Untitled104

June 24, 2023

[2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

1 Module 6: Data Visualization

with Matplotlib Case Study Contact us: support@intellipaat.com / © Copyright Intellipaat / All rights reserved Intel iPaat Python for Data Science Certification Course Problem Statement: You work in XYZ Company as a Python developer. The company officials want you to build a python program. Dataset Link Tasks To Be Performed: 1. Load cars data as dataframe using pandas and create a bar plot between number of cylinders and frequency of cars with that many number of cylinders Set xlabel as Number of cylinders Set ylabel as Frequency of cars Draw a bar plot 2. Write code to load data from cars and print a bar graph of count of columns with null values. 3. Use the 'mpg' (Miles Per Gallon column) and draw a histogram Set xlabel: Miles per gallon Set ylabel: Frequency Set title as Miles Per Gallon Histogram Use mpg column to generate a histogram 4. Draw a boxplot on the card dataframes hp column Set xlabel: Car Horsepower Set title as Boxplot for car horsepower Use hp column to generate a boxplot

[14]:
```
data=pd.read_csv('C:/Users/Vikas/Desktop/cars-6.csv')
```

[15]:
```
data
```

[15]:
```
                 model   mpg  cyl   disp   hp  drat     wt   qsec  vs  am  \
0            Mazda RX4  21.0    6  160.0  110  3.90  2.620  16.46   0   1
1        Mazda RX4 Wag  21.0    6  160.0  110  3.90  2.875  17.02   0   1
2           Datsun 710  22.8    4  108.0   93  3.85  2.320  18.61   1   1
3       Hornet 4 Drive  21.4    6  258.0  110  3.08  3.215  19.44   1   0
4    Hornet Sportabout  18.7    8  360.0  175  3.15  3.440  17.02   0   0
5              Valiant  18.1    6  225.0  105  2.76  3.460  20.22   1   0
6           Duster 360  14.3    8  360.0  245  3.21  3.570  15.84   0   0
7            Merc 240D  24.4    4  146.7   62  3.69  3.190  20.00   1   0
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
```

[5]:
```
data.head(10)
```

[5]:
```
                 model   mpg  cyl   disp   hp  drat     wt   qsec  vs  am  gear  \
0            Mazda RX4  21.0    6  160.0  110  3.90  2.620  16.46   0   1     4
1        Mazda RX4 Wag  21.0    6  160.0  110  3.90  2.875  17.02   0   1     4
2           Datsun 710  22.8    4  108.0   93  3.85  2.320  18.61   1   1     4
3       Hornet 4 Drive  21.4    6  258.0  110  3.08  3.215  19.44   1   0     3
4    Hornet Sportabout  18.7    8  360.0  175  3.15  3.440  17.02   0   0     3
5              Valiant  18.1    6  225.0  105  2.76  3.460  20.22   1   0     3
6           Duster 360  14.3    8  360.0  245  3.21  3.570  15.84   0   0     3
7            Merc 240D  24.4    4  146.7   62  3.69  3.190  20.00   1   0     4
8             Merc 230  22.8    4  140.8   95  3.92  3.150  22.90   1   0     4
9             Merc 280  19.2    6  167.6  123  3.92  3.440  18.30   1   0     4

   carb
0     4
1     4
2     1
3     1
4     2
5     1
6     4
7     2
8     2
9     4
```

[6]:
```
data.columns
```

[6]:
```
Index(['model', 'mpg', 'cyl', 'disp', 'hp', 'drat', 'wt', 'qsec', 'vs', 'am',
       'gear', 'carb'],
      dtype='object')
```

[8]:
```
data.describe().T
```

[8]:
```
      count        mean         std     min        25%      50%     75%  \
mpg    32.0   20.090625    6.026948  10.400   15.42500   19.200   22.80
cyl    32.0    6.187500    1.785922   4.000    4.00000    6.000    8.00
disp   32.0  230.721875  123.938694  71.100  120.82500  196.300  326.00
hp     32.0  146.687500   68.562868  52.000   96.50000  123.000  180.00
drat   32.0    3.596563    0.534679   2.760    3.08000    3.695    3.92
```

```
wt     32.0    3.217250    0.978457   1.513    2.58125    3.325    3.61
qsec   32.0   17.848750    1.786943  14.500   16.89250   17.710   18.90
vs     32.0    0.437500    0.504016   0.000    0.00000    0.000    1.00
am     32.0    0.406250    0.498991   0.000    0.00000    0.000    1.00
gear   32.0    3.687500    0.737804   3.000    3.00000    4.000    4.00
carb   32.0    2.812500    1.615200   1.000    2.00000    2.000    4.00

          max
mpg    33.900
cyl     8.000
disp  472.000
hp    335.000
drat    4.930
wt      5.424
qsec   22.900
vs      1.000
am      1.000
gear    5.000
carb    8.000
```

