.615  18.52   1   1
19       Toyota Corolla  33.9    4   71.1   65  4.22  1.835  19.90   1   1
20        Toyota Corona  21.5    4  120.1   97  3.70  2.465  20.01   1   0
21     Dodge Challenger  15.5    8  318.0  150  2.76  3.520  16.87   0   0
22          AMC Javelin  15.2    8  304.0  150  3.15  3.435  17.30   0   0
23           Camaro Z28  13.3    8  350.0  245  3.73  3.840  15.41   0   0
24     Pontiac Firebird  19.2    8  400.0  175  3.08  3.845  17.05   0   0
25            Fiat X1-9  27.3    4   79.0   66  4.08  1.935  18.90   1   1
26        Porsche 914-2  26.0    4  120.3   91  4.43  2.140  16.70   0   1
27         Lotus Europa  30.4    4   95.1  113  3.77  1.513  16.90   1   1
28       Ford Pantera L  15.8    8  351.0  264  4.22  3.170  14.50   0   1
29         Ferrari Dino  19.7    6  145.0  175  3.62  2.770  15.50   0   1
30        Maserati Bora  15.0    8  301.0  335  3.54  3.570  14.60   0   1
31           Volvo 142E  21.4    4  121.0  109  4.11  2.780  18.60   1   1

    gear  carb
0      4     4
1      4     4
2      4     1
3      3     1
4      3     2
5      3     1
6      3     4
7      4     2
8      4     2
9      4     4
10     4     4
11     3     3
12     3     3
13     3     3
14     3     4
15     3     4
16     3     4
17     4     1
18     4     2
19     4     1
20     3     1
21     3     2
22     3     2
23     3     4
24     3     2
25     4     1
26     5     2
27     5     2
28     5     4
29     5     6
30     5     8
31     4     2
```

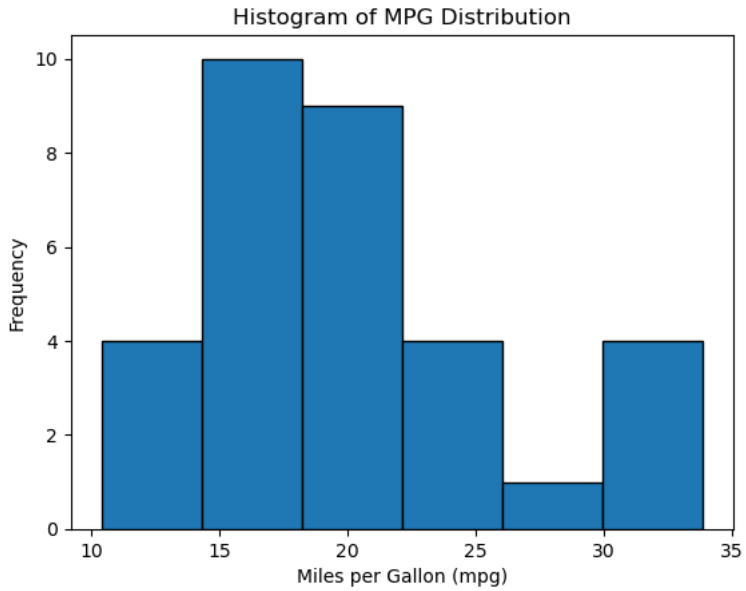
```
#program 2 - Find the head of the dataset.

df.head()
```

	model	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
0	Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
1	Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
2	Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
3	Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
4	Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2

```
#Program 3 - Find the Datatype of Dataset (each column).
datatypes=df.dtypes
print(datatypes)
```

```
model     object
mpg      float64
cyl        int64
disp     float64
hp         int64
drat     float64
wt       float64
qsec     float64
vs         int64
am         int64
gear       int64
carb       int64
dtype: object
```

```
#Program 4 -  From the given dataset "mtcars.csv" plot a histogram
# to check the frequency distribution of the variable "mpg" (Miles per gallon).
# Extract the 'mpg' column
mpg_data = df['mpg']

# Plot the histogram
plt.hist(mpg_data, bins='auto', edgecolor='black')

# Set labels and title
plt.xlabel('Miles per Gallon (mpg)')
plt.ylabel('Frequency')
plt.title('Histogram of MPG Distribution')

# Display the histogram
plt.show()
```




```
#Program 5 -  Find the highest frequency of interval.

