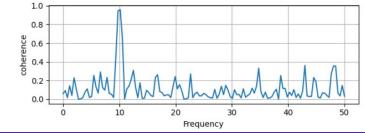
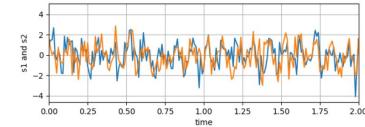
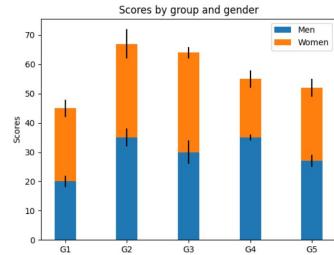


# matplotlib



Data Visualization Library

<https://matplotlib.org/>

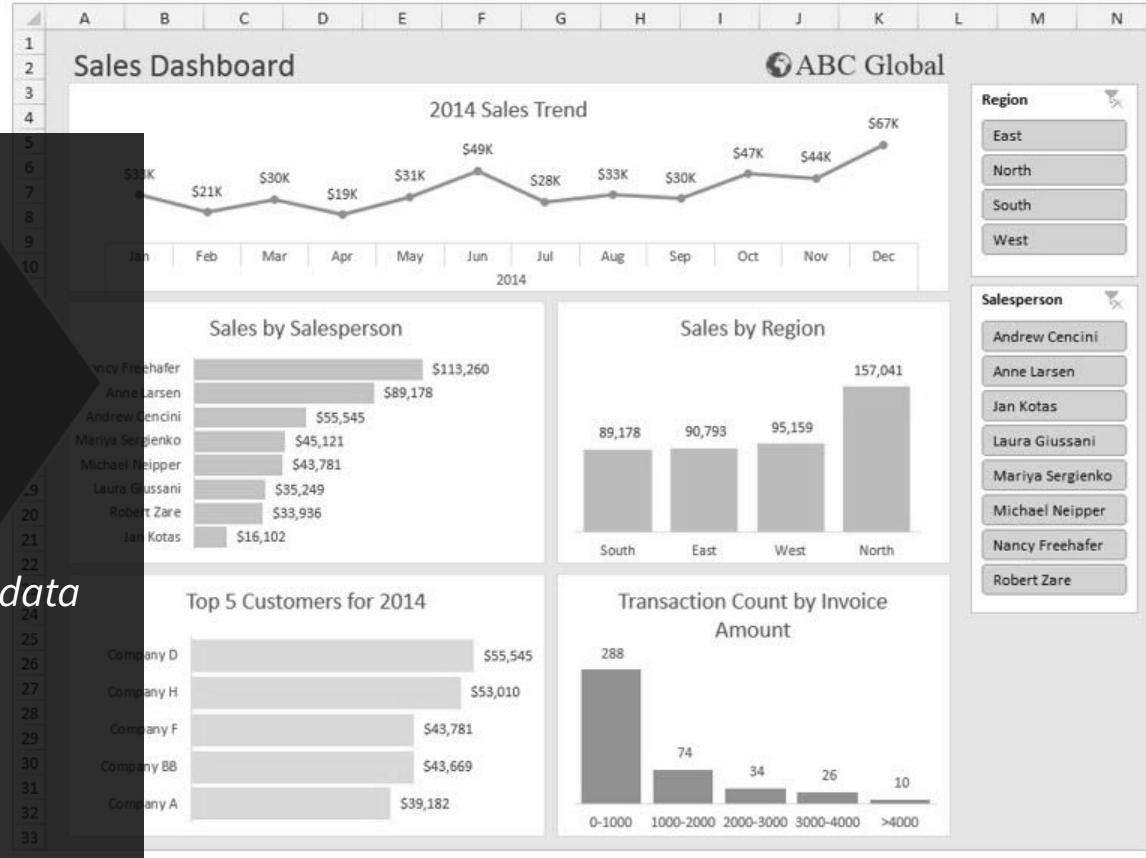


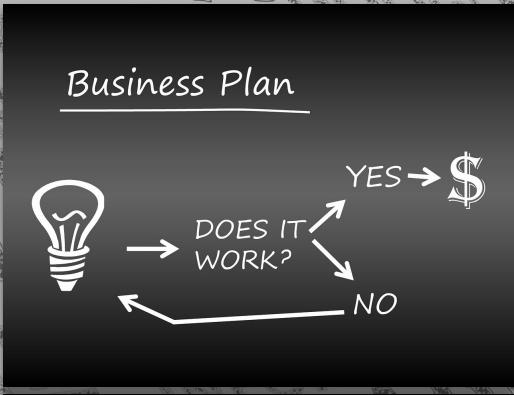
Nachiketh

# What is Data Visualization ?

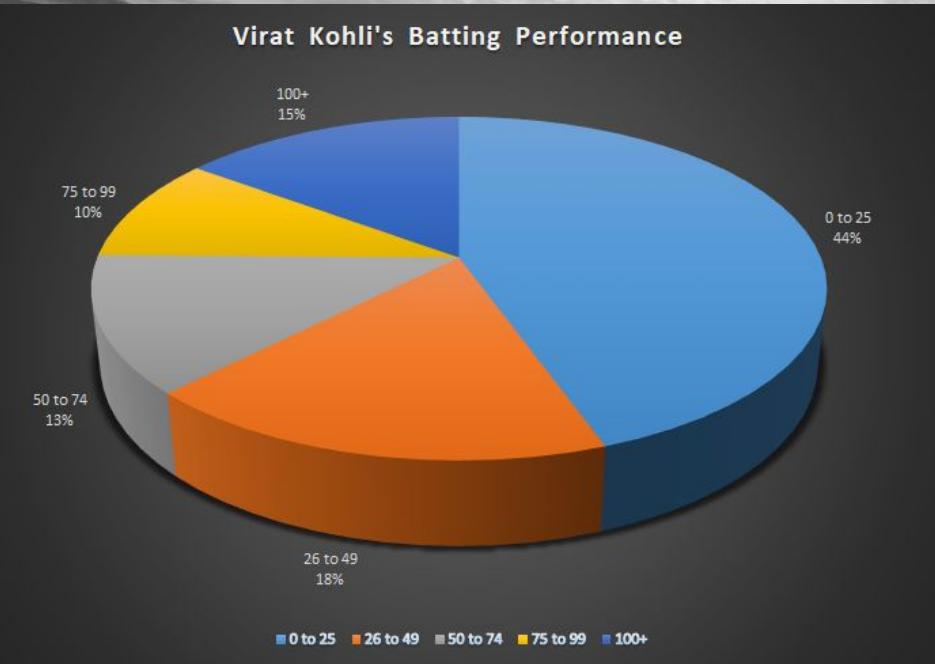
*Data visualization is the graphical representation of information and data with images.*