[9]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32 entries, 0 to 31
Data columns (total 12 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   model   32 non-null     object
 1   mpg     32 non-null     float64
 2   cyl     32 non-null     int64
 3   disp    32 non-null     float64
 4   hp      32 non-null     int64
 5   drat    32 non-null     float64
 6   wt      32 non-null     float64
 7   qsec    32 non-null     float64
 8   vs      32 non-null     int64
 9   am      32 non-null     int64
 10  gear    32 non-null     int64
 11  carb    32 non-null     int64
dtypes: float64(5), int64(6), object(1)
memory usage: 3.1+ KB
```

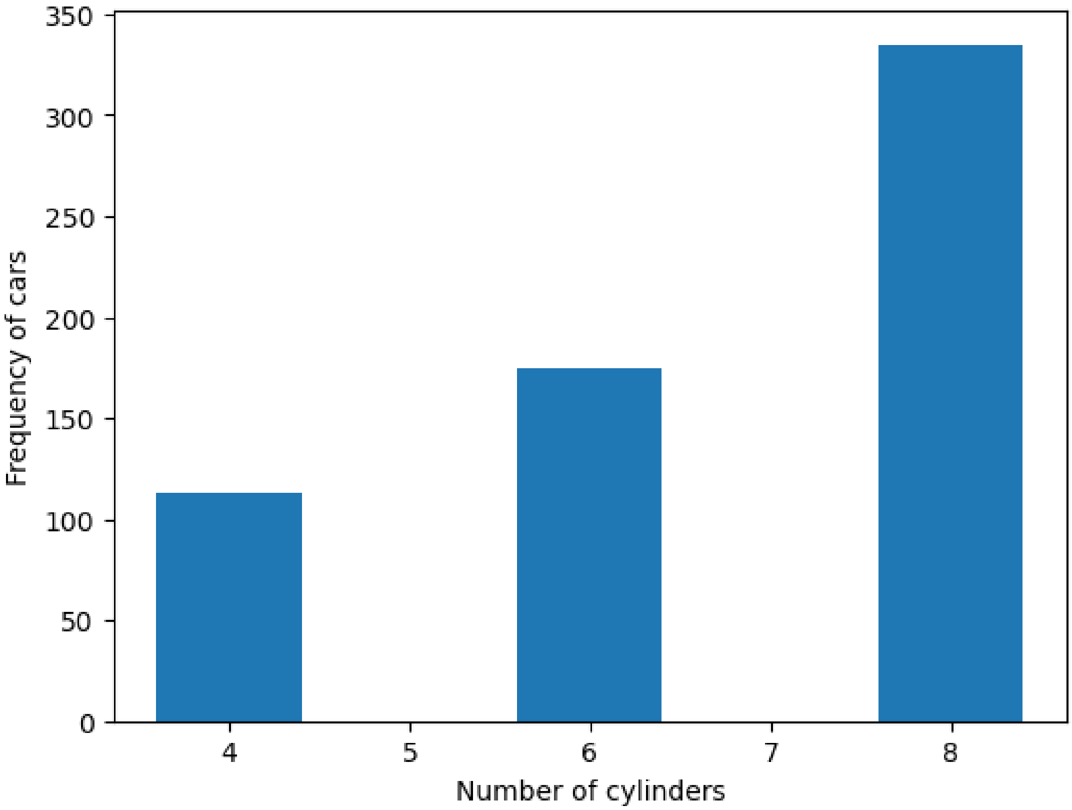
[23]:

```
sns.barplot(data=data,x='cyl',values_counts())
plt.xlabel('Number of cylinders ')
plt.ylabel('Frequency of cars ')
plt.show()
```

