Pandas (https://pandas.pydata.org/)

In [1]:

```
import pandas as pd
```

In [2]:

```
#pip install pandas
```

In [3]:

```
import pandas as pd
```

In [4]:

```
s1 = pd.Series(['a','b','c','d'])
s1
```

Out[4]:

```
0    a
1    b
2    c
3    d
dtype: object
```

In [5]:

```
type(s1)
```

Out[5]:

```
pandas.core.series.Series
```

In [6]:

```
s1.index
```

Out[6]:

```
RangeIndex(start=0, stop=4, step=1)
```

In [7]:

```
s1.index = ['num1','num2','num3','num4']
s1
```

Out[7]:

```
num1    a
num2    b
num3    c
num4    d
dtype: object
```

In [8]:

```
s1.dtype
```

Out[8]:

```
dtype('O')
```

In [9]:

```
fruits = ['Apple','Banana','Strawberry','Orange']
veg = ['Carrot','potato','Tomato','beans']
flowers = ['Lily','Sun Flower','Jasmine','Rose']
col = ['Fruits','Vegetables','Flowers']

df = pd.DataFrame([fruits,veg,flowers])
df
```

Out[9]:

	0	1	2	3
0	Apple	Banana	Strawberry	Orange
1	Carrot	potato	Tomato	beans
2	Lily	Sun Flower	Jasmine	Rose

In [10]:

```
type(df)
```

Out[10]:

```
pandas.core.frame.DataFrame
```

In [11]:

```
df.index
```

Out[11]:

```
RangeIndex(start=0, stop=3, step=1)
```

In [12]:

```
df.index = ['Fruits','Vegetables','Flowers']
df
```

Out[12]:

	0	1	2	3
Fruits	Apple	Banana	Strawberry	Orange
Vegetables	Carrot	potato	Tomato	beans
Flowers	Lily	Sun Flower	Jasmine	Rose

In [13]:

```
df.columns
```

Out[13]:

```
RangeIndex(start=0, stop=4, step=1)
```

In [14]:

```
df.columns = ['item1','item2','item3','item4']
df
```

Out[14]:

	item1	item2	item3	item4
Fruits	Apple	Banana	Strawberry	Orange
Vegetables	Carrot	potato	Tomato	beans
Flowers	Lily	Sun Flower	Jasmine	Rose

In [15]:

```
col1 = df['item1']
col1
```

Out[15]:

```
Fruits         Apple
Vegetables    Carrot
Flowers         Lily
Name: item1, dtype: object
```

In [16]:

```
type(col1)
```

Out[16]:

```
pandas.core.series.Series
```

In [17]:

```
col2 = df[['item1','item3']]
type(col2)
```

Out[17]:

```
pandas.core.frame.DataFrame
```

In [18]:

```
col2
```

Out[18]:

	item1	item3
Fruits	Apple	Strawberry
Vegetables	Carrot	Tomato
Flowers	Lily	Jasmine

In [19]:

```
df.sort_values(by = 'item3')
```

Out[19]:

	item1	item2	item3	item4
Flowers	Lily	Sun Flower	Jasmine	Rose
Fruits	Apple	Banana	Strawberry	Orange
Vegetables	Carrot	potato	Tomato	beans