A	B	C	D	E		
3	Order	Order Da	Customer	Customer Nan	Address	City
4	1015	01/27/14	27	Company AA	789 27th Street	Las V
5	1015	01/27/14	27	Company AA	789 27th Street	Las V
6	1016	01/04/14	4	Company D	123 4th Street	New
7	1016	01/04/14	4	Company D	123 4th Street	New
8	1016	01/04/14	4	Company D	123 4th Street	New
9	1017	01/12/14	12	Company L	123 12th Street	Las V
10	1017	01/12/14	12	Company L	123 12th Street	Las V
11	1018	01/08/14	8	Company H	123 8th Street	Portl
12	1019	01/04/14	4	Company D	123 4th Street	Portl
13	1020	01/04/14	1	Company C	123 1st Street	Seat
14	1021	01/04/14	6	Company F	123 6th Street	Seat
15	1022	01/06/14	28	Company BB	789 28th Street	Mem
16	1023	01/20/14	28	Company BB	789 28th Street	Mem
17	1024	01/21/14	10	Company J	123 10th Street	Salt I
18	1025	01/21/14	10	Company J	123 10th Street	Salt I
19	1026	01/07/14	7	Company G	123 7th Street	Chic
20	1027	01/10/14	10	Company J	123 10th Street	Chic
21	1027	01/10/14	10	Company J	123 10th Street	Chic
22	1027	01/10/14	10	Company J	123 10th Street	Chic
23	1028	01/11/14	11	Company K	123 11th Street	Mian
24	1028	01/11/14	1	Company A	123 1st Street	Seat
25	1029	01/01/14	1	Company A	123 1st Street	Seat
26	1029	01/01/14	1	Company BB	789 28th Street	Mem
27	1030	01/28/14	28	Company BB	789 28th Street	Mem
28	1030	01/28/14	28	Company BB	789 28th Street	Mem
29	1031	01/09/14	9	Company I	123 9th Street	Salt I
30	1031	01/09/14	9	Company I	123 9th Street	Salt I
31	1031	01/09/14	9	Company I	123 9th Street	Salt I
32	1032	01/06/14	6	Company F	123 6th Street	Milw
33	1033	02/08/14	8	Company H	123 8th Street	Portl
34	1034	02/03/14	3	Company C	123 3rd Street	Los A
35	1034	02/03/14	3	Company C	123 3rd Street	Los A



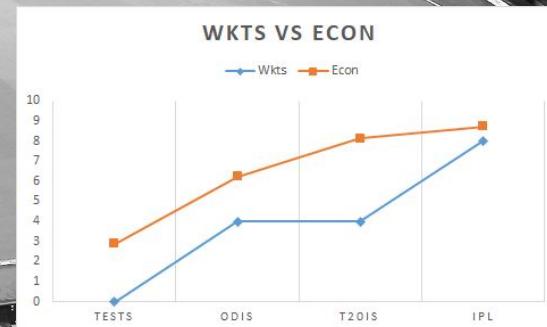
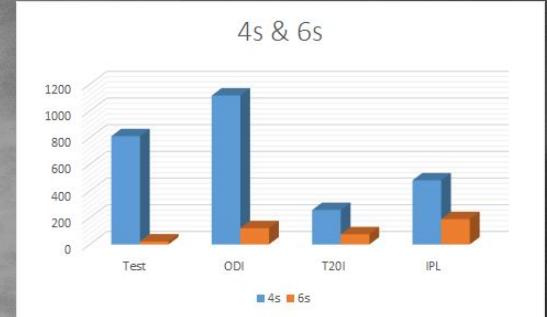
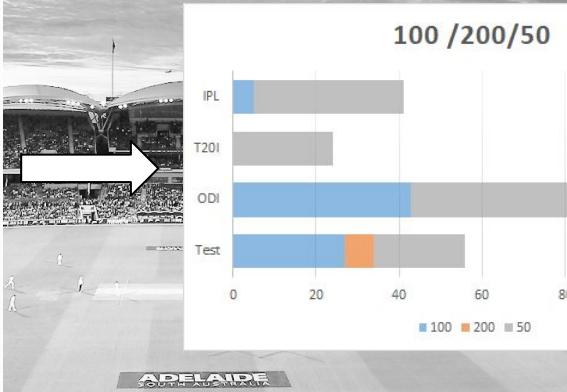


# Example for Visualization



# Example for Visualization

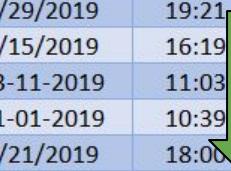
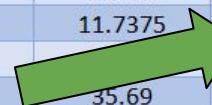
14	14	3	<b>593</b>	100*	53.90	438	135.38	1	6	49	25
16	16	1	<b>529</b>	81	35.26	398	132.91	0	4	45	25
16	16	1	<b>521</b>	97*	34.73	384	135.67	0	5	64	11
14	13	4	<b>510</b>	80*	56.66	249	204.81	0	4	31	52
13	13	1	<b>490</b>	99*	40.83	319	153.60	0	4	45	34
16	16	3	<b>488</b>	78*	37.53	300	162.66	0	3	37	27
14	14	0	<b>464</b>	100	33.14	328	141.46	1	2	46	13
16	16	1	<b>463</b>	67	30.86	386	119.94	0	3	41	14
10	10	2	<b>445</b>	114	55.62	283	157.24	1	2	48	18
13	13	3	<b>442</b>	82*	44.20	287	154.00	0	5	31	26
16	15	2	<b>424</b>	71*	32.61	324	130.86	0	2	45	10
15	12	7	<b>416</b>	84*	83.20	309	134.62	0	3	22	23
13	13	0	<b>405</b>	82	31.15	290	139.65	0	4	41	22
15	15	1	<b>405</b>	67	28.92	315	128.57	0	2	52	10