```
  Cell In[23], line 1
```

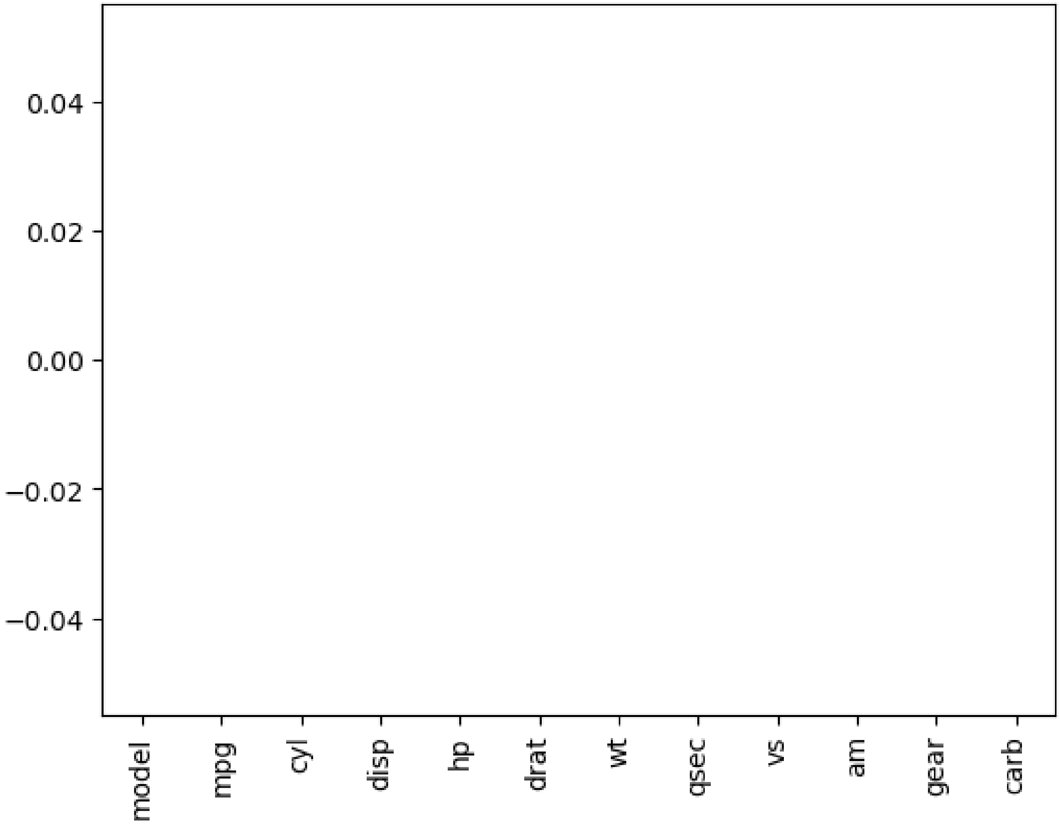
```
    sns.barplot(data=data,x='cyl',values_counts())
                                                 ^

