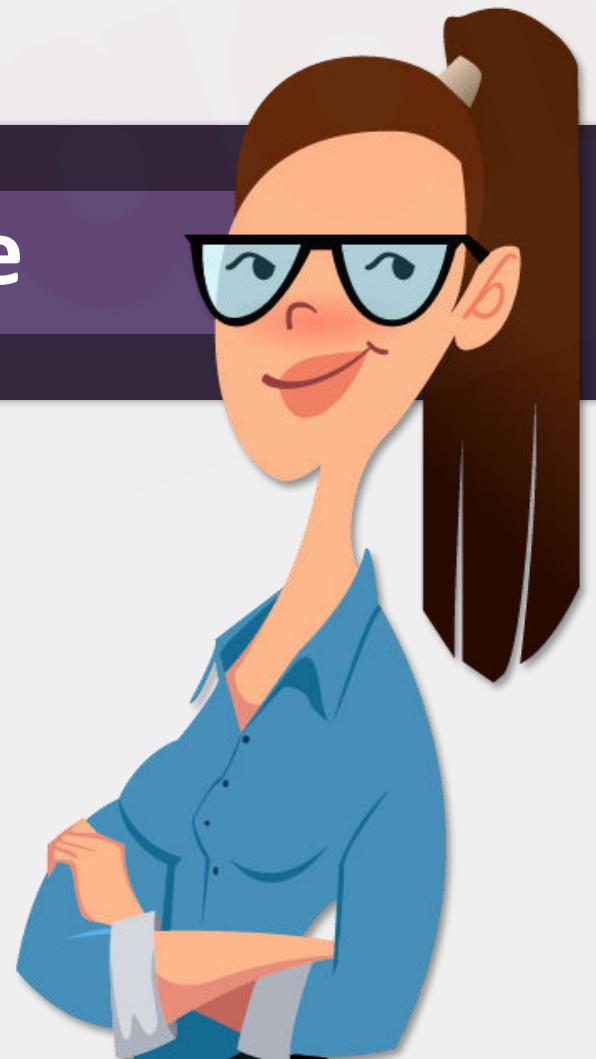


Python Certification Course



Data Visualization Basics:

- What is Data Visualization?
- Why visualize the data?
- What are various Data Visualization library in python?

Introduction to Matplotlib:

- What is Matplotlib?
- Why choose matplotlib for visualizing the data?
- What are different types of plot created using Matplotlib?



Basics of Data Visualization

What is Data Visualization?

- Data visualization is the presentation of data in a pictorial or graphical format
- Enables decision makers to see analytics and grasp difficult concepts or identify new patterns
- Data visualization is the representation of data in a pictorial or graphical format
- Allows the decision makers to see analytics, grasp difficult concepts and identify new patterns at ease

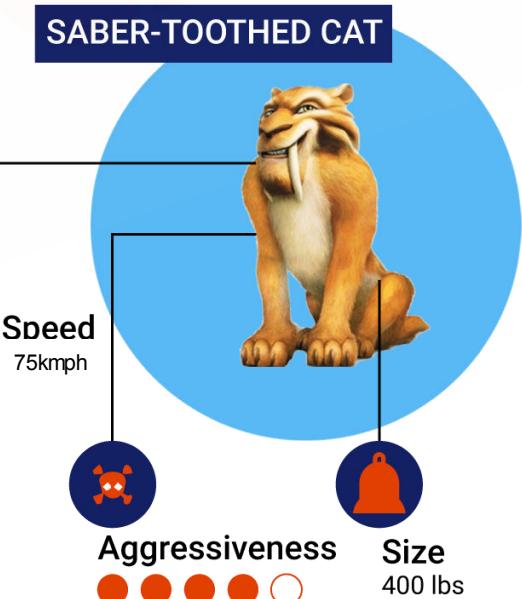
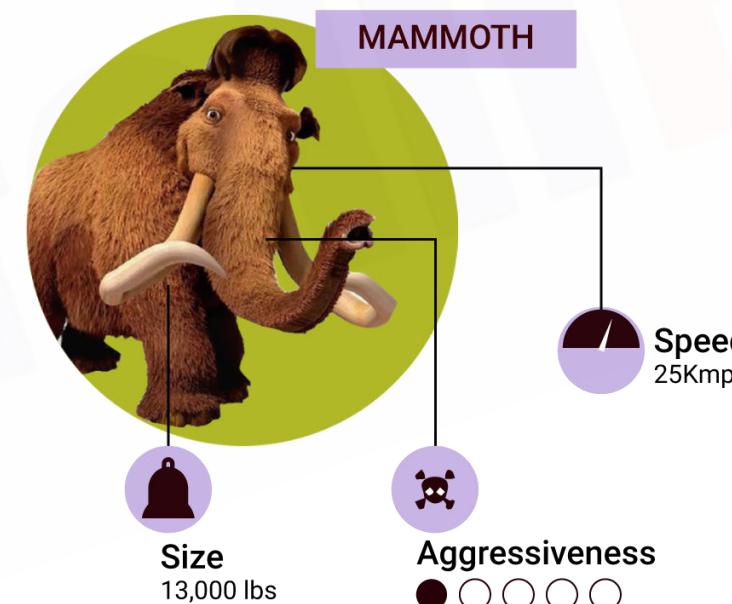


Basics of Data Visualization

Data Visualization Example:

| Name of Animal | Speed | Aggressiveness | Size |
|-----------------|-------|----------------|----------|
| Mammoth | 25 | low | 13000lbs |
| Saber-Tooth Cat | 75 | High | 400lbs |

Data



Data Visualization

Basics of Data Visualization

Why do we need Data Visualization?

Anscombe's Quartet

| | I | | II | | III | | IV | |
|--|----|-------|----|------|-----|-------|----|------|
| | x | y | x | y | x | y | x | y |
| | 10 | 8,04 | 10 | 9,14 | 10 | 7,46 | 8 | 6,58 |
| | 8 | 6,95 | 8 | 8,14 | 8 | 6,77 | 8 | 5,76 |
| | 13 | 7,58 | 13 | 8,74 | 13 | 12,74 | 8 | 7,71 |
| | 9 | 8,81 | 9 | 8,77 | 9 | 7,11 | 8 | 8,84 |
| | 11 | 8,33 | 11 | 9,26 | 11 | 7,81 | 8 | 8,47 |
| | 14 | 9,96 | 14 | 8,1 | 14 | 8,84 | 8 | 7,04 |
| | 6 | 7,24 | 6 | 6,13 | 6 | 6,08 | 8 | 5,25 |
| | 4 | 4,26 | 4 | 3,1 | 4 | 5,39 | 19 | 12,5 |
| | 12 | 10,84 | 12 | 9,13 | 12 | 8,15 | 8 | 5,56 |
| | 7 | 4,82 | 7 | 7,26 | 7 | 6,42 | 8 | 7,91 |
| | 5 | 5,68 | 5 | 4,74 | 5 | 5,73 | 8 | 6,89 |

| | | | | | | | | |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| SUM | 99,00 | 82,51 | 99,00 | 82,51 | 99,00 | 82,50 | 99,00 | 82,51 |
| AVG | 9,00 | 7,50 | 9,00 | 7,50 | 9,00 | 7,50 | 9,00 | 7,50 |
| STDEV | 3,32 | 2,03 | 3,32 | 2,03 | 3,32 | 2,03 | 3,32 | 2,03 |

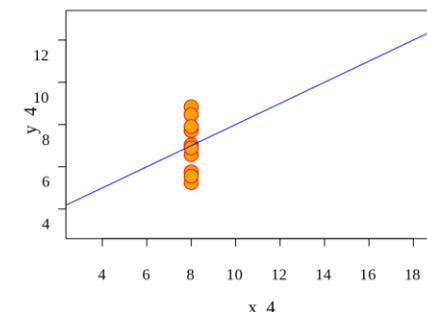
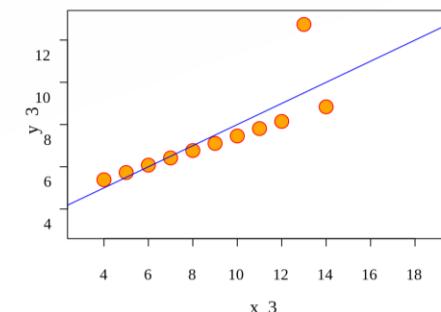
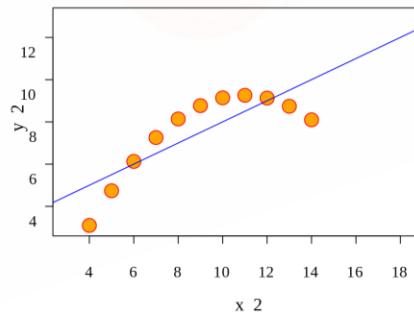
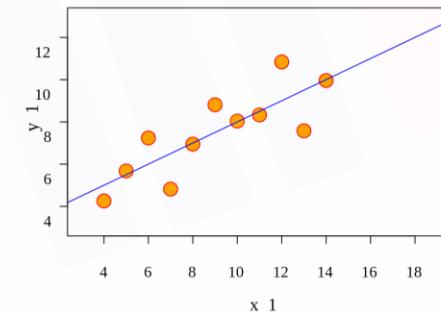
Anscombe's quartet comprises four datasets that have nearly identical simple descriptive statistics, yet appear very different when graphed.

Basics of Data Visualization

Why do we need Data Visualization?

Anscombe's Quartet

| I | | II | | III | | IV | | |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| x | y | x | y | x | y | x | y | |
| 10 | 8,04 | 10 | 9,14 | 10 | 7,46 | 8 | 6,58 | |
| 8 | 6,95 | 8 | 8,14 | 8 | 6,77 | 8 | 5,76 | |
| 13 | 7,58 | 13 | 8,74 | 13 | 12,74 | 8 | 7,71 | |
| 9 | 8,81 | 9 | 8,77 | 9 | 7,11 | 8 | 8,84 | |
| 11 | 8,33 | 11 | 9,26 | 11 | 7,81 | 8 | 8,47 | |
| 14 | 9,96 | 14 | 8,1 | 14 | 8,84 | 8 | 7,04 | |
| 6 | 7,24 | 6 | 6,13 | 6 | 6,08 | 8 | 5,25 | |
| 4 | 4,26 | 4 | 3,1 | 4 | 5,39 | 19 | 12,5 | |
| 12 | 10,84 | 12 | 9,13 | 12 | 8,15 | 8 | 5,56 | |
| 7 | 4,82 | 7 | 7,26 | 7 | 6,42 | 8 | 7,91 | |
| 5 | 5,68 | 5 | 4,74 | 5 | 5,73 | 8 | 6,89 | |
| SUM | 99,00 | 82,51 | 99,00 | 82,51 | 99,00 | 82,50 | 99,00 | 82,51 |
| AVG | 9,00 | 7,50 | 9,00 | 7,50 | 9,00 | 7,50 | 9,00 | 7,50 |
| STDEV | 3,32 | 2,03 | 3,32 | 2,03 | 3,32 | 2,03 | 3,32 | 2,03 |