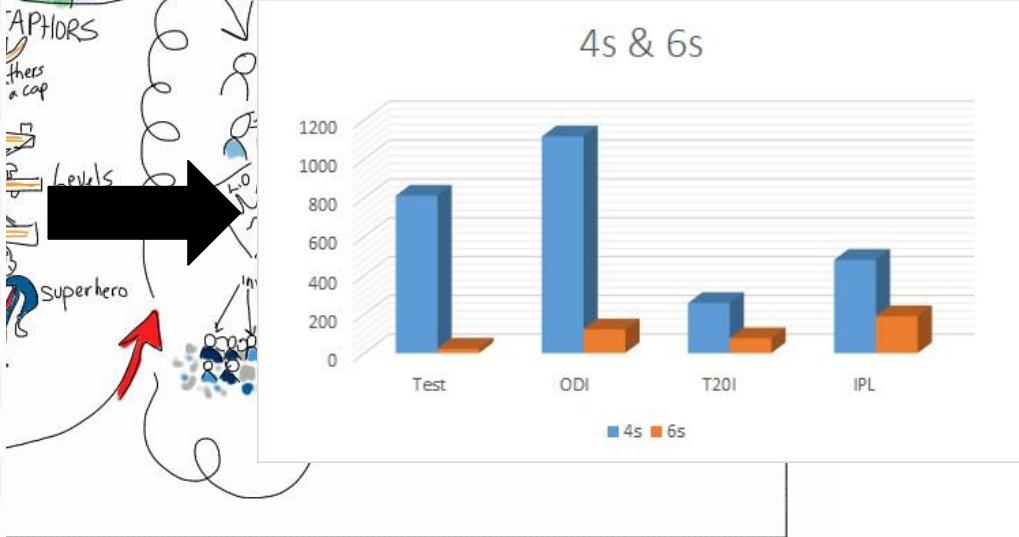
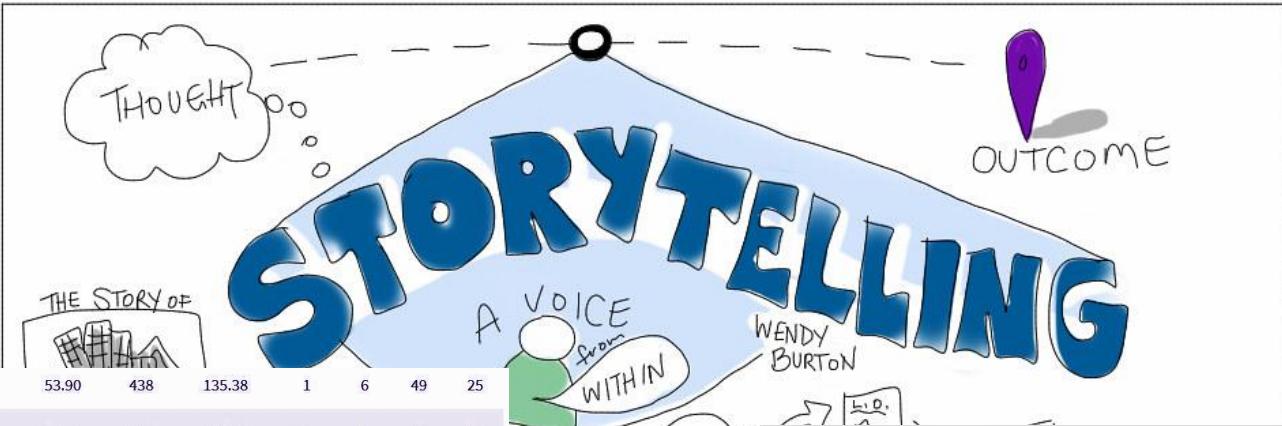


Gender	Product line	Unit price	Quantity	Tax %	Total	Date	Time	Payer	cogs
Female	Health and beauty	74.69	7	26.1415	548.971	02-06-2019	12:51	Cash	50.02
Female	Electronic accessories	15.28	5	3.82	80.22	02-06-2019	12:51	Cash	50.02
Male	Home and lifestyle	46.33	7	16.2155	340.525	02-06-2019	12:51	Cash	50.02
Male	Health and beauty	58.22	8	23.288	489.048	02-06-2019	12:51	Cash	50.02
Male	Sports and travel	86.31	7	30.2085	634.378	02-06-2019	12:51	Cash	50.02
Male	Electronic accessories	85.39	7	29.8865	627.616	02-06-2019	12:51	Cash	50.02
Female	Electronic accessories	68.84	6	20.652	433.692	02-06-2019	12:51	Cash	50.02
Female	Home and lifestyle	73.56	10	36.78	772.38	02-06-2019	12:51	Cash	50.02
Female	Health and beauty	36.26	2	3.626	76.146	02-06-2019	12:51	Cash	50.02
Female	Food and beverages	54.84	3	8.226	172.740	02-06-2019	12:51	Cash	50.02
Female	Fashion accessories	14.48	4	2.896	60.816	02-06-2019	12:51	Cash	50.02
Male	Electronic accessories	25.51	4	5.102	107.14	02-06-2019	12:51	Cash	50.02
Female	Electronic accessories	46.95	5	11.7375	246.487	02-06-2019	12:51	Cash	50.02
Male	Food and beverages	43.19	10	8.638	453.495	02-07-2019	16:48	Ewallet	431.9
Female	Health and beauty	71.38	10	35.69	749.49	02-07-2019	16:48	Ewallet	431.9
		93.72	6	28.116	590.436	02-07-2019	16:48	Ewallet	431.9
		68.93	7	24.1255	506.6355	02-07-2019	16:48	Ewallet	431.9
		72.61	6	21.783	457.443	02-07-2019	16:48	Ewallet	431.9
		54.67	3	8.2005	172.2105	02-07-2019	16:48	Ewallet	431.9
Female	Home and lifestyle	40.3	2	4.03	84.63	02-07-2019	16:48	Ewallet	431.9
Male	Electronics	80.04	5	21.51	451.71	02-07-2019	16:48	Ewallet	431.9
Male	Health and beauty	87.98	3	13.197	277.137	02-07-2019	16:48	Ewallet	431.9
Male	Home and lifestyle	33.2	2	3.32	69.72	02-07-2019	16:48	Ewallet	431.9
Male	Electronic accessories	34.56	5	8.64	181.44	02-07-2019	16:48	Ewallet	431.9
Male	Sports and travel	88.63	3	13.2945	279.1845	02-07-2019	16:48	Ewallet	431.9
Female	Home and lifestyle	52.59	8	21.036	441.756	02-07-2019	16:48	Ewallet	431.9
Male	Fashion accessories	33.52	1	1.676	35.196	02-07-2019	16:48	Ewallet	431.9
Female	Fashion accessories	87.67	2	8.767	184.107	02-07-2019	16:48	Ewallet	431.9
Female	Food and beverages	88.36	5	22.09	463.89	02-07-2019	16:48	Ewallet	431.9

# amazon

## Sales Data





# Advantages of Data Visualization

- Better analysis
- Quick action
- Identifying patterns
- Finding errors
- Understanding the story
- Exploring business insights
- Grasping the Latest Trends

# What is Matplotlib ?

---

Matplotlib is a comprehensive library for creating static plot in Python.