SyntaxError: positional argument follows keyword argument
```

[24]:
```
plt.bar(data['cyl'], data['hp'])
plt.xlabel('Number of cylinders')
plt.ylabel('Frequency of cars')
plt.show()
```



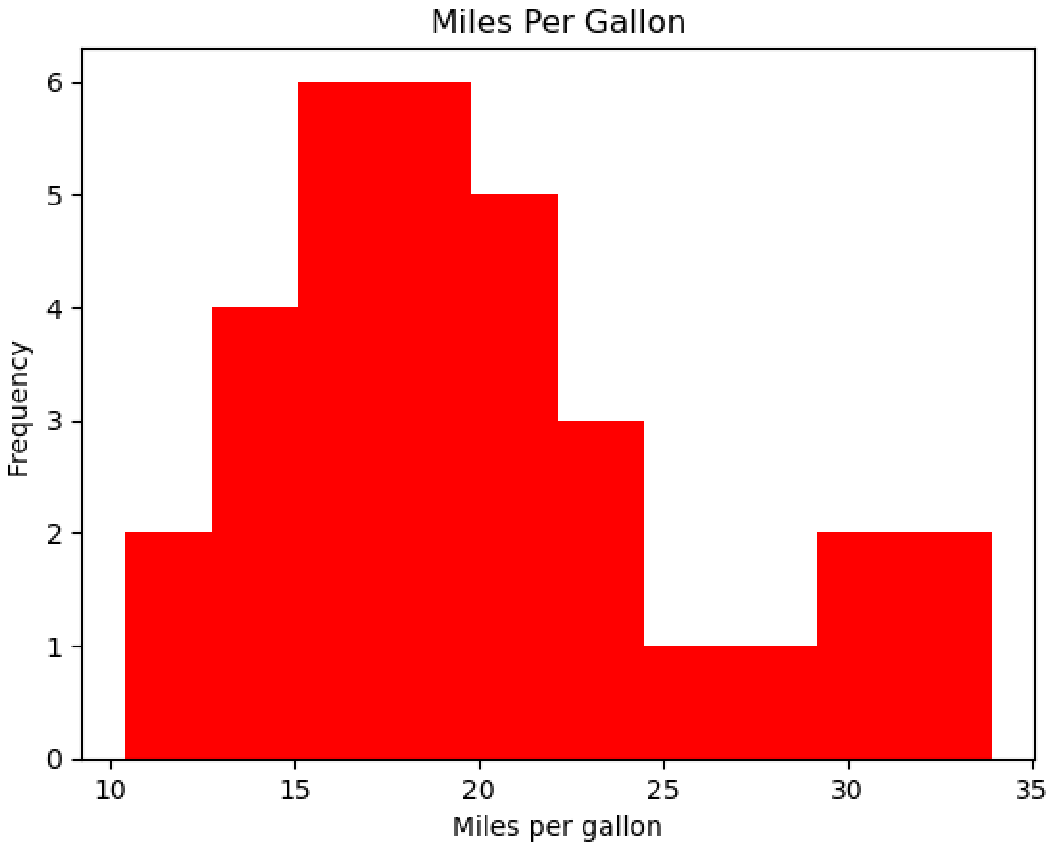
[26]:
```
##2
null_counts = data.isnull().sum()
null_counts.plot(kind='bar')
plt.show
```

[26]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



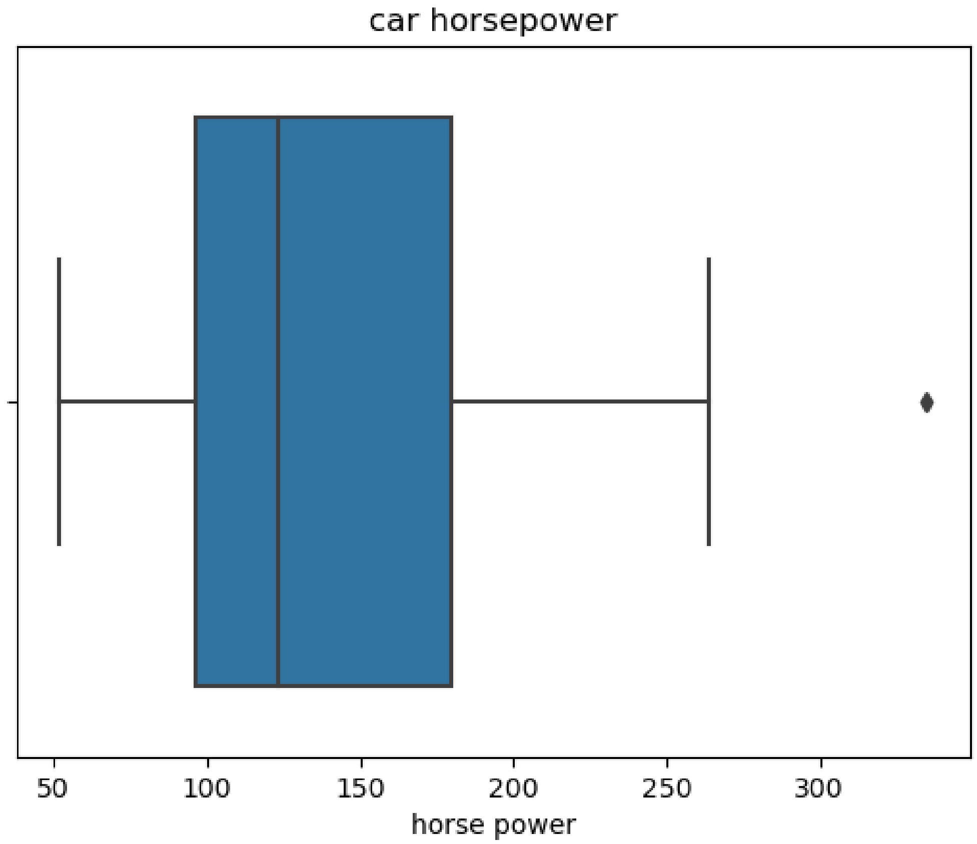
```
[36]: plt.hist(data['mpg'],color='r')
      plt.xlabel('Miles per gallon')
      plt.ylabel('Frequency ')
      plt.title('Miles Per Gallon')
      plt.show
```

```
[36]: <function matplotlib.pyplot.show(close=None, block=None)>
```



[33]:
```
##4
sns.boxplot(x=data['hp'])
plt.xlabel('horse power')
plt.title('car horsepower')

plt.show
```

[33]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```


car horsepower
50
100
150
200
250
300
horse power

[ ]: