```
#importing libaries
import numpy as np
import scipy as sp
import pandas as pd
import matplotlib as mpl
```

```
#Display the first five rows of the data.
df = pd.read_csv('mtcars.csv')
df.head()
```

	rownames	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

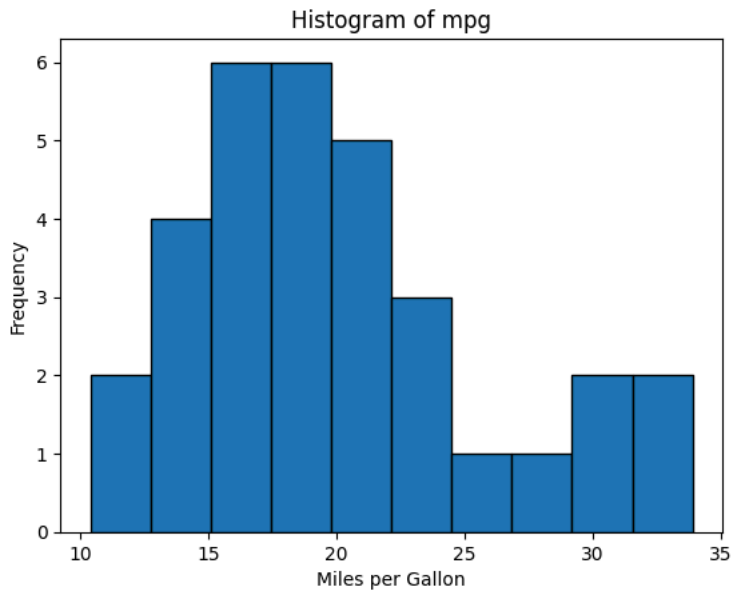
Next steps: Generate code with df | View recommended plots | New interactive sheet

```
#For each category in the cyl column, calculate the average mpg for all cars with that cyl value.
df.groupby('cyl')['mpg'].mean()
```

cyl	mpg
4	26.663636
6	19.742857
8	15.100000

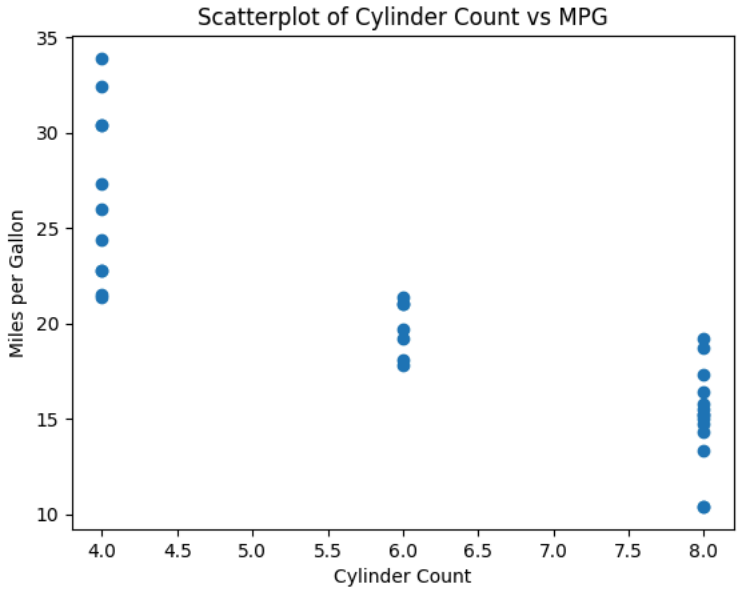
dtype: float64

```
#Create a histogram using the mpg column.
import matplotlib.pyplot as plt
plt.hist(df['mpg'], edgecolor = 'black', linewidth = 1)
plt.xlabel('Miles per Gallon')
plt.ylabel('Frequency')
plt.title('Histogram of mpg')
plt.show()
```




```
#Choose two variables in the data and create a scatterplot.
#decided to use cyl for x and mpg for y
```

```
plt.scatter(df['cyl'], df['mpg'])
plt.xlabel('Cylinder Count')
plt.ylabel('Miles per Gallon')
plt.title('Scatterplot of Cylinder Count vs MPG')
plt.show()
#End part A
```



```
#importing libaries
import numpy as np
import scipy as sp
import pandas as pd
import matplotlib as mpl
import matplotlib.pyplot as plt
```

```
# Find a tabular dataset that interests you that has "tidy" data.
#Tidy data has data that is ready for your data analysis.
#For our tasks we want something where columns represent variables and rows represent unique observations.
#Give a brief description of the dataset.  Provide a citation of the dataset (any format is fine).
#Displaying the dataset for Part B, which details price data for the Nvidia Corporation between August 12th, 2024 and September 10th, 2024.
#variables include Date, Close/Last (price), Open (price), High (price), and Low (price), 22 observations.
#The dataset is updated daily, so although you can access the dataset from the Nasdaq stock exchange historical data section,
#https://www.nasdaq.com/market-activity/stocks/nvda/historical?page=1&rows_per_page=10&timeline=m1 it will add days as time goes on.
#As a result, a file for the dataset used at the time of the making of this notebook will be attached separately.
```

```
#Display the first five rows of the data.
df = pd.read_csv('HistoricalData_1726087540117.csv')
df.head()
```

	Date	Close/Last	Volume	Open	High	Low
0	9/10/2024	$108.10	268283700	$107.81	$109.40	$104.95
1	9/9/2024	$106.47	273912000	$104.88	$106.55	$103.69
2	9/6/2024	$102.83	413638100	$108.04	$108.15	$100.95
3	9/5/2024	$107.21	306850700	$104.99	$109.65	$104.76
4	9/4/2024	$106.21	372470300	$105.41	$113.27	$104.12

```
#Create a visualization using one or two variables from this data.
#Line plot using Close Price for x and Trade Volume for y
plt.plot(df['Close/Last'], df['Volume'])
plt.xlabel('Close Price')
plt.ylabel('Volume')
plt.title('Plot of Close Prices and Trade Volume')
#adjusting the labels
plt.xticks(rotation = 45)
plt.show()
#This line plot tells us that at certain close prices levels traders may buy or abstain from buying.
#This is shown by the various peaks and troughs.
#Essentially, low support vs high support for the stock at certain price levels.
#End Part B
```


Plot of Close Prices and Trade Volume
1e8
Volume
4.5
4.0
3.5
3.0
2.5
$108.10
$106.47
$102.83
$107.21
$106.21
$108.00
$119.37
$117.59
$125.61
$128.30
$126.46
$129.37
$123.74
$128.50
$127.25
$130.00
$124.58
$122.86
$118.08
$116.14
$109.02
Close Price