*Installation Instructions:*

```
$ pip install matplotlib
```

*Import Instructions:*

```
import matplotlib as mpl  
import matplotlib.pyplot as plt
```



# Anatomy of a plot

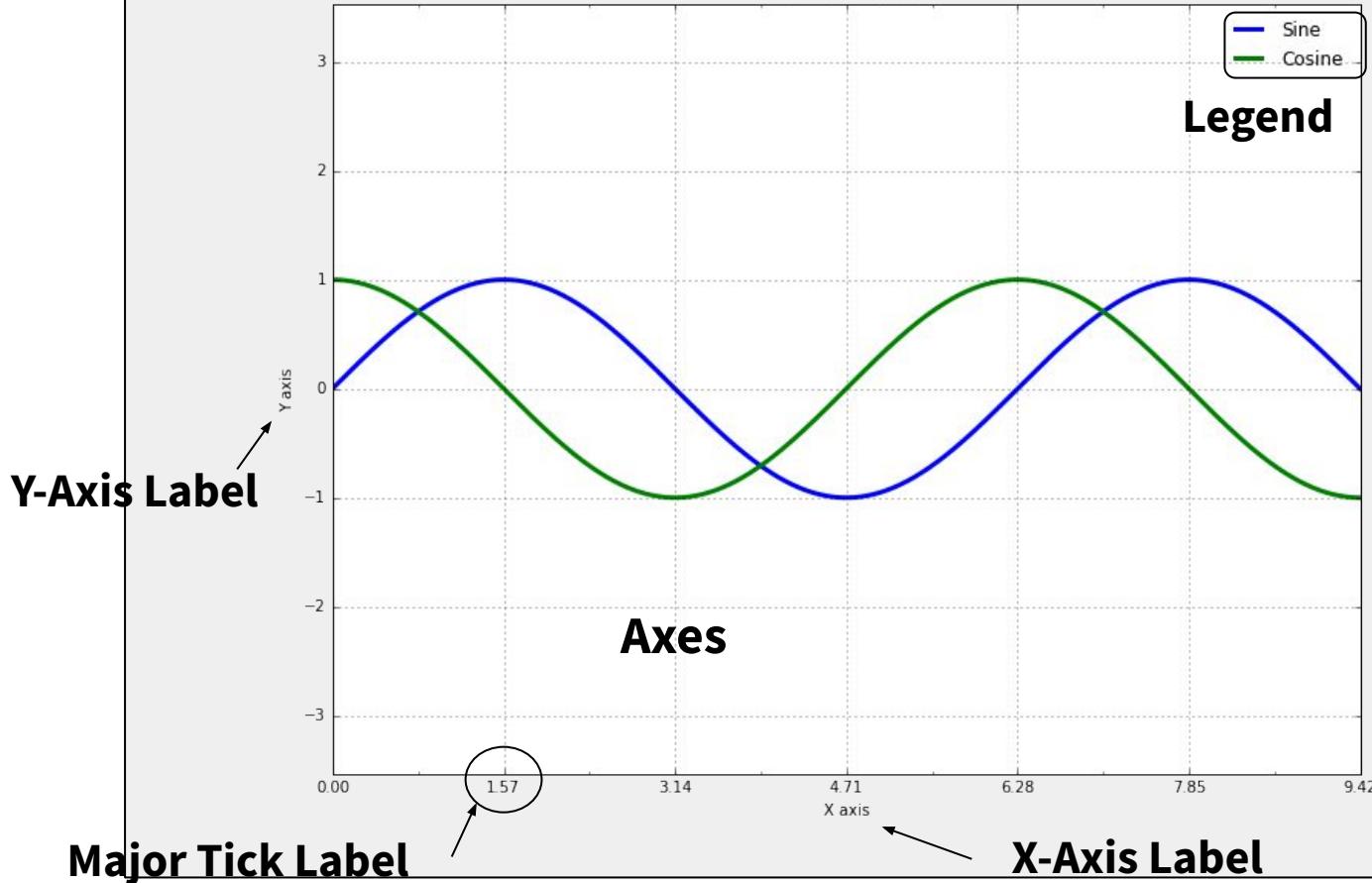


**TV Mounted on a Wall**

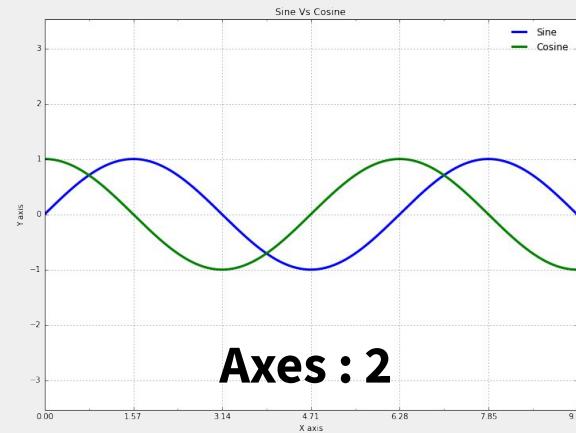
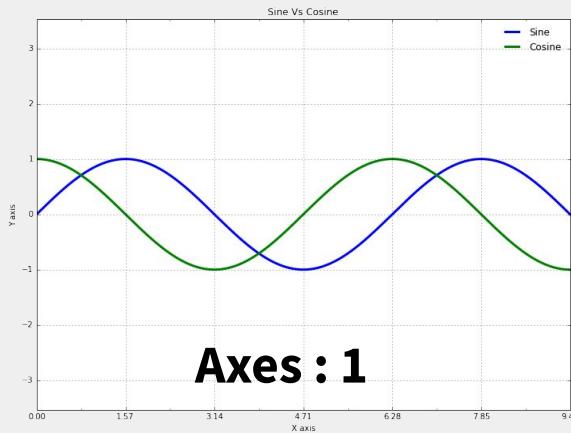
**Figure**

**Title of the Plot**

Sine Vs Cosine



# Figure



# How to create figure & axes



## Figure

```
In [12]: plt.figure()
```

```
Out[12]: <Figure size 432x288 with 0 Axes>
```

```
<Figure size 432x288 with 0 Axes>
```

## Figure and Axes

```
In [13]: plt.figure()  
plt.axes()
```

```
Out[13]: <matplotlib.axes._subplots.AxesSubplot at 0x2a7e0c8e388>
```



### Note:

fig → Refer the **figure** instance

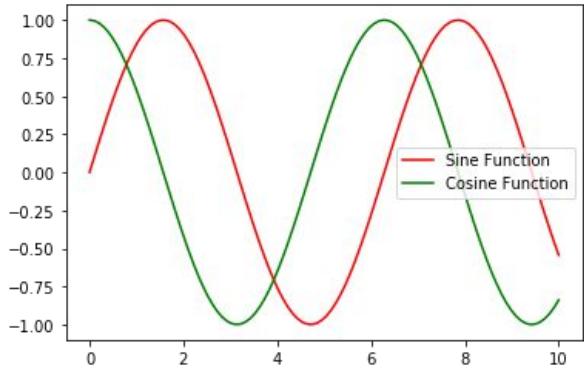
ax → Refer an **axes** instance

# Types of Plot

*Depending on Data  
& Application, we  
can create plots of  
our interest, using  
**matplotlib***

- Line Plots
- Scatter Plots
- Histogram Plots
- Bar Chart
- Box Plot
- Correlation Plot
- Heatmap
- Violin Plot
- Swarm Plot
- Pie Chart, etc.

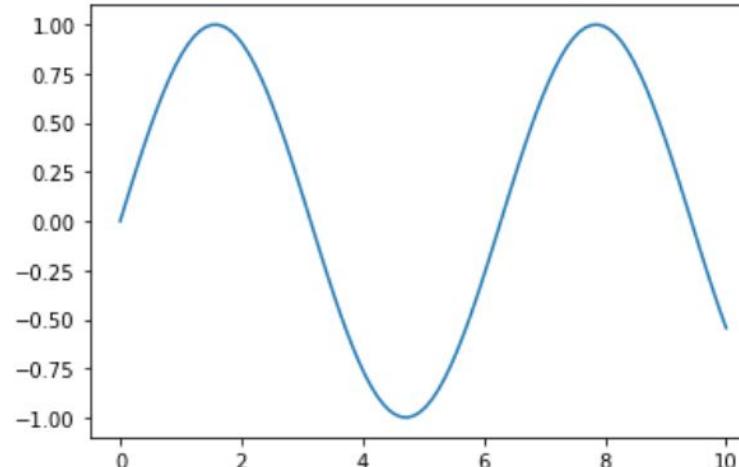
# Line Plots



# Line Plots

```
matplotlib.pyplot.plot(*args, scalex=True, scaley=True, data=None, **kwargs)
```

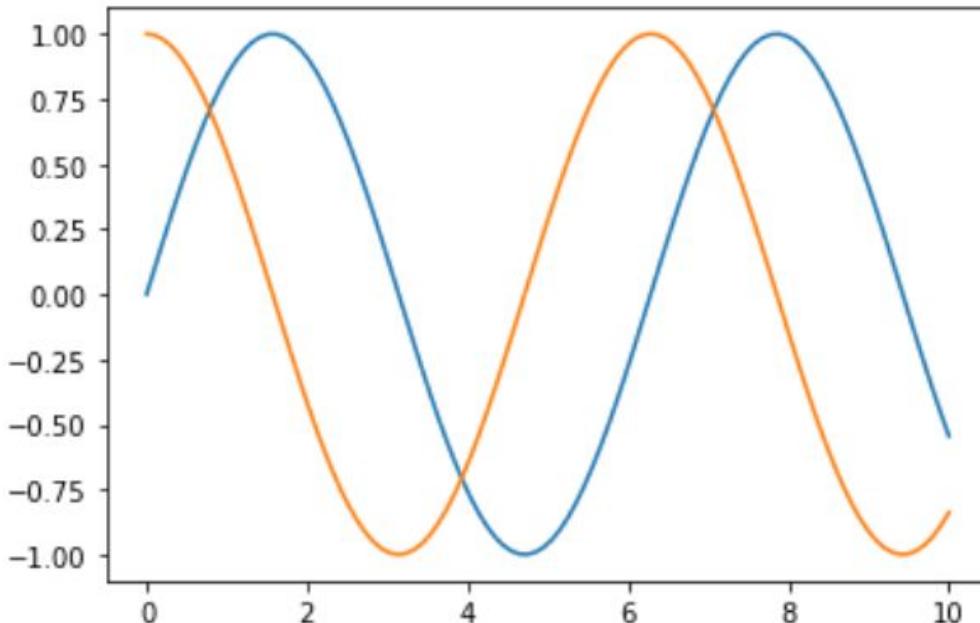
```
In [4]: x = np.linspace(0,10,100)
plt.plot(x,np.sin(x))
plt.show()
```



# Create Multiple Line Plots in single figure

simply call the plot function multiple times

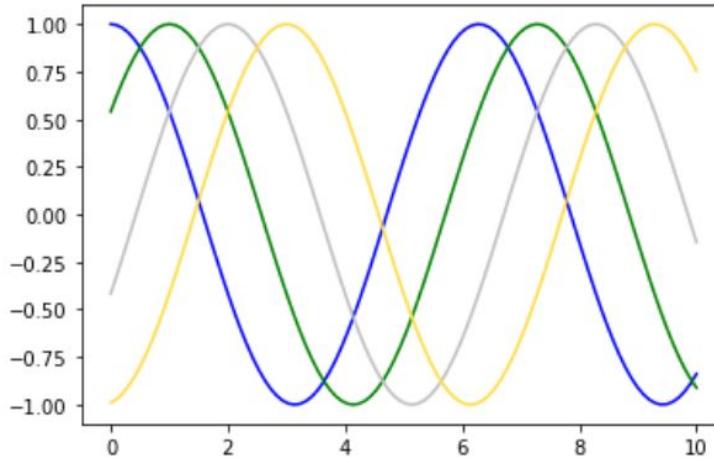
```
plt.plot(x,np.sin(x))  
plt.plot(x,np.cos(x))  
plt.show()
```



# Adjusting the plots : Line Color

Use **color** keyword inside plt.plot() function

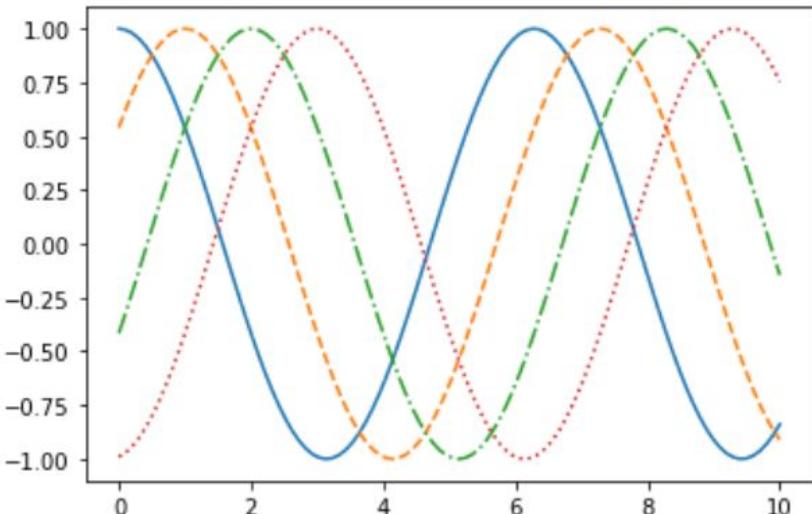
```
plt.plot(x, np.cos(x - 0), color='blue')      # specify color by name
plt.plot(x, np.cos(x - 1), color='g')          # short color code (rgbcmky)
plt.plot(x, np.cos(x - 2), color='0.75')        # Grayscale between 0 and 1
plt.plot(x, np.cos(x - 3), color='#FFDD44')     # Hex code (RRGGBB from 00 to FF)
plt.show()
```



# Adjusting line plots : Line Style

Use **linestyle** keyword to adjust the Line Style

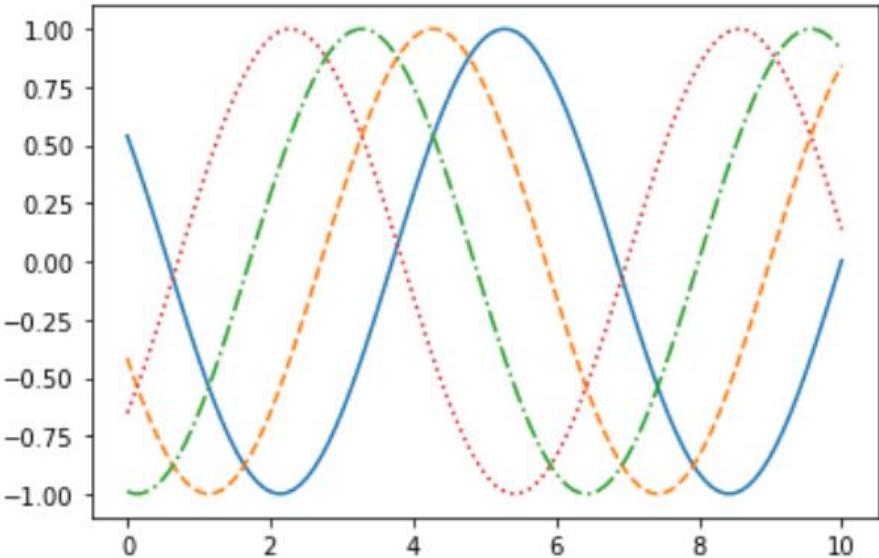
```
plt.plot(x, np.cos(x - 0), linestyle='solid')
plt.plot(x, np.cos(x - 1), linestyle='dashed')
plt.plot(x, np.cos(x - 2), linestyle='dashdot')
plt.plot(x, np.cos(x - 3), linestyle='dotted');
```



# Adjusting line plots : Line Style

Use **linestyle** keyword to adjust the Line Style

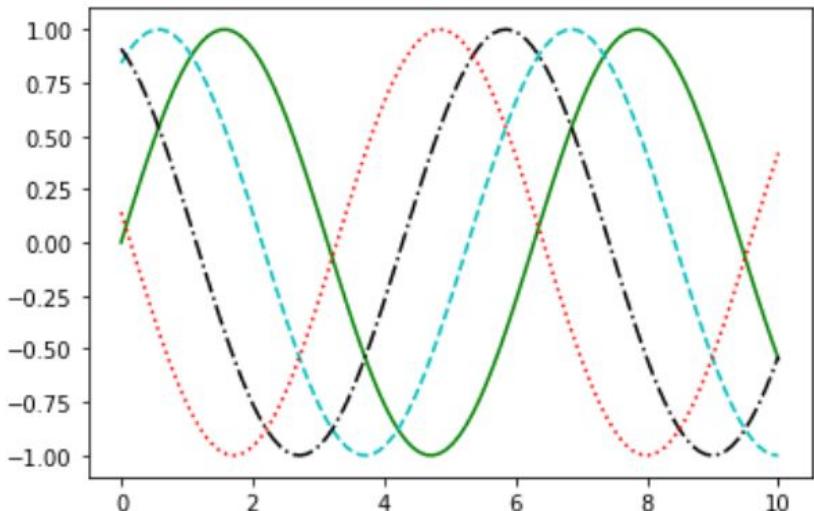
```
: # For short, you can use the following codes:  
plt.plot(x, np.cos(x + 1), linestyle='--') # solid  
plt.plot(x, np.cos(x + 2), linestyle='---') # dashed  
plt.plot(x, np.cos(x + 3), linestyle='-.') # dashdot  
plt.plot(x, np.cos(x + 4), linestyle=':'); # dotted
```



# Adjusting Line Plot : Line Style & Color

Line Style & Color can be combined into Single Non-Keyword Argument

```
plt.plot(x, np.sin(x), '-g') # solid green
plt.plot(x, np.sin(x+1), '--c') # dashed cyan
plt.plot(x, np.sin(x+2), '-.k') # dashdot black
plt.plot(x, np.sin(x+3), ':r') # dotted red
plt.show()
```



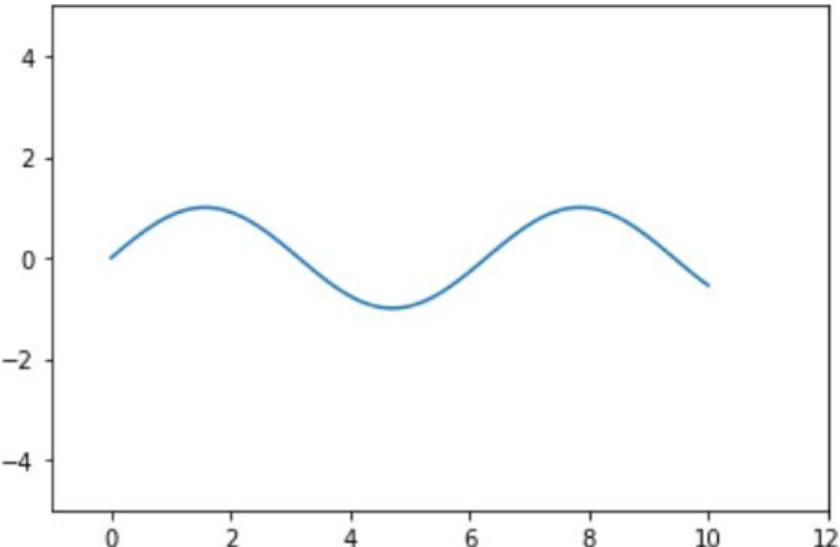
# Adjusting Plots: Axes Limits

```
plt.plot(x,np.sin(x))
plt.xlim(-1,12) #Specify X Axis Range
plt.ylim(-5,5)  #Specify Y Axis Range
plt.show()
```

Methods:

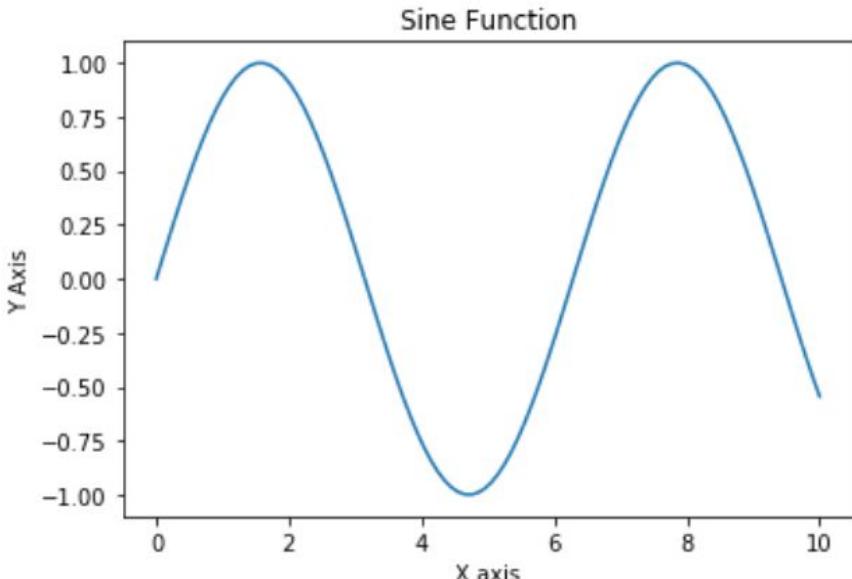
`plt.xlim()`

`plt.ylim()`



# Adjusting Plots: Labelling Plots

```
|: plt.plot(x,np.sin(x))
plt.xlabel("X axis")
plt.ylabel("Y Axis")
plt.title("Sine Function")
plt.show()
```



Method:

`plt.xlabel()`

`plt.ylabel()`

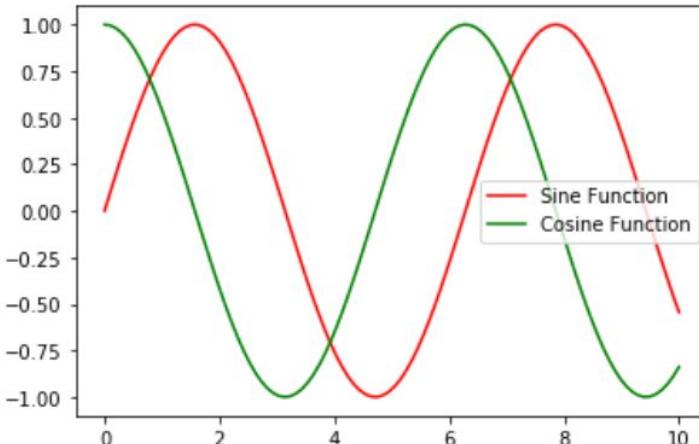
`plt.title()`

# Adjusting Plots: Adding Legends

Method:

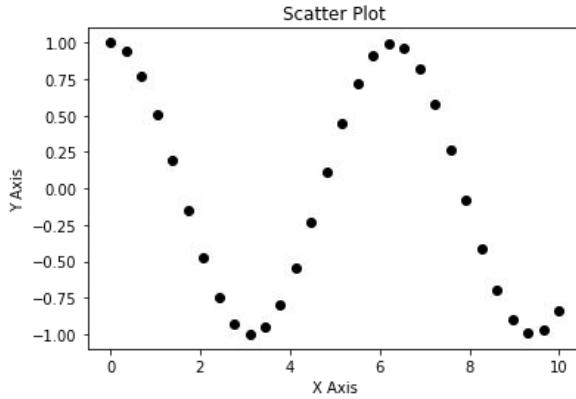
```
plt.legend()
```

```
: plt.plot(x,np.sin(x), color = "r", label = "Sine Function")
plt.plot(x,np.cos(x), color = "g", label = "Cosine Function")
plt.legend()
plt.show()
```



# Hands-On

# Scatter Plot





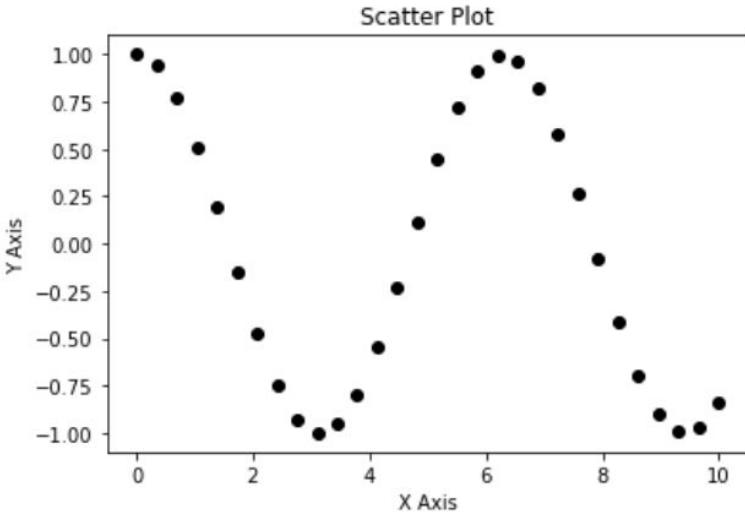
# Scatter Plot

A **Scatter (XY) Plot** has points that show the relationship between two sets of data.

# Scatter Plot

plt.plot() Method

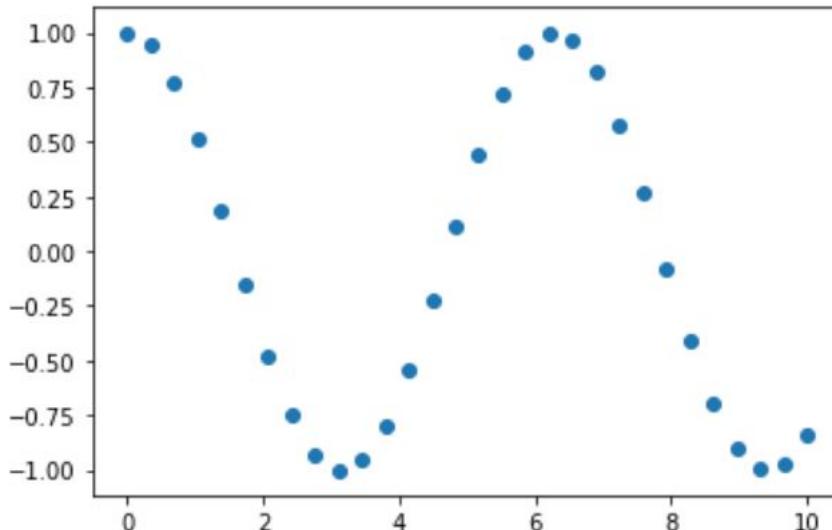
```
x = np.linspace(0, 10, 30)
y = np.cos(x)
plt.plot(x, y, 'o', color='black')
plt.xlabel("X Axis")
plt.ylabel("Y Axis")
plt.title("Scatter Plot")
plt.show()
```



# Scatter Plot

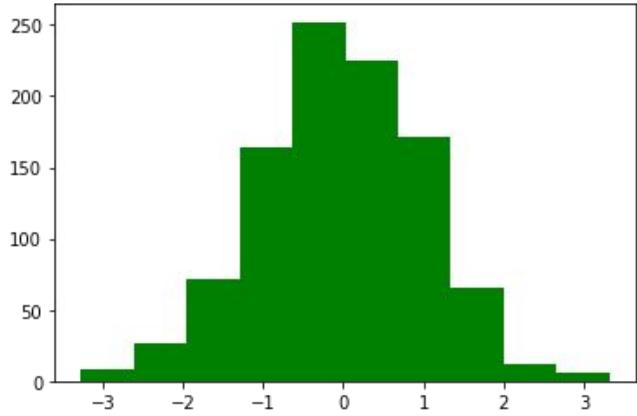
```
matplotlib.pyplot.scatter(x, y, s=None, c=None, marker=None, cmap=None, norm=None, vmin=None, vmax=None, alpha=None, linewidths=None, verts=<deprecated parameter>, edgecolors=None, *, plotnonfinite=False, data=None, **kwargs)
```

```
: plt.scatter(x, y, marker='o')
plt.show()
```



# Hands-On

# Histogram Plot





# What is Histogram Plot ?

A **histogram** is a display of statistical information that uses rectangles to show the frequency of data items in successive numerical intervals of equal size.

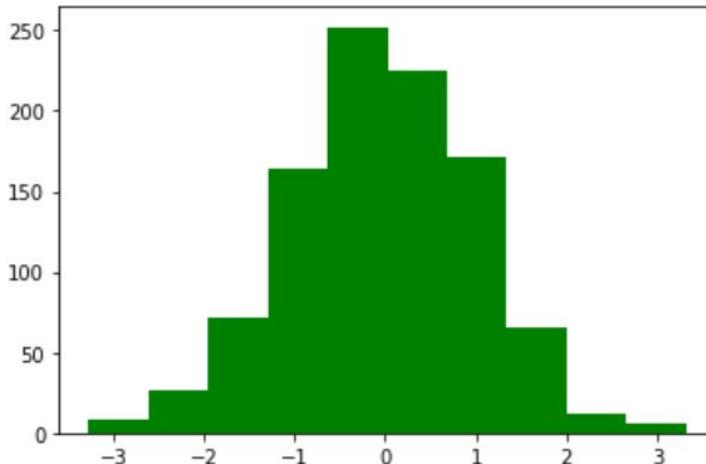
# Histogram plot

```
matplotlib.pyplot.hist(x, bins=None, range=None, density=False, weights=None, cumulative=False, bottom=None, histtype='bar', align='mid', orientation='vertical', rwidth=None, Log=False, color=None, label=None, stacked=False, *, data=None, **kwargs)
```

```
data = np.random.randn(1000)
plt.hist(data, color = "green")
plt.show()
```

Method:

```
plt.hist()
```

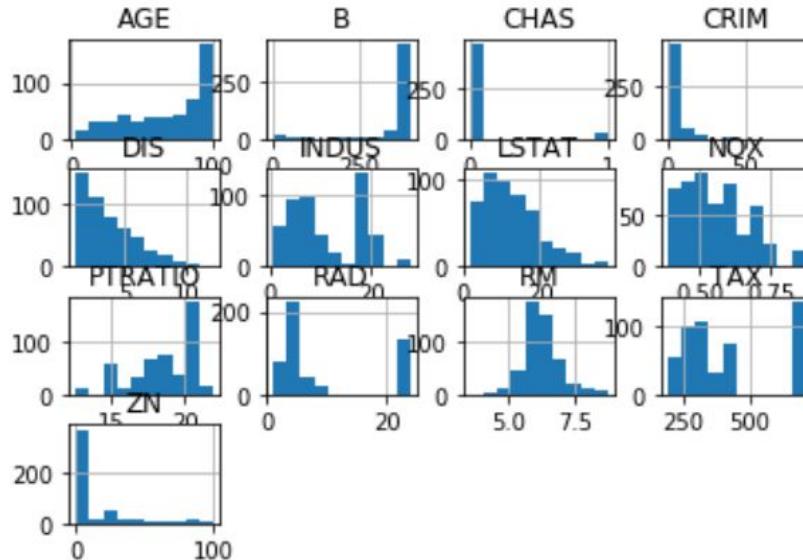


# Histogram plot

```
df = pd.DataFrame(data = boston.data, columns = boston.feature_names)
df.hist()
plt.show()
```

On Pandas DataFrame:

`df.hist()`



# Hands-On