Basics of Data Visualization

Data Visualization Libraries

matplotlib

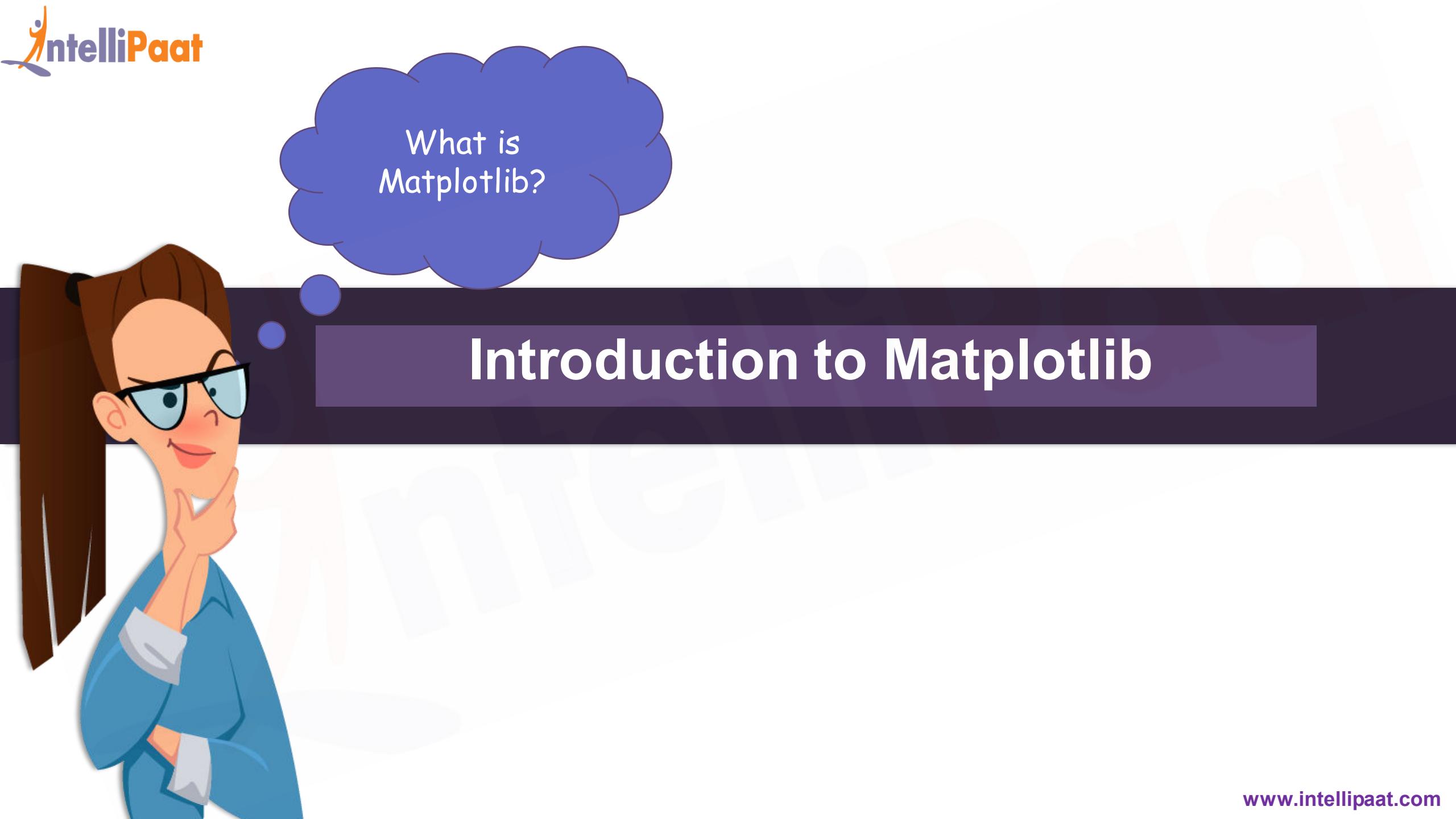


seaborn

ggplot



geoplotlib

A cartoon illustration of a person with brown hair and glasses, wearing a blue hoodie, looking thoughtful with a hand on their chin. A large blue thought bubble above them contains the text "What is Matplotlib?".

What is
Matplotlib?

Introduction to Matplotlib

Introduction to Matplotlib

What is Matplotlib?

- Python library for Data Visualization
- Create 2D graphs and plots by using python scripts.
- Produces output in a variety of hardcopy formats



Introduction to Matplotlib

Why choose Matplotlib?

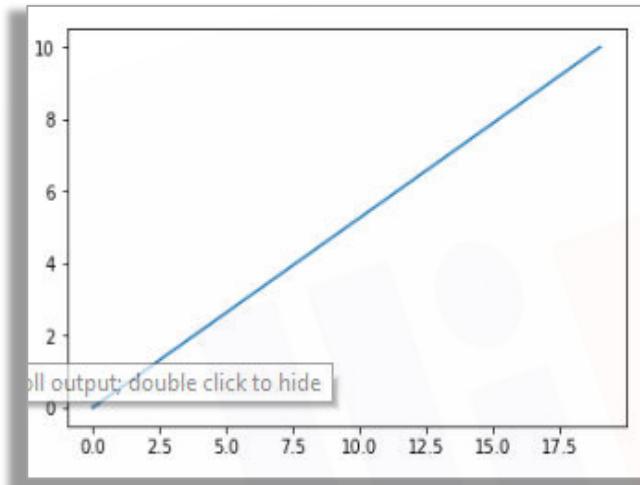
- Provides a module called Pyplot.
- Simple functions used for visualization
- Supports a very wide variety of graphs
- Easy integration with Pandas and Numpy.
- Provides an Object-Oriented API



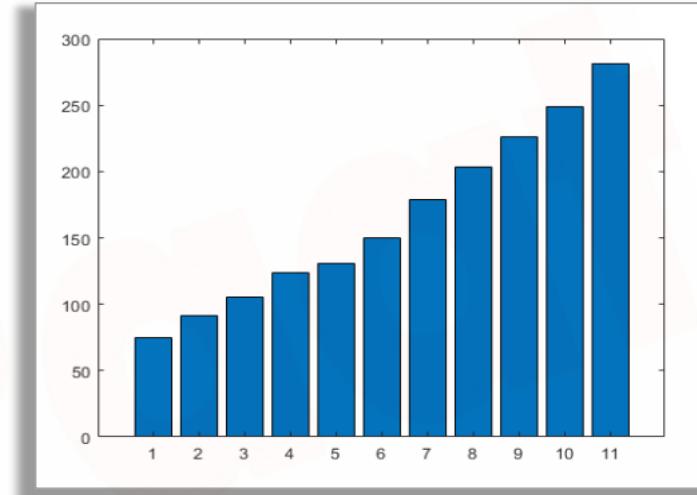
A cartoon illustration of a man with brown hair tied back, wearing blue-rimmed sunglasses and a blue shirt. He is peeking over a dark blue wall, with his hand resting on his chin in a thoughtful pose. The background behind the wall is a light, cloudy texture.

What are the types of Plots?

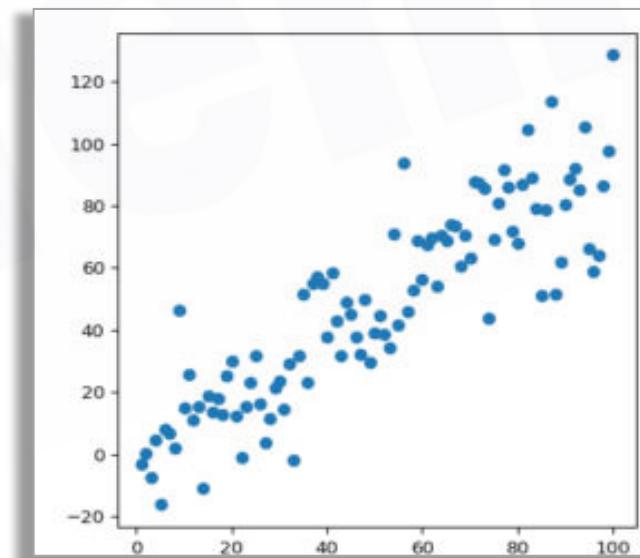
Types of Plots



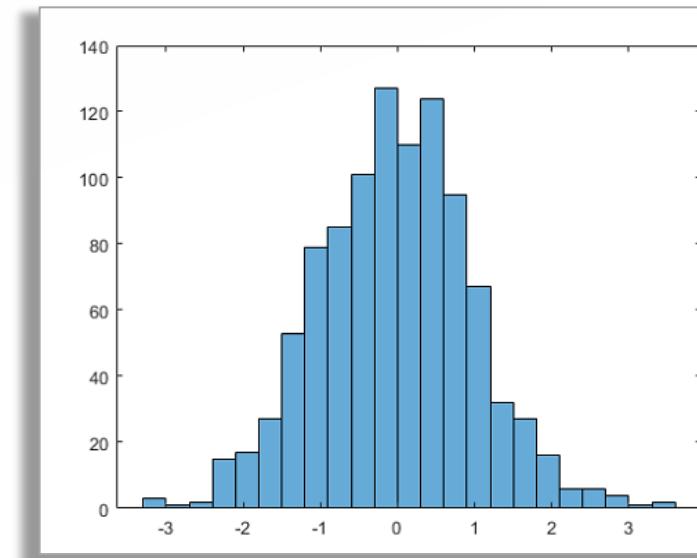
Line Plot



Bar Plot



Scatter Plot



Histogram

Types of Plots

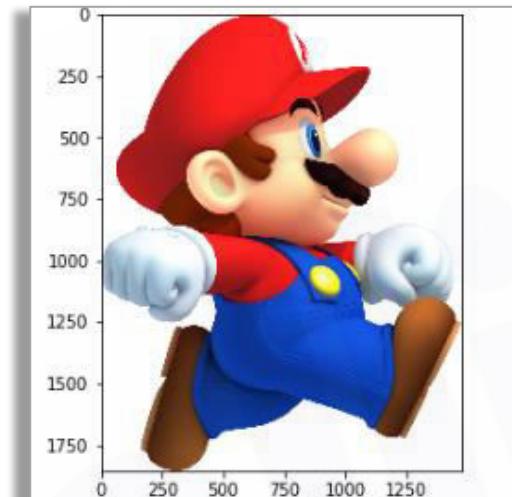
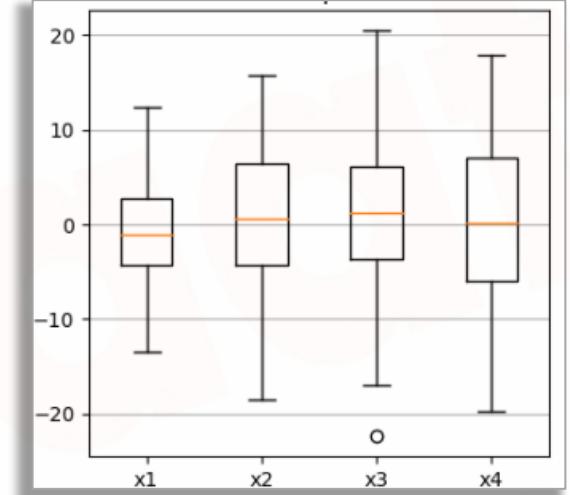
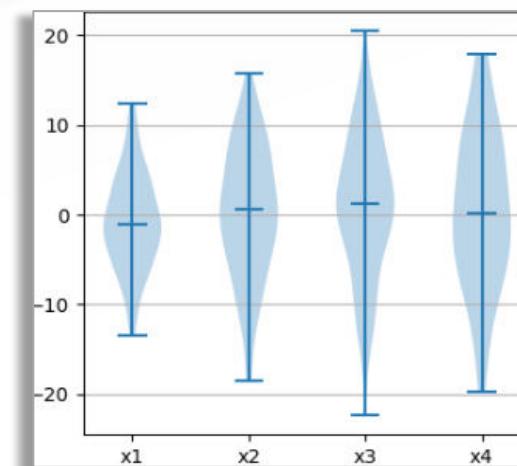


Image Plot

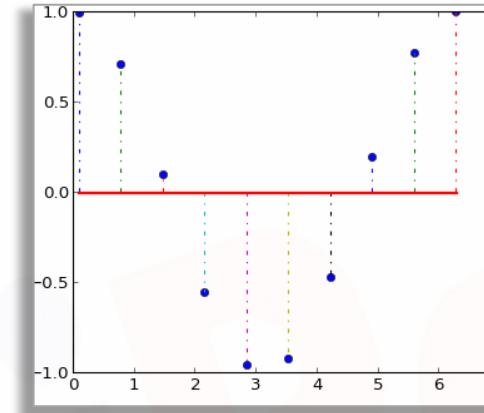


Box Plot

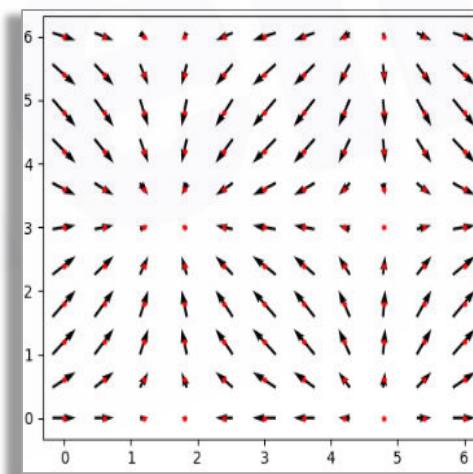


Violin Plot

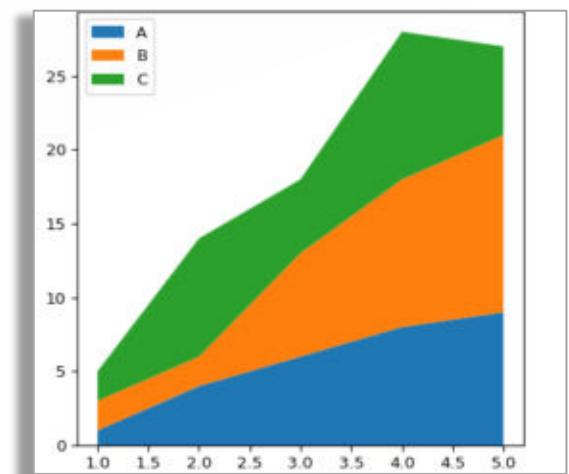
Types of Plots



Stream Plot

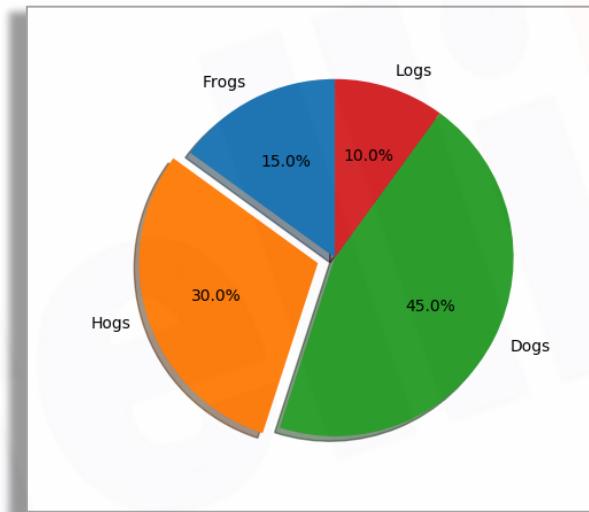


Quiver Plot

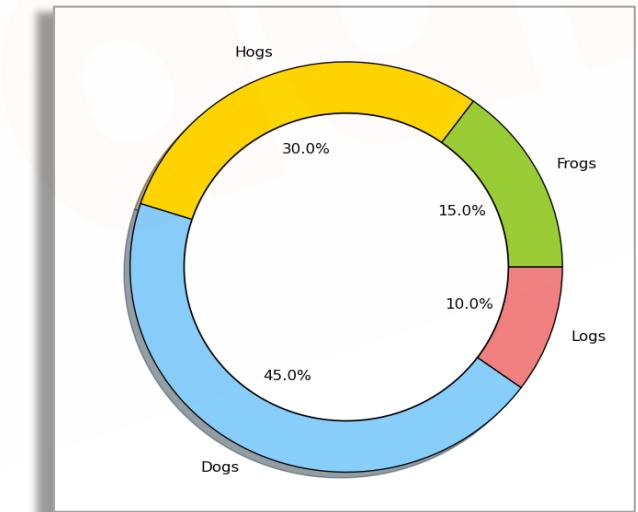


Area Plot

Types of Plots



Pie Plot



Donut Plot

Let's try some hands-on exercise



A cartoon illustration of a man with brown hair tied back, wearing blue-rimmed sunglasses and a blue shirt. He is peeking over a dark blue wall, with his hand resting on his chin in a thoughtful pose. The background behind the wall is a light, cloudy sky.

How to create a Line Plot?



Hands On: How to create a Line Plot

- How to create a Line Plot?
- How to customize a Line Plot?
- How to create two or more plots in one figure?

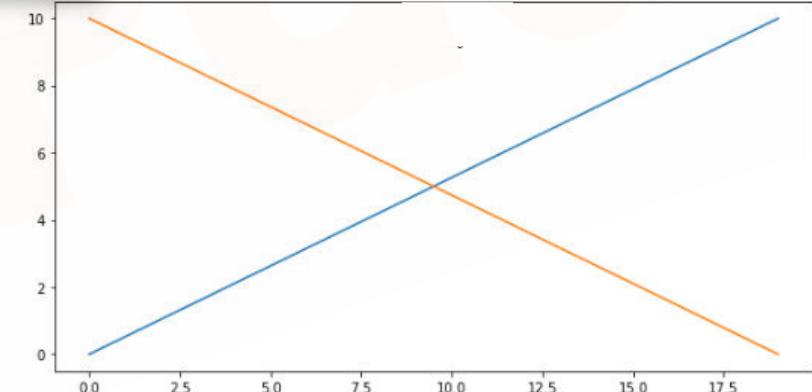
Demonstration: Line Plot

Input

```
In [33]: #import libraries
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

#preparing data
a = np.linspace(0, 10, 20)
b = np.linspace(10, 0, 20)
#Adding figure
fig=plt.figure(figsize=(10,5))
#Adding axes
ax1 = plt.subplot()
#simple Line plot of both a and b
ax1.plot(a)
ax1.plot(b)
#show the plot
plt.show()
```

Output



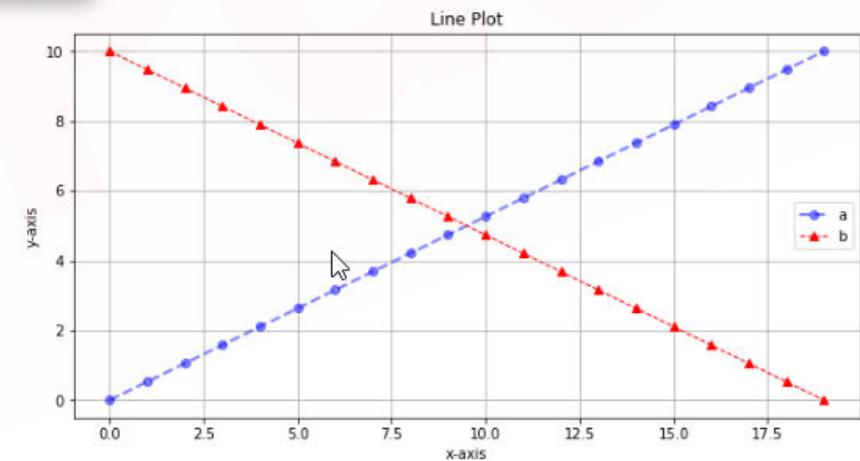
It is best to use a line plot when comparing fewer than 25 numbers. It is a quick, simple way to organize data.

Demonstration: Customized Line Plot

Input

```
In [16]: #import the libraries
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
#prepare the data
a = np.linspace(0, 10, 20)
b = np.linspace(10, 0, 20)
#Add figure
fig=plt.figure(figsize=(10,5))
#Add axes
ax1 = plt.subplot()
#Customization- Line Width, Line Style, Line Color, Line Opacity and Marker Options
ax1.plot(a,linewidth=2.0,linestyle='--',color='b',alpha=0.5,marker='o')
ax1.plot(b,linewidth=1.0,linestyle='--',color='r',alpha=1,marker='^')
#Customization-Title
plt.title('Line Plot')
#Customization-x-axis Label, y-axis Label
plt.xlabel('x-axis')
plt.ylabel('y-axis')
#Customization-Legend
plt.legend(['a','b'], loc='best')
#Add grid to the plot
plt.grid(True)
#save the plot
plt.savefig('LinePlot.png')
#show the plot
plt.show()
```

Output

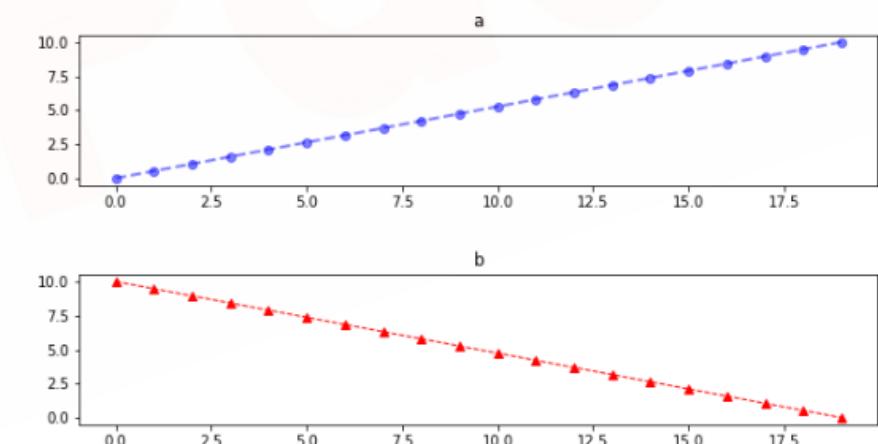


Demonstration: Sub-plotting

Input

```
In [34]: #importing libraries
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
#preparing data
a = np.linspace(0, 10, 20)
b = np.linspace(10, 0, 20)
#Add figure
fig=plt.figure(figsize=(10,5))
#Sub-plotting
ax1 = plt.subplot(211)    #2 rows 1 column 1st position
ax2 = plt.subplot(212)    #2 rows 1 column 2nd position
#Customization- Line Width, Line Style, Line Color, Line Opacity and Marker Options
ax1.plot(a,linewidth=2.0,linestyle='--',color='b',alpha=0.5,marker='o')
ax2.plot(b,linewidth=1.0,linestyle='--',color='r',alpha=1,marker='^')
#setting title of first subplot
ax1.set(title='a')
ax2.set(title='b')
#Adding Space between subplots
plt.subplots_adjust(left=None, bottom=None, right=None, top=None, wspace=None, hspace=0.6)
#showing plot
plt.show()
```

Output



Use sub-plotting while comparing plots

A cartoon illustration of a man with brown hair tied back, wearing dark sunglasses and a blue shirt. He is peeking over a dark blue wall, with his hand resting on his chin in a thoughtful pose. The background behind the wall is a light, out-of-focus gradient.

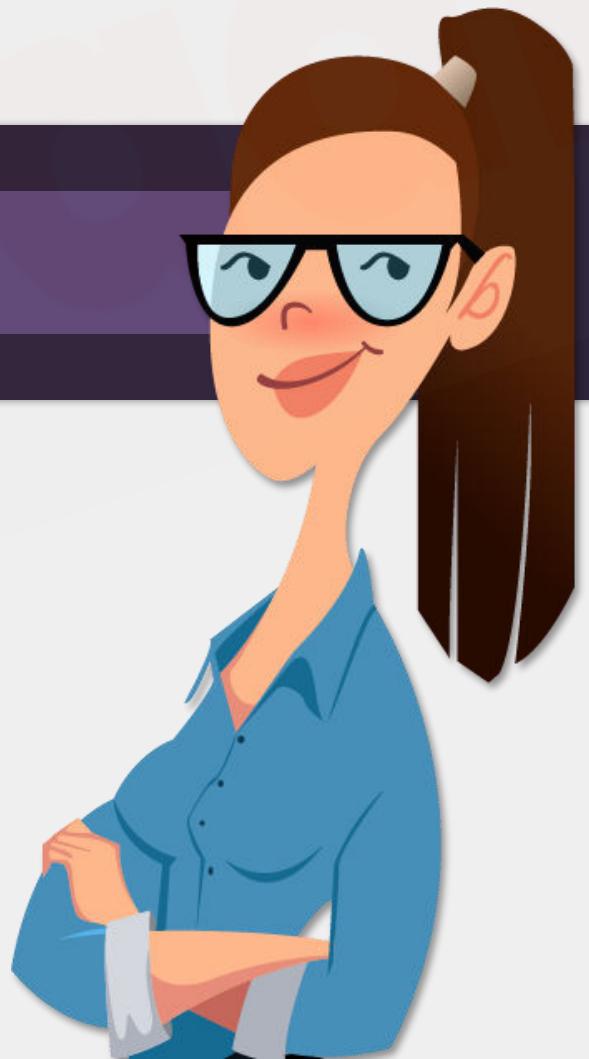
How to create a Bar Plot?



Hands On: How to create a Bar Plot

- How to create a Bar Plot?
- How to customize a Bar Plot?
- How to create a horizontal Bar Plot?

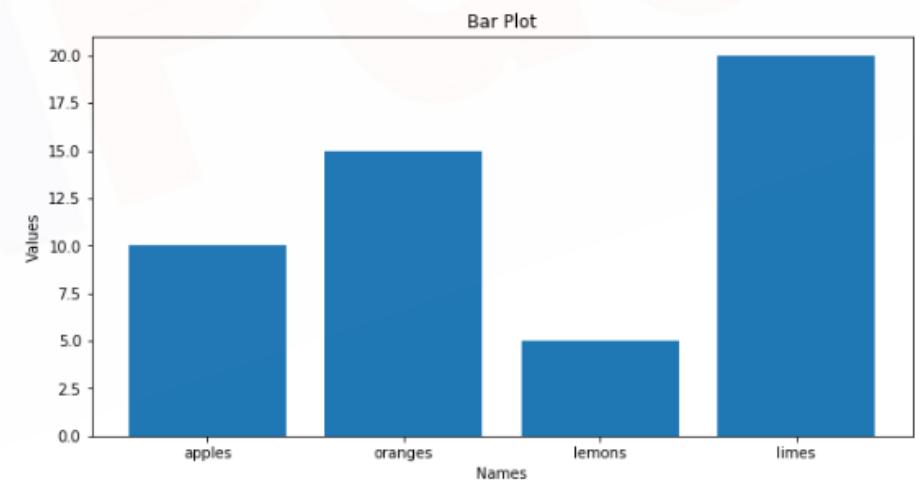
Hands-on: Bar Plot



Input

```
In [37]: #import library
import matplotlib.pyplot as plt
%matplotlib inline
#prepare data
data = {'apples': 10, 'oranges': 15, 'lemons': 5, 'limes': 20}
names = list(data.keys())
values = list(data.values())
#Add figure
fig=plt.figure(figsize=(10,5))
#Sub-plotting
ax1 = plt.subplot()
#plot
ax1.bar(names, values)
#Customization-Title
plt.title('Bar Plot')
#Customization-x-axis label, y-axis label
plt.xlabel('Names')
plt.ylabel('Values')
#showing plot
plt.show()
```

Output



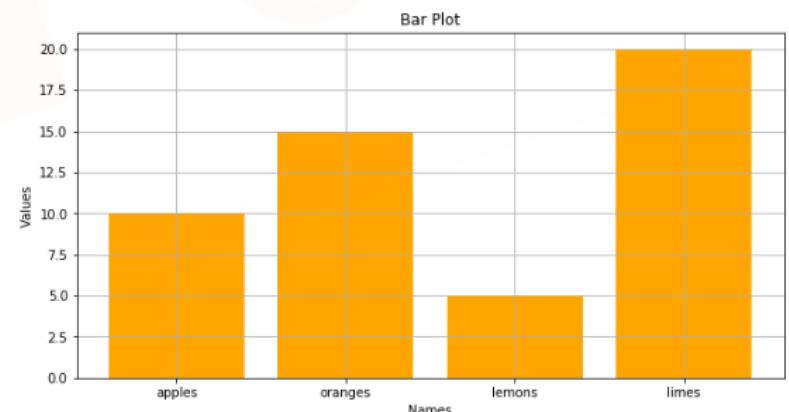
A bar chart or bar graph is a chart or graph that presents categorical data with rectangular bars with heights or lengths proportional to the values that they represent. The bars can be plotted vertically or horizontally.

Demonstration: Customized Bar Plot

Input

```
In [40]: #importing Libraries
import matplotlib.pyplot as plt
%matplotlib inline
#prepare data
data = {'apples': 10, 'oranges': 15, 'lemons': 5, 'limes': 20}
names = list(data.keys())
values = list(data.values())
#Add figure
fig=plt.figure(figsize=(10,5))
#adding axes
ax1 = plt.subplot()
#Customization-alignment, color
ax1.bar(names, values, align='center', color='orange')
#Customization-Title
plt.title('Bar Plot')
#Customization-x-axis label, y-axis label
plt.xlabel('Names')
plt.ylabel('Values')
#customization-add grid
plt.grid(True)
#Save the plot
plt.savefig('BarPlot.png')
#show plot
plt.show()
```

Output

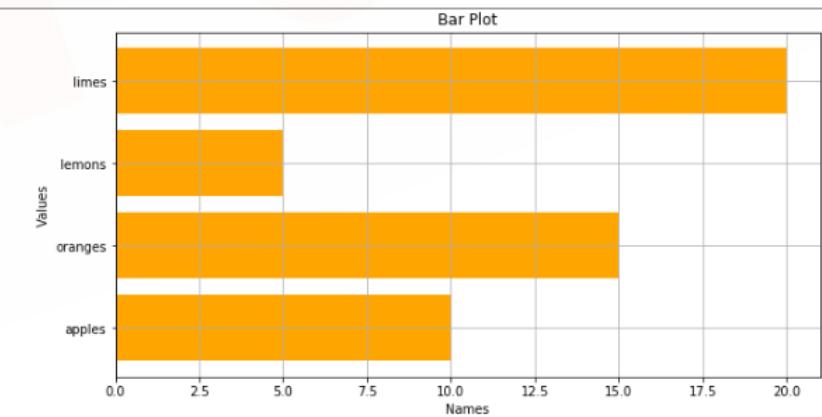


Demonstration: Horizontal Bar Plot

Input

```
In [41]: #import Libraries
import matplotlib.pyplot as plt
%matplotlib inline
#prepare the data
data = {'apples': 10, 'oranges': 15, 'lemons': 5, 'limes': 20}
names = list(data.keys())
values = list(data.values())
#Add figure
fig=plt.figure(figsize=(10,5))
#adding axes
ax1 = plt.subplot()
#Customization-alignment, color
ax1.barih(names, values, align='center', color='orange')
#Customization-Title
plt.title('Bar Plot')
#Customization-x-axis label, y-axis label
plt.xlabel('Names')
plt.ylabel('Values')
#customization-add grid
plt.grid(True)
#Save the plot
plt.savefig('HorizontalBarPlot.png')
#show plot
plt.show()
```

Output



Hands-on: Scatter Plot



A cartoon illustration of a man with brown hair tied back, wearing dark sunglasses and a blue shirt. He is peeking over a dark blue horizontal bar from behind a white wall, with his hand to his chin in a thoughtful pose. The bar contains the title text.

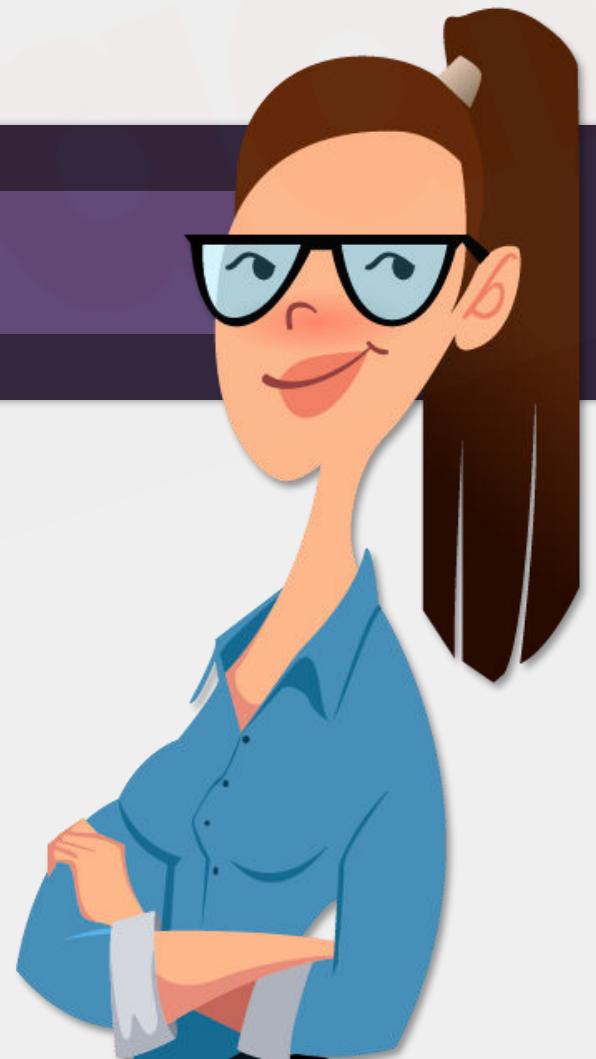
How to create a Scatter Plot?

Hands On: How to create a Scatter Plot

- How to create a Scatter Plot?
- How to customize a Scatter Plot?



Hands-on: Scatter Plot

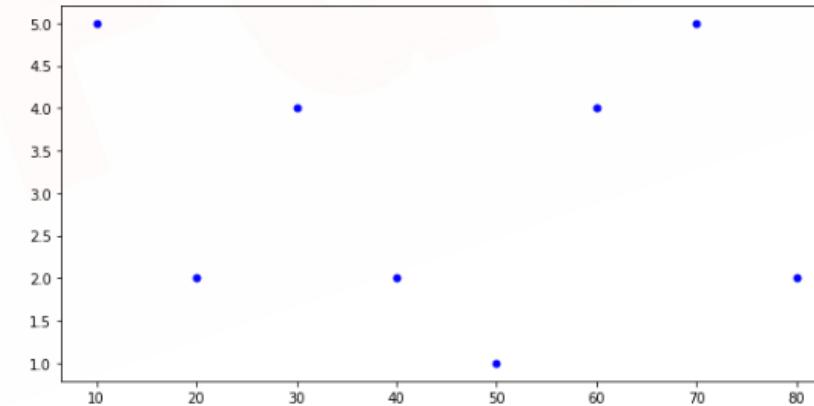


Demonstration: Scatter Plot

Input

```
In [41]: #import libraries
import matplotlib.pyplot as plt
%matplotlib inline
#prepare the data
data = {'apples': 10, 'oranges': 15, 'lemons': 5, 'limes': 20}
names = list(data.keys())
values = list(data.values())
#Add figure
fig=plt.figure(figsize=(10,5))
#adding axes
ax1 = plt.subplot()
#Customization-alignment, color
ax1.bart(names, values, align='center', color='orange')
#Customization-Title
plt.title('Bar Plot')
#Customization-x-axis label, y-axis label
plt.xlabel('Names')
plt.ylabel('Values')
#customization-add grid
plt.grid(True)
#Save the plot
plt.savefig('HorizontalBarPlot.png')
#show plot
plt.show()
```

Output



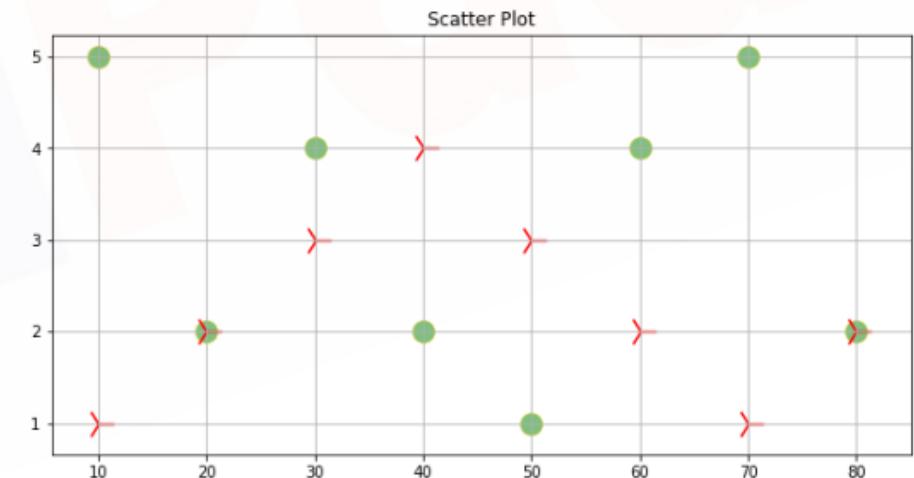
Scatter plots are used to plot data points on a horizontal and a vertical axis in the attempt to show how much one variable is affected by another. Helps visualizing the correlation.

Demonstration: Customized Scatter Plot

Input

```
In [47]: #importing library
import matplotlib.pyplot as plt
%matplotlib inline
#prepare data
a = [10,20,30,40,50,60,70,80]
b = [5,2,4,2,1,4,5,2]
x = [1,2,3,4,3,2,1,2]
#Add figure
fig=plt.figure(figsize=(10,5))
#add axes
ax1 = plt.subplot()
#customization-color,size,edgecolors,marker,alpha
ax1.scatter(a, b, c='g', s=200, edgecolors='y', marker='o', alpha=0.5)
ax1.scatter(a, x, c='r', s=400, edgecolors='b', marker='4',alpha=1)
#Customization-Title
plt.title('Scatter Plot')
#customization-add grid
plt.grid(True)
#Save the plot
plt.savefig('ScatterPlot.png')
#show plot
plt.show()
```

Output



A cartoon illustration of a man with brown hair tied back, wearing blue sunglasses and a blue shirt, peeking over a dark blue wall. He has a thoughtful expression, with his hand to his chin. The background is a light, out-of-focus landscape.

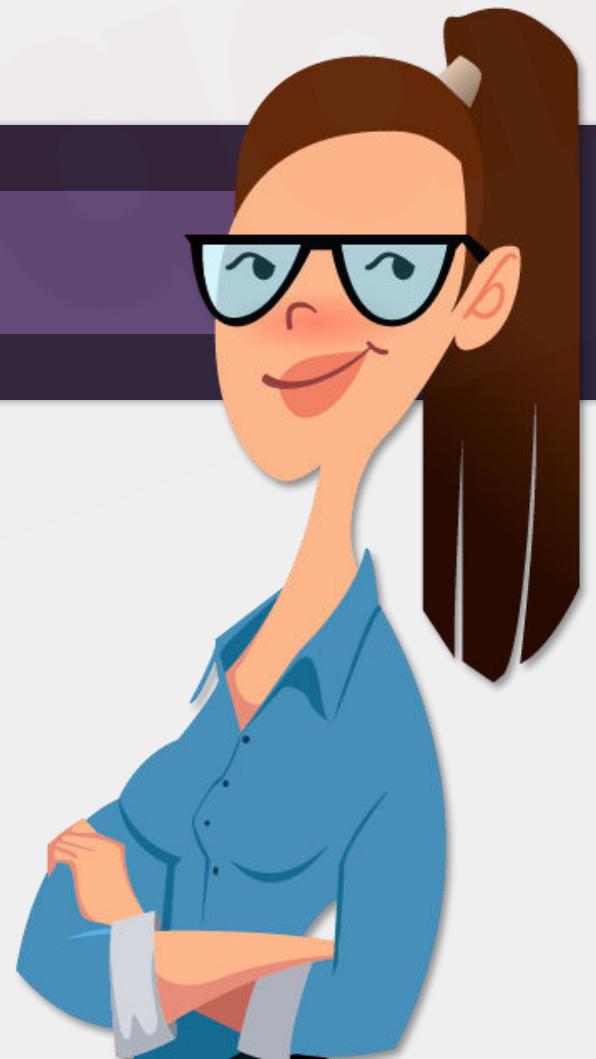
How to create a Histogram?

Hands On: How to create a Histogram?

- How to create a Histogram?
- How to customize a Histogram?



Hands-on: Histogram

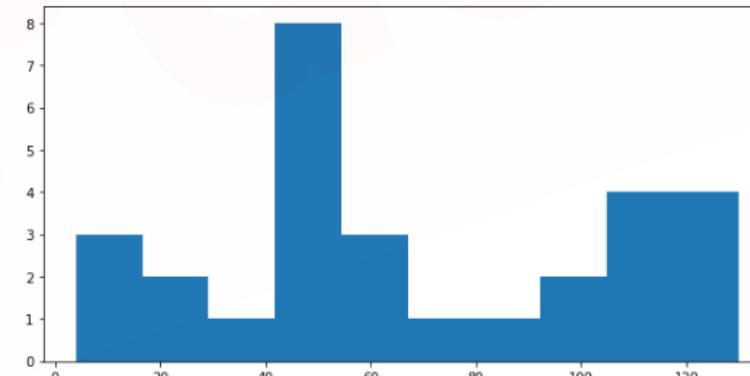


Demonstration: Histogram

Input

```
In [50]: #import Libraries
import matplotlib.pyplot as plt
%matplotlib inline
#prepare data
number = [12,55,11,62,45,21,22,34,42,42,4,99,102,110,120,121,122,130,111,115,112,80,75,65,54,44,43,42,48]
#Add figure
fig=plt.figure(figsize=(10,5))
#add axes
ax1 = plt.subplot()
#plot and customize
ax1.hist(number, bins=10)
#show plot
plt.show()
```

Output



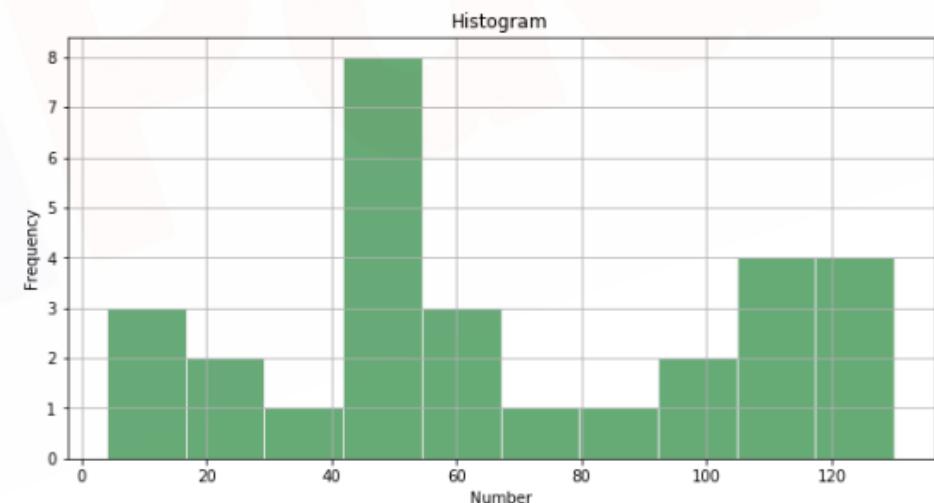
Plots used to display frequency across a continuous or discrete variable

Demonstration: Customized Histogram

Input

```
In [48]: #import libraries
import matplotlib.pyplot as plt
%matplotlib inline
#prepare data
number = [12,55,11,62,45,21,22,34,42,42,4,99,102,110,120,121,122,130,111,115,112,80,75,65,54,44,43,42,48]
#Add figure
fig=plt.figure(figsize=(10,5))
#add axes
ax1 = plt.subplot()
#adding hex color codes
ax1.hist(number, bins=10, edgecolor="#E6E6E6", color="#66aa76")
plt.title('Histogram')
#Customization-x-axis label, y-axis label
plt.xlabel('Number')
plt.ylabel('Frequency')
#customization-add grid
plt.grid(True)
#save the plot
plt.savefig('HistogramPlot.png')
#show plot
plt.show()
```

Output



A cartoon illustration of a man with brown hair tied back, wearing blue sunglasses and a blue shirt, peeking out from behind a dark brown door frame. He has a thoughtful expression with his hand to his chin. The background is a light gray.

How to create a Box Plot & Violin Plot?

Hands On: Box Plot and Violin Plot

- How to create a Box Plot?
- How to create a Violin Plot?



Hands-on: Box Plot and Violin Plot

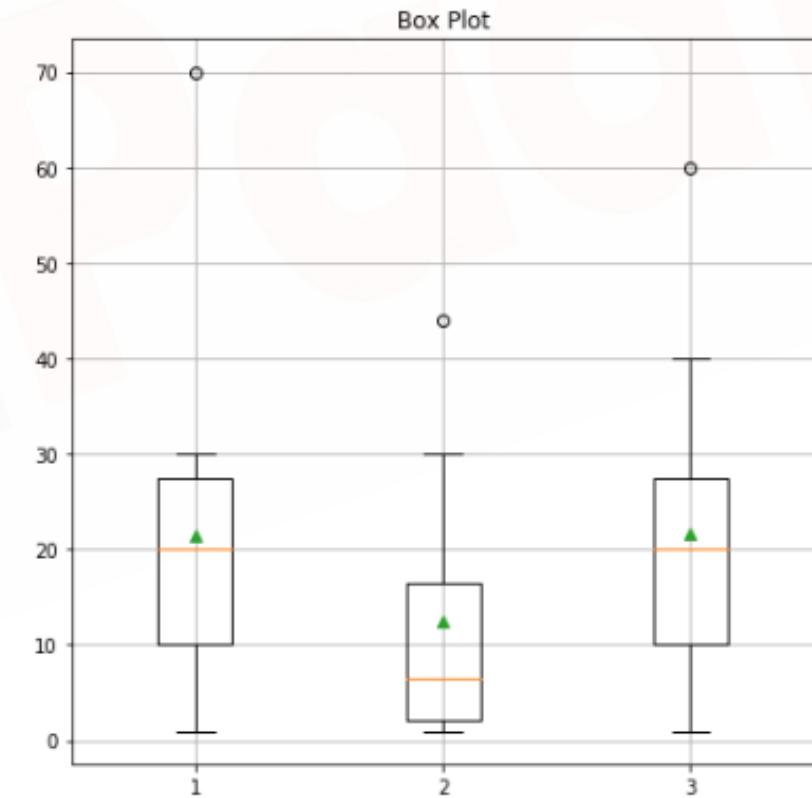


Demonstration: Box Plot

Input

```
In [52]: #import libraries
import matplotlib.pyplot as plt
%matplotlib inline
#data preparation
total = [20,4,1,30,20,10,20,70,30,10]
orders = [10,3,1,15,17,2,30,44,2,1]
discount = [30,20,10,5,20,10,60,20,40,1]
data = list([total, orders, discount ])
#Add figure
fig=plt.figure(figsize=(7,7))
#add axes
ax1 = plt.subplot()
#plot data
ax1.boxplot(data, showmeans=True)
#add title
plt.title('Box Plot')
#customization-add grid
plt.grid(True)
#save the plot
plt.savefig('BoxPlot.png')
#show plot
plt.show()
```

Output



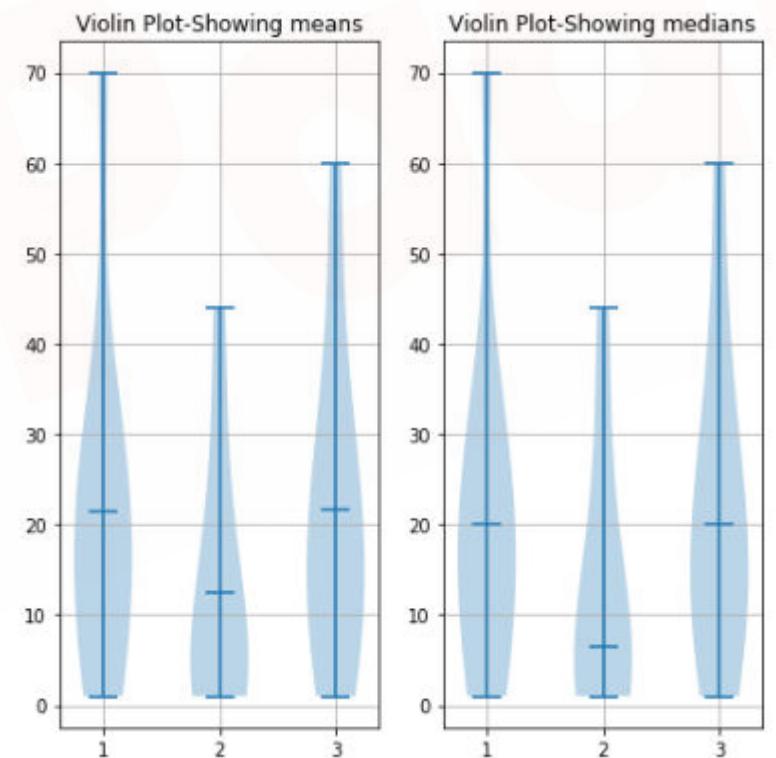
Box plot is very helpful in viewing the summary of dataset in an efficient way also box plot helps you in doing outlier analysis

Demonstration: Violin Plot

Input

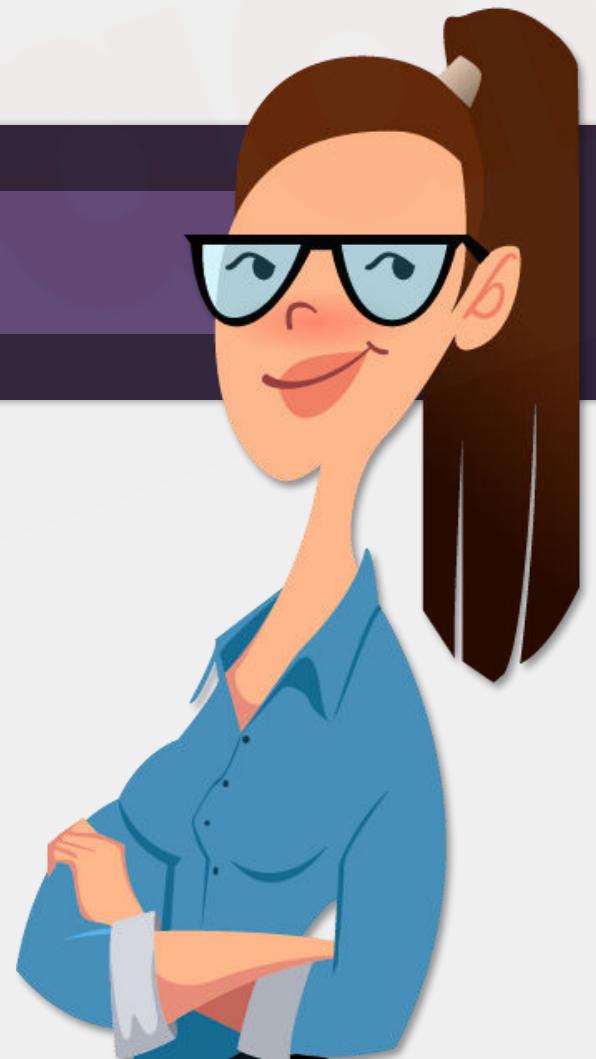
```
In [54]:  
import libraries  
import matplotlib.pyplot as plt  
%matplotlib inline  
#prepare data  
total = [20,4,1,30,20,10,20,70,30,10]  
orders = [10,3,1,15,17,2,30,44,2,1]  
discount = [30,20,10,5,20,10,60,20,40,1]  
data = list([total, orders, discount ])  
#Add figure  
fig=plt.figure(figsize=(7,7))  
#add axes  
ax1 = plt.subplot(121)  
ax2 = plt.subplot(122)  
ax1.violinplot(data, showmeans=True, showmedians=False)  
ax2.violinplot(data, showmeans=False, showmedians=True)  
#add axes title  
ax1.set_title('Violin Plot-Showing means')  
ax2.set_title('Violin Plot-Showing medians')  
#customization-add grid  
ax1.grid(True)  
ax2.grid(True)  
#save the plot  
plt.savefig('ViolinPlot.png')  
#show the plot  
plt.show()
```

Output



Allows to visualize the distribution of a numeric variable for one or several groups. Adapted when the amount of data is huge and showing individual observations gets impossible.

Hands-on: Image Plot



A cartoon illustration of a man with brown hair in a ponytail, wearing blue sunglasses and a blue shirt, peeking out from behind a dark door frame. He has a thoughtful expression with his hand to his chin.

How to create a Image Plot?

Hands On: Image Plot

- How to create a Box Plot?
- How to create a Violin Plot?



Hands-on: Image Plot



Converting PNG to Numpy Array

Input

```
In [56]: #import numpy and matplotlib
#Python Imaging Library
from PIL import Image
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
#Load the image
img = Image.open("mario.png")
#convert to .npy
arr = np.array(img)
#display array
print(arr)
```



.png

Used for image manipulation

Output

```
[[[255 255 255  0]
[255 255 255  0]
[255 255 255  0]
...
[255 255 255  0]
[255 255 255  0]
[255 255 255  0]]
[[255 255 255  0]
[255 255 255  0]
[255 255 255  0]
...
[255 255 255  0]
[255 255 255  0]
[255 255 255  0]]
[[255 255 255  0]
[255 255 255  0]
[255 255 255  0]
...
[255 255 255  0]
[255 255 255  0]
[255 255 255  0]]
[[255 255 255  0]
[255 255 255  0]
[255 255 255  0]
...
[255 255 255  0]
[255 255 255  0]
[255 255 255  0]]
[[255 255 255  0]
[255 255 255  0]
[255 255 255  0]
...
[255 255 255  0]
[255 255 255  0]
[255 255 255  0]]]
```



Numpy array

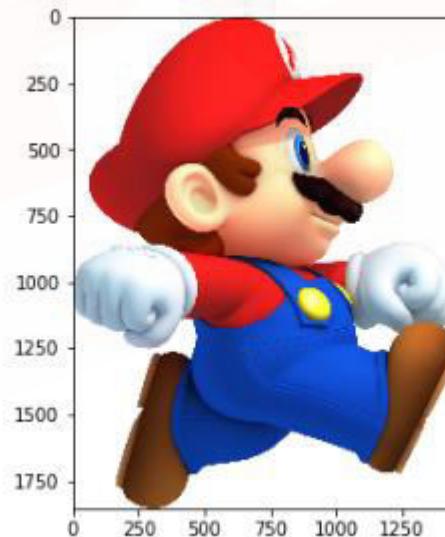
Demonstration: Image Plot

Input

```
In [59]: #import numpy and matplotlib
#Python Imaging Library
from PIL import Image
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
#prepare data
img = Image.open("mario.png")
#convert to .npy
arr = np.array(img)
#Add figure
fig=plt.figure(figsize=(10,5))
#add axes
ax1 = plt.subplot()
#plot image
ax1.imshow(arr)
```

Output

```
Out[59]: <matplotlib.image.AxesImage at 0x25d09eb7f60>
```



Demonstration: Image to Histogram

Input

```
In [64]: #import numpy and matplotlib
#Python Imaging Library
from PIL import Image
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
#Load the image
img = Image.open("mario.png")
#convert to .npy
arr = np.array(img)
#Add figure
fig=plt.figure(figsize=(7,5))
#add axes
ax1 = plt.subplot()
#Plot the histogram of this image
ax1.hist(arr.ravel(), bins=20, range=(0, 260), fc='orange', ec='white')
#arr.ravel()-returns contiguous flattened array(1D array with all
#the input-array melements and with the same type as it)
#display the histogram
plt.show()
```

.png



Numpy array

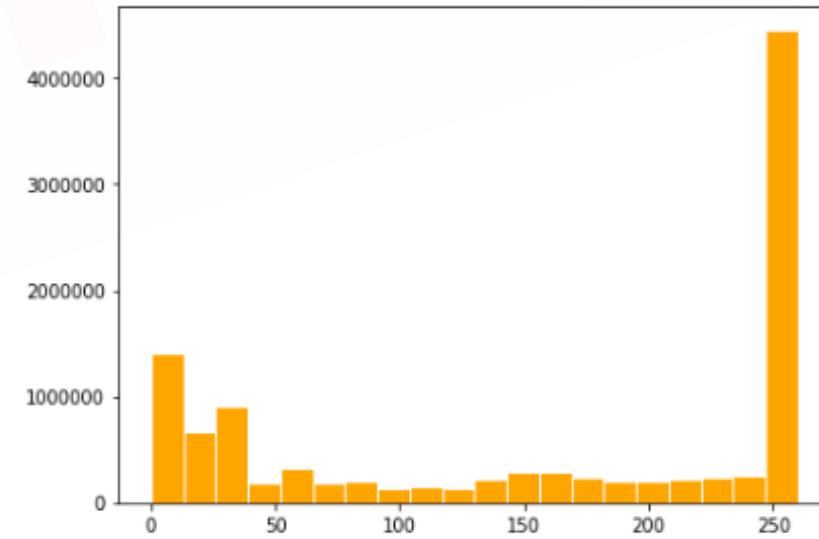
```
[[[255 255 255  0]
[255 255 255  0]
[255 255 255  0]
...
[255 255 255  0]
[255 255 255  0]
[255 255 255  0]]]

[[255 255 255  0]
[255 255 255  0]
[255 255 255  0]
...
[255 255 255  0]
[255 255 255  0]
[255 255 255  0]]]

[[255 255 255  0]
[255 255 255  0]
[255 255 255  0]
...
[255 255 255  0]
[255 255 255  0]
[255 255 255  0]]]
```



Output



Hands-on: Quiver and Stream Plot

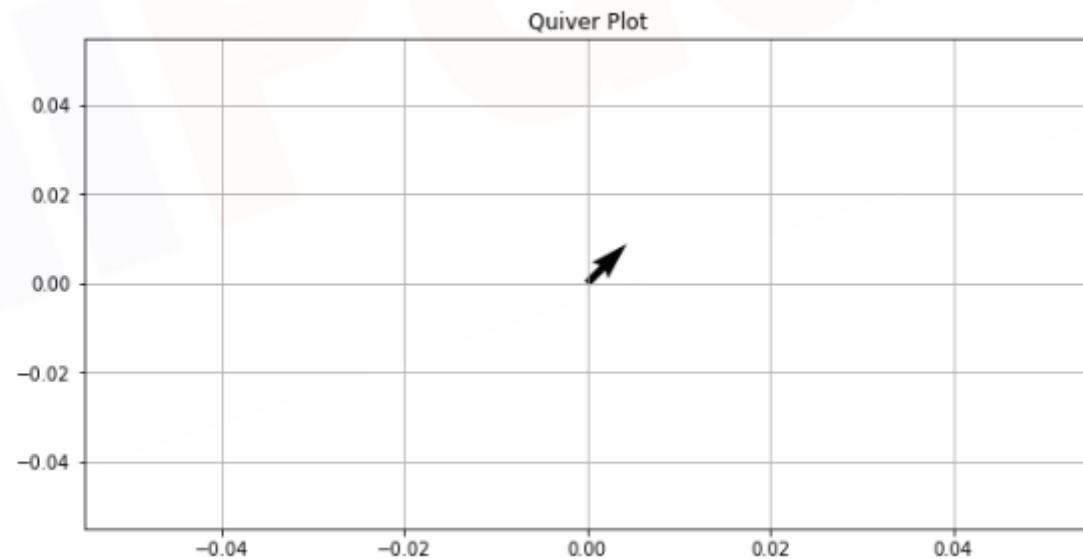


Demonstration: Quiver Plot

Input

```
In [67]: #import libraries
import matplotlib.pyplot as plt
%matplotlib inline
#prepare data
x_pos = 0
y_pos = 0
x_direct = 1
y_direct = 1
#Add figure
fig=plt.figure(figsize=(10,5))
#add axes
ax1 = plt.subplot()
#plot
ax1.quiver(x_pos, y_pos, x_direct, y_direct)
#Customization-title
plt.title('Quiver Plot')
#customization-add grid
plt.grid(True)
#show
plt.show()
```

Output



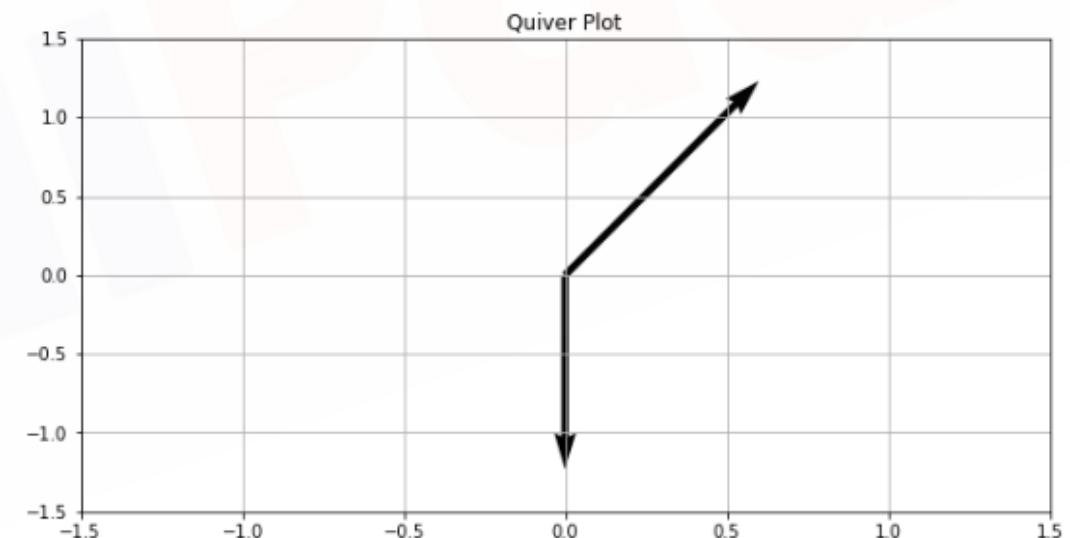
Shows vector lines as arrows, useful in electrical engineering to visualize electrical potential.

Demonstration: Quiver Plot

Input

```
In [69]: #import libraries
import matplotlib.pyplot as plt
%matplotlib inline
#prepare data
x_pos = [0, 0]
y_pos = [0, 0]
x_direct = [1, 0]
y_direct = [1, -1]
#Add figure
fig=plt.figure(figsize=(10,5))
#add axes
ax1 = plt.subplot()
#plot
ax1.quiver(x_pos,y_pos,x_direct,y_direct,scale=5)
#Changing the scale limits
ax1.axis([-1.5, 1.5, -1.5, 1.5])
#Customization-title
plt.title('Quiver Plot')
#customization-add grid
plt.grid(True)
#save the plot
plt.savefig('QuiverPlot.png')
#show plot
plt.show()
```

Output

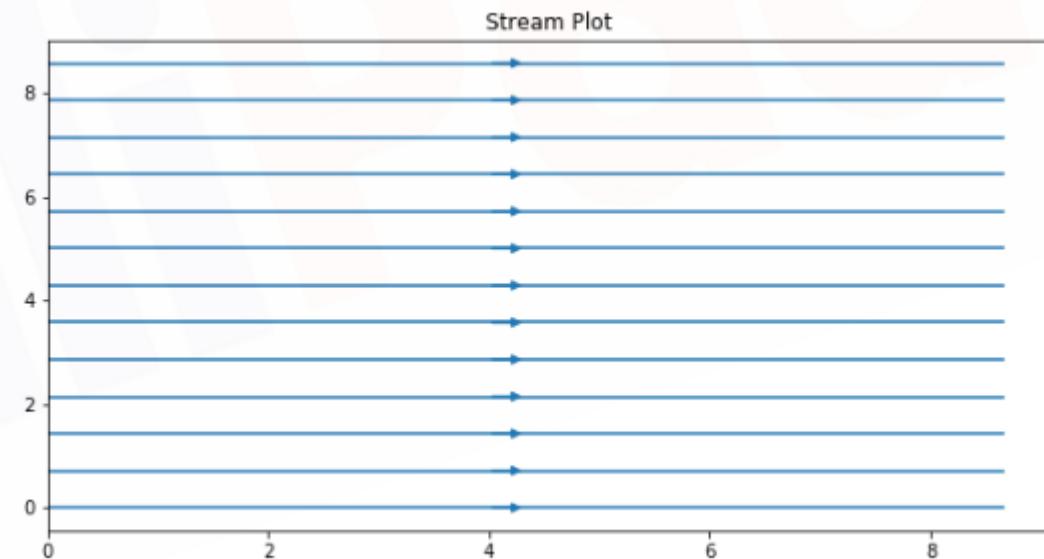


Demonstration: Stream Plot

Input

```
In [71]: #import Libraries
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
#prepare data
x = np.arange(0,10)
y = np.arange(0,10)
X, Y = np.meshgrid(x,y)
u = np.ones((10,10)) # x-component to the right
v = np.zeros((10,10)) # y-component zero
#Add figure
fig=plt.figure(figsize=(10,5))
#add axes
ax1 = plt.subplot()
#plot
ax1.streamplot(X,Y,u,v, density = 0.5)
#Customization-title
plt.title('Stream Plot')
#save the plot
plt.savefig('StreamPlot1.png')
#show plot
plt.show()
```

Output



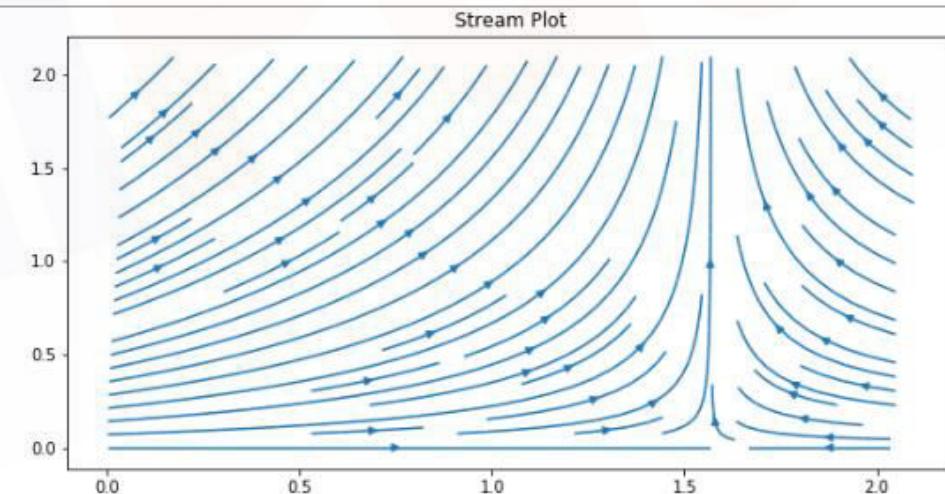
A stream plot is a type of 2D plot used to show fluid flow and 2D field gradients.

Demonstration: Stream Plot

Input

```
In [72]: #import libraries
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
#prepare data
x = np.arange(0,2.2,0.1)
y = np.arange(0,2.2,0.1)
X, Y = np.meshgrid(x, y)
u = np.cos(X)*y
v = np.sin(y)*X
#Add figure
fig=plt.figure(figsize=(10,5))
#add axes
ax1 = plt.subplot()
#plot
ax1.streamplot(X,Y,u,v, density = 1)
#Customization-title
plt.title('Stream Plot')
#save the plot
plt.savefig('StreamPlot2.png')
#show plot
plt.show()
```

Output



A cartoon illustration of a man with brown hair tied back, wearing blue-rimmed sunglasses and a blue shirt. He is peeking over a dark blue wall, with his hand resting on his chin in a thoughtful pose. The background behind the wall is a light, out-of-focus grey.

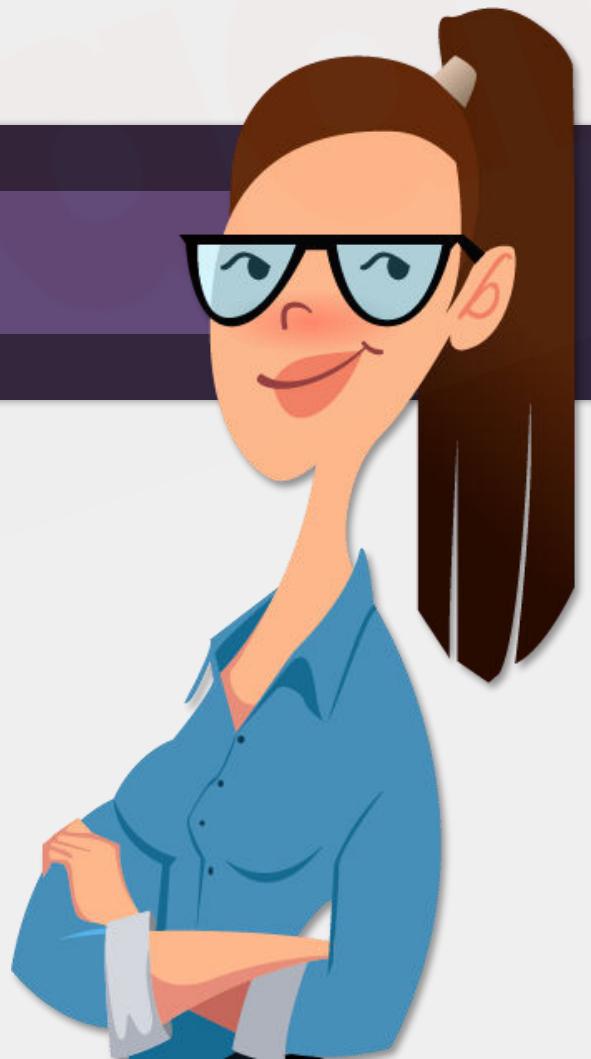
How to create a Pie Chart?

Hands On: Pie Chart

- How to create a Pie Chart?
- How to customize a Pie Chart?
- How to create a Doughnut Chart?



Hands-on: Pie Chart

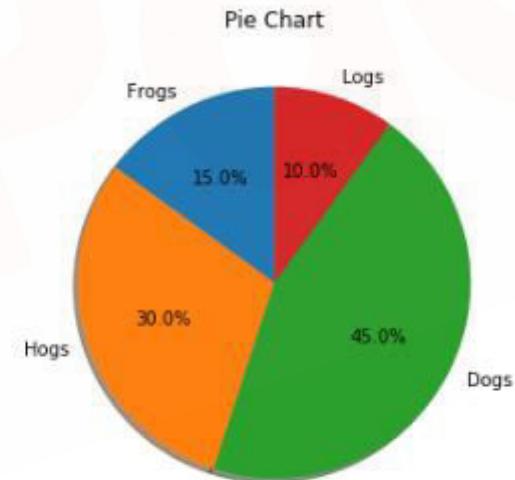


Demonstration: Pie Chart

Input

```
In [73]: #import libraries
import matplotlib.pyplot as plt
%matplotlib inline
#prepare data
labels = ['Frogs', 'Hogs', 'Dogs', 'Logs']
sizes = [15, 30, 45, 10]
#Add figure
fig=plt.figure(figsize=(10,5))
#add axes
ax1 = plt.subplot()
#plot-sezes, labels, autopercantage, shadow, start-angle=90
ax1.pie(sizes, labels=labels, autopct='%1.1f%%', shadow=True, startangle=90)
#Customization-title
plt.title('Pie Chart')
#save the plot
plt.savefig('PieChart.png')
#show plot
plt.show()
```

Output



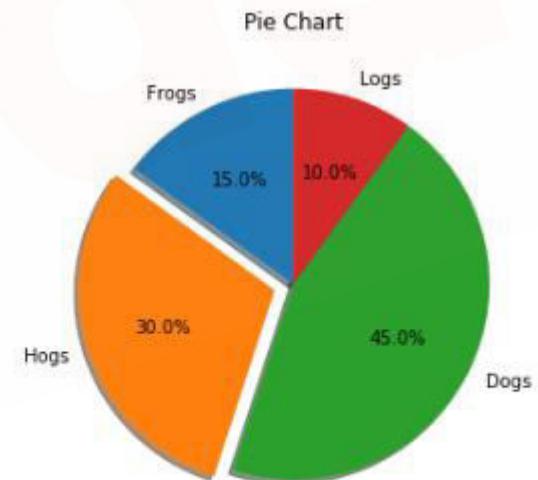
Used to show percentage or proportional data, good for displaying data for around 6 categories or fewer.

Demonstration: Customized Pie Chart

Input

```
In [74]: #import libraries
import matplotlib.pyplot as plt
%matplotlib inline
#prepare data
labels = ['Frogs', 'Hogs', 'Dogs', 'Logs']
sizes = [15, 30, 45, 10]
#add explode if required or else keep 0
explode = (0, 0.1, 0, 0)
#Add figure
fig=plt.figure(figsize=(10,5))
#add axes
ax1 = plt.subplot()
#plot-sizes, labels, autopct, shadow, start-angle=90
ax1.pie(sizes, explode=explode, labels=labels, autopct='%1.1f%%', shadow=True, startangle=90)
#Customization-title
plt.title('Pie Chart')
#save the plot
plt.savefig('PieChart2.png')
#show
plt.show()
```

Output



Demonstration: Donut Chart

Input

```
In [7]: # Libraries
import matplotlib.pyplot as plt

# Make data: I have 3 groups and 7 subgroups
group_names=['groupA', 'groupB', 'groupC']
group_size=[12,11,30]
subgroup_names=['A.1', 'A.2', 'A.3', 'B.1', 'B.2', 'C.1', 'C.2', 'C.3', 'C.4', 'C.5']
subgroup_size=[4,3,5,6,5,10,5,5,4,6]
# Create colors
a, b, c=[plt.cm.Blues, plt.cm.Reds, plt.cm.Greens]

# Add figure and axes
fig, ax = plt.subplots()
ax.axis('equal')

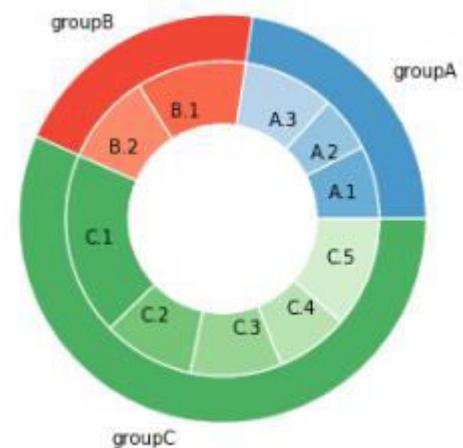
#plot first ring
mypie, _ = ax.pie(group_size, radius=1.3, labels=group_names, colors=[a(0.6), b(0.6), c(0.6)] )
# plot Second Ring (Inside)
mypie2, _ = ax.pie(subgroup_size, radius=1.3-0.3, labels=subgroup_names, labeldistance=0.7, colors=[a(0.5), a(0.4), a(0.3)]

# Customize
plt.setp( mypie, width=0.3, edgecolor='white')
plt.setp( mypie2, width=0.4, edgecolor='white')
plt.margins(0,0)

#save the plot
plt.savefig('NestedPieChart.png')

# show it
plt.show()
```

Output



Donut chart can contain more than one data series. Each data series that you plot in a doughnut chart adds a ring to the chart

A cartoon illustration of a man with brown hair tied back, wearing dark sunglasses and a blue shirt. He is peeking out from behind a dark brown door, with his hand resting on his chin in a thoughtful pose. The background is a light gray.

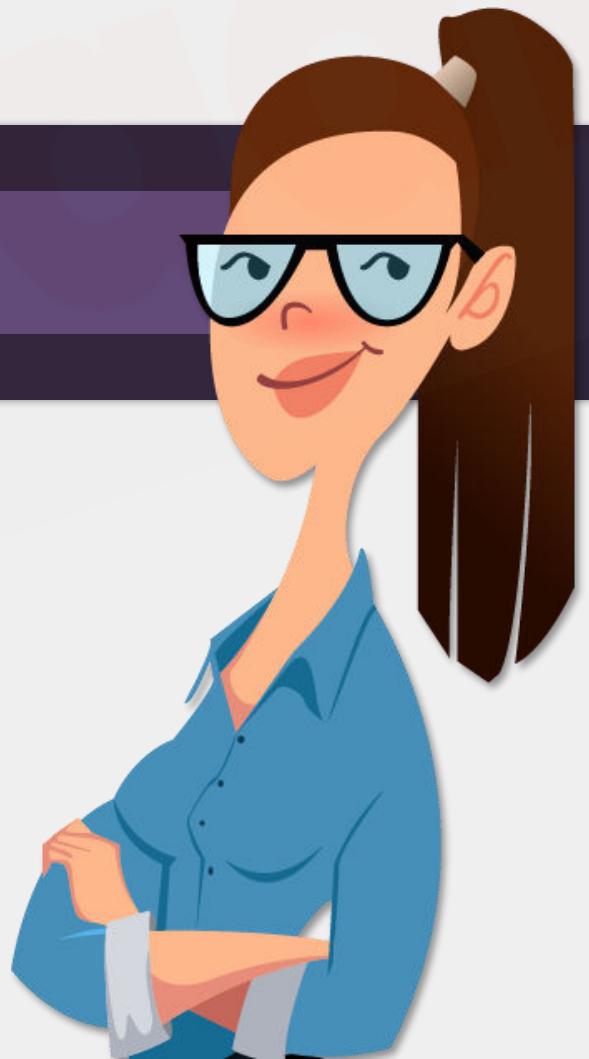
How to create an Area Chart?

Hands On: Area Chart

- How to create an Area Chart?
- How to merge line chart and area chart?



Hands-on: Area Plot

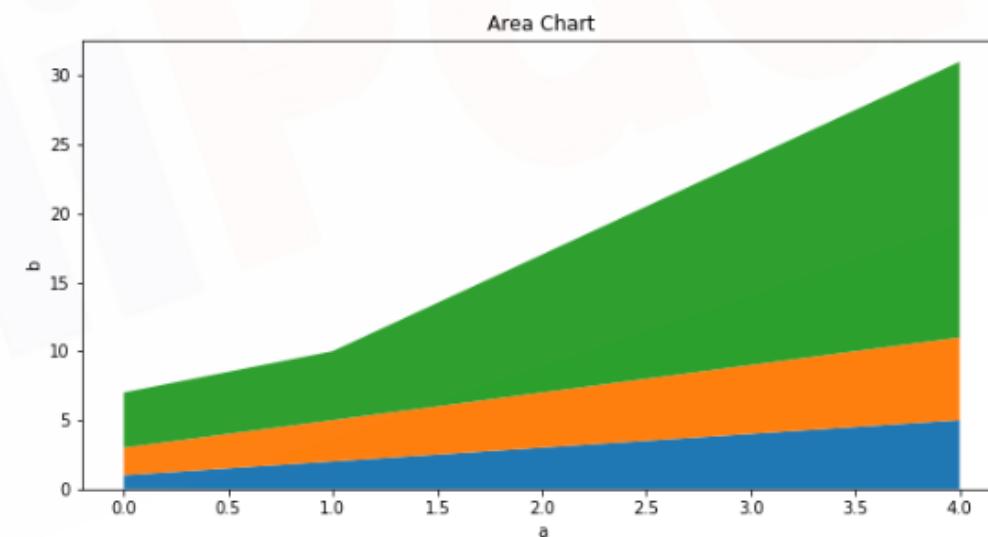


Demonstration: Area Plot

Input

```
In [79]: #import libraries
import matplotlib.pyplot as plt
%matplotlib inline
#prepare data
a= range(0,5)
b= [[1,2,3,4,5],[2,3,4,5,6],[4,5,10,15,20]]
#Add figure
fig=plt.figure(figsize=(10,5))
#add axes
ax1 = plt.subplot()
#plot the area plot
ax1.stackplot(a,b)
#Customization-title
plt.title('Area Chart')
#Customization-x-axis label, y-axis label
plt.xlabel('a')
plt.ylabel('b')
#save the plot
plt.savefig('AreaPlot.png')
#show
plt.show()
```

Output



Used to represent cumulative totals using numbers or percentages over time.

QUIZ

Quiz 1

Which of the following is used to plot horizontal bar plot in matplotlib:

A

`matplotlib.pyplot.bar.horizontal()`

B

`matplotlib.pyplot.bahr()`

C

`matplotlib.pyplot.horizontal.bar()`

D

`matplotlib.pyplot.hbar()`



Answer 1

Which of the following is used to plot horizontal bar plot in matplotlib:

A

`matplotlib.pyplot.bar.horizontal()`

B

`matplotlib.pyplot.bahr()`

C

`matplotlib.pyplot.horizontal.bar()`

D

`matplotlib.pyplot.hbar()`



Quiz 2

We are plotting a line plot between two variables, a and b. To add a legend which of the following will be used:

A

```
matplotlib.pyplot.legend([a,b],[loc= 'best'])
```

B

```
matplotlib.pyplot.legend(['a', 'b'],[loc='best'])
```

C

```
matplotlib.pyplot.legend(['a', 'b', loc='best'])
```

D

```
matplotlib.pyplot.legend([a, b, loc='best'])
```



Answer 2

We are plotting a line plot between two variables, a and b. To add a legend which of the following will be used:

A

```
matplotlib.pyplot.legend([a,b],[loc= 'best'])
```

B

```
matplotlib.pyplot.legend(['a', 'b'],[loc='best'])
```

C

```
matplotlib.pyplot.legend(['a', 'b', loc='best'])
```

D

```
matplotlib.pyplot.legend([a, b, loc='best'])
```



Quiz 3

To add title in the plot which of the following is used?

A

`matplotlib.pyplot.plot(title=['name'])`

B

`matplotlib.pyplot.plot.title('name')`

C

`matplotlib.pyplot.title('name')`

D

`matplotlib.pyplot.title(name)`



Answer 3

To add title in the plot which of the following is used?

A

`matplotlib.pyplot.plot(title=['name'])`

B

`matplotlib.pyplot.plot.title('name')`

C

`matplotlib.pyplot.title('name')`

D

`matplotlib.pyplot.title(name)`



Quiz 4

Which of the following represents categorical data?

A

Area plot

B

Histogram

C

Quiver plot

D

Bar plot



Answer 4

Which of the following represents categorical data?

A

Area plot

B

Histogram

C

Quiver plot

D

Bar plot



Quiz 5

Which of the following is useful to display data with 10 categories?

A

Histogram

B

Bar plot

C

Pie chart

D

Area plot



Answer 5

Which of the following is useful to display data with 10 categories?

A

Histogram

B

Bar plot

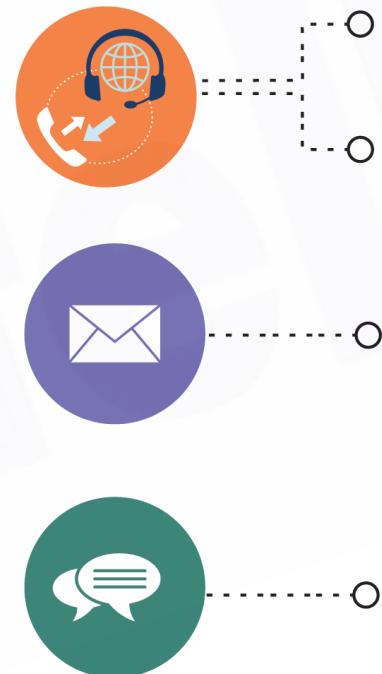
C

Pie chart

D

Area plot





India : +91-7847955955

US : 1-800-216-8930 (TOLL FREE)

sales@intellipaat.com

24X7 Chat with our Course Advisor