mpg_data=df['mpg']
print(mpg_data.mode)
```

```
<bound method Series.mode of 0     21.0
1     21.0
2     22.8
3     21.4
4     18.7
5     18.1
6     14.3
7     24.4
8     22.8
9     19.2
10    17.8
11    16.4
12    17.3
13    15.2
14    10.4
15    10.4
16    14.7
17    32.4
18    30.4
19    33.9
20    21.5
21    15.5
22    15.2
```

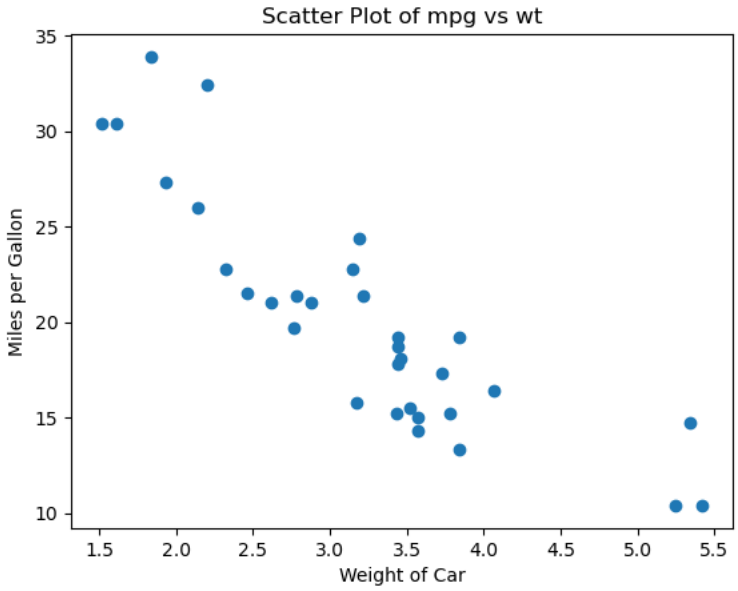
```
23    13.3
24    19.2
25    27.3
26    26.0
27    30.4
28    15.8
29    19.7
30    15.0
31    21.4
Name: mpg, dtype: float64>
```

```
# Find the highest frequency of interval.
# most frequent value in Team
df['mpg'].value_counts().idxmax()
```

```
21.0
```

```
#Program 6

#Which can be inferred from scatter plot of mpg(Miles per gallon) vs

# wt(Weight of car) from the dataset mtcars.csv.


"""Negative correlation: The scatter plot shows a negative relationship between "mpg" and "wt."
As the weight of the car increases, the miles per gallon tends to decrease.
This suggests that lighter cars tend to have better fuel efficiency."""


# Calculate the correlation between "mpg" and "wt"
correlation = df['mpg'].corr(df['wt'])

# Check if the correlation is negative
if correlation < 0:
    print("There is a negative correlation between 'mpg' and 'wt' in the mtcars dataset.")
else:
    print("There is no negative correlation between 'mpg' and 'wt' in the mtcars dataset.")


# Create the scatter plot
plt.scatter(df['wt'], df['mpg'])
plt.xlabel('Weight of Car')
plt.ylabel('Miles per Gallon')

# Set title and display the plot
plt.title('Scatter Plot of mpg vs wt')
plt.show()
```

```
There is a negative correlation between 'mpg' and 'wt' in the mtcars dataset.
```




```
"""Outliers: The scatter plot may reveal outliers, which are data points that
deviate significantly from the general trend.
Outliers could represent cars with unusual characteristics,
such as exceptionally low or high fuel efficiency given their weight."""
```

```
# Calculate z-scores for "mpg" and "wt"
z_scores_mpg = (df['mpg'] - df['mpg'].mean()) / df['mpg'].std()
z_scores_wt = (df['wt'] - df['wt'].mean()) / df['wt'].std()

# Set the threshold for outlier detection
z_score_threshold = 2

# Find outliers
outliers_mpg = df[z_scores_mpg.abs() > z_score_threshold]
outliers_wt = df[z_scores_wt.abs() > z_score_threshold]

# Create the scatter plot
plt.scatter(df['wt'], df['mpg'])
plt.xlabel('Weight of Car')
plt.ylabel('Miles per Gallon')

# Plot the outliers
plt.scatter(outliers_wt['wt'], outliers_wt['mpg'], color='red', label='Outliers (wt)')
plt.scatter(outliers_mpg['wt'], outliers_mpg['mpg'], color='blue', label='Outliers (mpg)')

# Set labels and title
plt.xlabel('Weight of Car')
plt.ylabel('Miles per Gallon')
plt.title('Scatter Plot of mpg vs wt')
plt.legend()

# Display the plot
plt.show()
```