In [20]:

```
df.to_csv('items.csv')
```

In [21]:

```
roll = '18A51A04'
rollNumbers = []
for num in range(1,101):
    if num < 10:
        s = roll + str(0) + str(num)
    else:
        s = roll + str(num)
    rollNumbers.append(s)
print(rollNumbers)
```

```
['18A51A0401', '18A51A0402', '18A51A0403', '18A51A0404', '18A51A0405', '18
A51A0406', '18A51A0407', '18A51A0408', '18A51A0409', '18A51A0410', '18A51A
0411', '18A51A0412', '18A51A0413', '18A51A0414', '18A51A0415', '18A51A041
6', '18A51A0417', '18A51A0418', '18A51A0419', '18A51A0420', '18A51A0421',
'18A51A0422', '18A51A0423', '18A51A0424', '18A51A0425', '18A51A0426', '18A
51A0427', '18A51A0428', '18A51A0429', '18A51A0430', '18A51A0431', '18A51A0
432', '18A51A0433', '18A51A0434', '18A51A0435', '18A51A0436', '18A51A043
7', '18A51A0438', '18A51A0439', '18A51A0440', '18A51A0441', '18A51A0442',
'18A51A0443', '18A51A0444', '18A51A0445', '18A51A0446', '18A51A0447', '18A
51A0448', '18A51A0449', '18A51A0450', '18A51A0451', '18A51A0452', '18A51A0
453', '18A51A0454', '18A51A0455', '18A51A0456', '18A51A0457', '18A51A045
8', '18A51A0459', '18A51A0460', '18A51A0461', '18A51A0462', '18A51A0463',
'18A51A0464', '18A51A0465', '18A51A0466', '18A51A0467', '18A51A0468', '18A
51A0469', '18A51A0470', '18A51A0471', '18A51A0472', '18A51A0473', '18A51A0
474', '18A51A0475', '18A51A0476', '18A51A0477', '18A51A0478', '18A51A047
9', '18A51A0480', '18A51A0481', '18A51A0482', '18A51A0483', '18A51A0484',
'18A51A0485', '18A51A0486', '18A51A0487', '18A51A0488', '18A51A0489', '18A
51A0490', '18A51A0491', '18A51A0492', '18A51A0493', '18A51A0494', '18A51A0
495', '18A51A0496', '18A51A0497', '18A51A0498', '18A51A0499', '18A51A0410
0']
```

In [22]:

```
import numpy as np
python = [np.random.randint(0,100) for i in range(1,101)]
C = [np.random.randint(0,100) for i in range(1,101)]
pandas = [np.random.randint(0,100) for i in range(1,101)]
numpy = [np.random.randint(0,100) for i in range(1,101)]
```

Creating the Email Id's

username@domainName.extension

rollNumber@apssdc.in

In [23]:

```
email = [roll + '@apssdc.in' for roll in rollNumbers]
```

In [24]:

```
data = {'Roll Numbers':rollNumbers,'Email': email,'C':C,
 'Python':python,'Pandas':pandas,'Numpy':numpy}
print(data)
```

{'Roll Numbers': ['18A51A0401', '18A51A0402', '18A51A0403', '18A51A0404',
'18A51A0405', '18A51A0406', '18A51A0407', '18A51A0408', '18A51A0409', '18A
51A0410', '18A51A0411', '18A51A0412', '18A51A0413', '18A51A0414', '18A51A0
415', '18A51A0416', '18A51A0417', '18A51A0418', '18A51A0419', '18A51A042
0', '18A51A0421', '18A51A0422', '18A51A0423', '18A51A0424', '18A51A0425',
'18A51A0426', '18A51A0427', '18A51A0428', '18A51A0429', '18A51A0430', '18A
51A0431', '18A51A0432', '18A51A0433', '18A51A0434', '18A51A0435', '18A51A0
436', '18A51A0437', '18A51A0438', '18A51A0439', '18A51A0440', '18A51A044
1', '18A51A0442', '18A51A0443', '18A51A0444', '18A51A0445', '18A51A0446',
'18A51A0447', '18A51A0448', '18A51A0449', '18A51A0450', '18A51A0451', '18A
51A0452', '18A51A0453', '18A51A0454', '18A51A0455', '18A51A0456', '18A51A0
457', '18A51A0458', '18A51A0459', '18A51A0460', '18A51A0461', '18A51A046
2', '18A51A0463', '18A51A0464', '18A51A0465', '18A51A0466', '18A51A0467',
'18A51A0468', '18A51A0469', '18A51A0470', '18A51A0471', '18A51A0472', '18A
51A0473', '18A51A0474', '18A51A0475', '18A51A0476', '18A51A0477', '18A51A0
478', '18A51A0479', '18A51A0480', '18A51A0481', '18A51A0482', '18A51A048
3', '18A51A0484', '18A51A0485', '18A51A0486', '18A51A0487', '18A51A0488',
'18A51A0489', '18A51A0490', '18A51A0491', '18A51A0492', '18A51A0493', '18A
51A0494', '18A51A0495', '18A51A0496', '18A51A0497', '18A51A0498', '18A51A0
499', '18A51A04100'], 'Email': ['18A51A0401@apssdc.in', '18A51A0402@apssd
c.in', '18A51A0403@apssdc.in', '18A51A0404@apssdc.in', '18A51A0405@apssdc.
in', '18A51A0406@apssdc.in', '18A51A0407@apssdc.in', '18A51A0408@apssdc.i
n', '18A51A0409@apssdc.in', '18A51A0410@apssdc.in', '18A51A0411@apssdc.i
n', '18A51A0412@apssdc.in', '18A51A0413@apssdc.in', '18A51A0414@apssdc.i
n', '18A51A0415@apssdc.in', '18A51A0416@apssdc.in', '18A51A0417@apssdc.i
n', '18A51A0418@apssdc.in', '18A51A0419@apssdc.in', '18A51A0420@apssdc.i
n', '18A51A0421@apssdc.in', '18A51A0422@apssdc.in', '18A51A0423@apssdc.i
n', '18A51A0424@apssdc.in', '18A51A0425@apssdc.in', '18A51A0426@apssdc.i
n', '18A51A0427@apssdc.in', '18A51A0428@apssdc.in', '18A51A0429@apssdc.i
n', '18A51A0430@apssdc.in', '18A51A0431@apssdc.in', '18A51A0432@apssdc.i
n', '18A51A0433@apssdc.in', '18A51A0434@apssdc.in', '18A51A0435@apssdc.i
n', '18A51A0436@apssdc.in', '18A51A0437@apssdc.in', '18A51A0438@apssdc.i
n', '18A51A0439@apssdc.in', '18A51A0440@apssdc.in', '18A51A0441@apssdc.i
n', '18A51A0442@apssdc.in', '18A51A0443@apssdc.in', '18A51A0444@apssdc.i
n', '18A51A0445@apssdc.in', '18A51A0446@apssdc.in', '18A51A0447@apssdc.i
n', '18A51A0448@apssdc.in', '18A51A0449@apssdc.in', '18A51A0450@apssdc.i
n', '18A51A0451@apssdc.in', '18A51A0452@apssdc.in', '18A51A0453@apssdc.i
n', '18A51A0454@apssdc.in', '18A51A0455@apssdc.in', '18A51A0456@apssdc.i
n', '18A51A0457@apssdc.in', '18A51A0458@apssdc.in', '18A51A0459@apssdc.i
n', '18A51A0460@apssdc.in', '18A51A0461@apssdc.in', '18A51A0462@apssdc.i
n', '18A51A0463@apssdc.in', '18A51A0464@apssdc.in', '18A51A0465@apssdc.i
n', '18A51A0466@apssdc.in', '18A51A0467@apssdc.in', '18A51A0468@apssdc.i
n', '18A51A0469@apssdc.in', '18A51A0470@apssdc.in', '18A51A0471@apssdc.i
n', '18A51A0472@apssdc.in', '18A51A0473@apssdc.in', '18A51A0474@apssdc.i
n', '18A51A0475@apssdc.in', '18A51A0476@apssdc.in', '18A51A0477@apssdc.i
n', '18A51A0478@apssdc.in', '18A51A0479@apssdc.in', '18A51A0480@apssdc.i
n', '18A51A0481@apssdc.in', '18A51A0482@apssdc.in', '18A51A0483@apssdc.i
n', '18A51A0484@apssdc.in', '18A51A0485@apssdc.in', '18A51A0486@apssdc.i
n', '18A51A0487@apssdc.in', '18A51A0488@apssdc.in', '18A51A0489@apssdc.i
n', '18A51A0490@apssdc.in', '18A51A0491@apssdc.in', '18A51A0492@apssdc.i
n', '18A51A0493@apssdc.in', '18A51A0494@apssdc.in', '18A51A0495@apssdc.i
n', '18A51A0496@apssdc.in', '18A51A0497@apssdc.in', '18A51A0498@apssdc.i
n', '18A51A0499@apssdc.in', '18A51A04100@apssdc.in'], 'C': [65, 24, 7, 92,
17, 27, 63, 8, 89, 59, 30, 50, 16, 80, 47, 70, 51, 81, 40, 20, 36, 46, 1,
70, 68, 79, 14, 3, 14, 93, 34, 80, 36, 59, 17, 35, 6, 28, 93, 30, 29, 71,
29, 20, 6, 91, 43, 5, 45, 40, 5, 1, 85, 85, 73, 63, 75, 70, 33, 6, 56, 28,
55, 57, 23, 99, 38, 22, 89, 78, 76, 68, 15, 3, 90, 72, 1, 5, 45, 78, 80, 5
0, 55, 51, 71, 8, 17, 4, 41, 37, 77, 47, 66, 79, 2, 28, 46, 72, 2, 9], 'Py
thon': [27, 41, 28, 52, 50, 8, 44, 20, 59, 59, 15, 6, 15, 50, 83, 64, 52,
55, 18, 20, 60, 87, 30, 56, 78, 6, 52, 62, 92, 39, 95, 85, 20, 20, 81, 47,
63, 32, 66, 39, 45, 10, 64, 87, 73, 18, 46, 9, 81, 9, 4, 75, 70, 82, 26, 7

```
3, 48, 76, 84, 88, 41, 39, 55, 72, 3, 22, 61, 94, 37, 15, 72, 84, 32, 61,
6, 95, 79, 54, 45, 50, 86, 61, 95, 4, 26, 9, 27, 16, 29, 37, 82, 90, 45, 5
1, 54, 93, 81, 17, 42, 5], 'Pandas': [84, 7, 52, 46, 34, 40, 65, 54, 2, 9
1, 51, 63, 40, 31, 42, 35, 61, 59, 83, 25, 53, 33, 20, 10, 59, 63, 24, 29,
38, 99, 68, 48, 0, 28, 81, 99, 12, 55, 0, 52, 15, 73, 42, 11, 87, 15, 4, 6
8, 88, 77, 12, 86, 7, 17, 10, 92, 75, 91, 35, 39, 92, 90, 58, 21, 64, 59,
8, 17, 63, 84, 55, 21, 15, 64, 63, 65, 28, 33, 42, 83, 81, 75, 4, 35, 95,
99, 97, 68, 35, 61, 36, 37, 27, 31, 61, 28, 78, 5, 75, 80], 'Numpy': [52,
44, 78, 15, 27, 8, 6, 1, 11, 76, 17, 36, 96, 68, 6, 75, 1, 72, 9, 1, 35, 5
8, 10, 12, 31, 65, 12, 88, 94, 20, 55, 8, 74, 24, 98, 89, 84, 95, 63, 92,
14, 30, 49, 43, 69, 92, 67, 4, 59, 21, 51, 32, 68, 35, 82, 36, 5, 13, 16,
63, 98, 9, 47, 2, 73, 74, 91, 80, 61, 78, 31, 86, 14, 15, 0, 95, 47, 88, 1
2, 90, 75, 31, 10, 27, 17, 5, 12, 18, 97, 64, 42, 62, 79, 20, 97, 56, 86,
96, 61, 59]}
```

In [25]:

```
sdata = pd.DataFrame(data)
```

In [26]:

```
sdata['TotalMarks'] = sdata['C'] + sdata['Python'] + sdata['Numpy'] + sdata['Pandas']
```

In [27]:

```
sdata['Percentage'] = (sdata['TotalMarks'] * 100) / 400
```

In [28]:

```
sdata['Status'] = ['Pass' if data > 40 else 'Fail' for data in sdata['Percentage']]
```

In [29]:

```
# # Exporting the data as excel file to memory
sdata.to_excel('sdata.xlsx',sheet_name='Class1')
```

In [30]:

```
# Exporting the data as csv file to memory
sdata.to_csv('sdata.csv')
```

In [31]:

```
# Import data Python
rdata = pd.read_csv('sdata.csv',index_col = 0)
```

Getting the top rows

In [32]:

```
rdata.head(6)
```

Out[32]:

	Roll Numbers	Email	C	Python	Pandas	Numpy	TotalMarks	Percentage
0	18A51A0401	18A51A0401@apssdc.in	65	27	84	52	228	57.00
1	18A51A0402	18A51A0402@apssdc.in	24	41	7	44	116	29.00
2	18A51A0403	18A51A0403@apssdc.in	7	28	52	78	165	41.25
3	18A51A0404	18A51A0404@apssdc.in	92	52	46	15	205	51.25
4	18A51A0405	18A51A0405@apssdc.in	17	50	34	27	128	32.00
5	18A51A0406	18A51A0406@apssdc.in	27	8	40	8	83	20.75

Getting the Bottom data

In [33]:

```
rdata.tail(6)
```

Out[33]:

	Roll Numbers	Email	C	Python	Pandas	Numpy	TotalMarks	Percenta
94	18A51A0495	18A51A0495@apssdc.in	2	54	61	97	214	53
95	18A51A0496	18A51A0496@apssdc.in	28	93	28	56	205	51
96	18A51A0497	18A51A0497@apssdc.in	46	81	78	86	291	72
97	18A51A0498	18A51A0498@apssdc.in	72	17	5	96	190	47
98	18A51A0499	18A51A0499@apssdc.in	2	42	75	61	180	45
99	18A51A04100	18A51A04100@apssdc.in	9	5	80	59	153	38

Getting the Column data

In [34]:

```
rdata['Email'].head()
```

Out[34]:

```
0    18A51A0401@apssdc.in
1    18A51A0402@apssdc.in
2    18A51A0403@apssdc.in
3    18A51A0404@apssdc.in
4    18A51A0405@apssdc.in
Name: Email, dtype: object
```

Getting the row

In [35]:

```
rdata.iloc[50]
```

Out[35]:

```
Roll Numbers              18A51A0451
Email           18A51A0451@apssdc.in
C                                  5
Python                             4
Pandas                            12
Numpy                             51
TotalMarks                        72
Percentage                        18
Status                          Fail
Name: 50, dtype: object
```

In [36]:

```
rdata.iloc[10:20]
```

Out[36]:

	Roll Numbers	Email	C	Python	Pandas	Numpy	TotalMarks	Percentage
10	18A51A0411	18A51A0411@apssdc.in	30	15	51	17	113	28.25
11	18A51A0412	18A51A0412@apssdc.in	50	6	63	36	155	38.75
12	18A51A0413	18A51A0413@apssdc.in	16	15	40	96	167	41.75
13	18A51A0414	18A51A0414@apssdc.in	80	50	31	68	229	57.25
14	18A51A0415	18A51A0415@apssdc.in	47	83	42	6	178	44.50
15	18A51A0416	18A51A0416@apssdc.in	70	64	35	75	244	61.00
16	18A51A0417	18A51A0417@apssdc.in	51	52	61	1	165	41.25
17	18A51A0418	18A51A0418@apssdc.in	81	55	59	72	267	66.75
18	18A51A0419	18A51A0419@apssdc.in	40	18	83	9	150	37.50
19	18A51A0420	18A51A0420@apssdc.in	20	20	25	1	66	16.50

In [37]:

```
rdata.shape
```

Out[37]:

```
(100, 9)
```

In [38]:

```
rdata.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 100 entries, 0 to 99
Data columns (total 9 columns):
Roll Numbers    100 non-null object
Email           100 non-null object
C               100 non-null int64
Python          100 non-null int64
Pandas          100 non-null int64
Numpy           100 non-null int64
TotalMarks      100 non-null int64
Percentage      100 non-null float64
Status          100 non-null object
dtypes: float64(1), int64(5), object(3)
memory usage: 7.8+ KB
```

In [39]:

```
rdata.describe()
```

Out[39]:

	C	Python	Pandas	Numpy	TotalMarks	Percentage
count	100.00000	100.000000	100.000000	100.000000	100.000000	100.000000
mean	44.93000	49.110000	49.130000	47.600000	190.770000	47.692500
std	29.03055	27.621502	28.554628	32.127587	58.670085	14.667521
min	1.00000	3.000000	0.000000	0.000000	61.000000	15.250000
25%	19.25000	26.000000	27.750000	15.000000	154.500000	38.625000
50%	45.00000	50.000000	51.500000	48.000000	192.500000	48.125000
75%	71.00000	73.000000	73.500000	75.250000	231.000000	57.750000
max	99.00000	95.000000	99.000000	98.000000	327.000000	81.750000

In [40]:

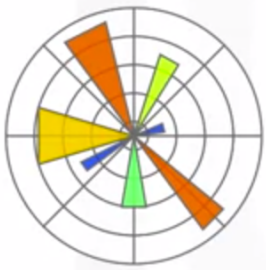
```
rdata['Percentage'].plot(kind = 'hist')
```

Out[40]:

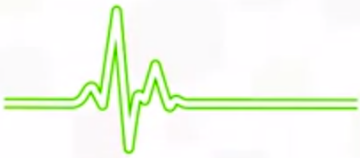
```
<matplotlib.axes._subplots.AxesSubplot at 0x17ad370dc18>
```

Matplotlib (https://matplotlib.org)

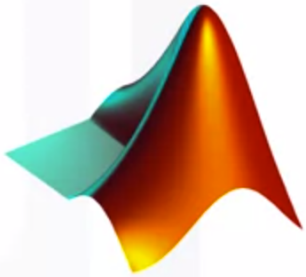
Matplotlib - History



John Hunter (1968 – 2012)



EEG/ECoG Visualization Tool



Analogous to Matlab scripting interface

Matplotlib Architecture

1. Scripting Layer (pyplot)
2. Artist Layer (Artist)
3. Backend Layer (FigureCanvas, Renderer, Event)

Basic and Specialized Visualization Tools

- Line Plots
- Scatter Plots
- Histograms
- Bar Charts
- Pie Charts
- Box Plots

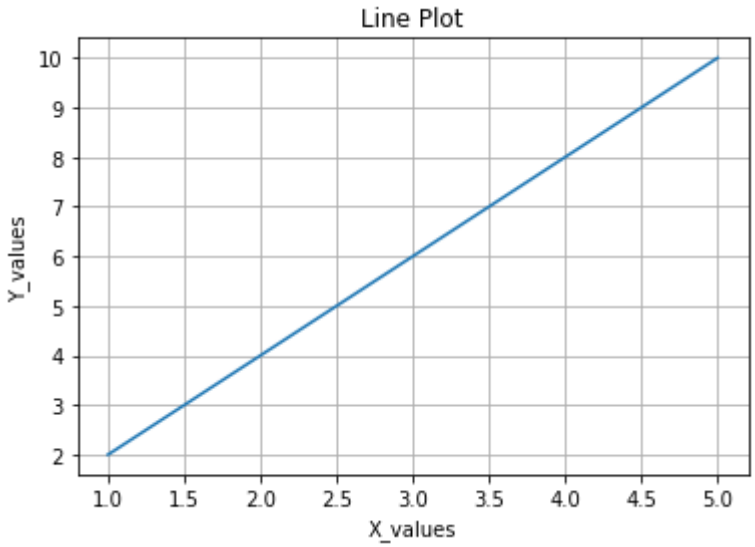
In [41]:

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

Line Plot

In [42]:

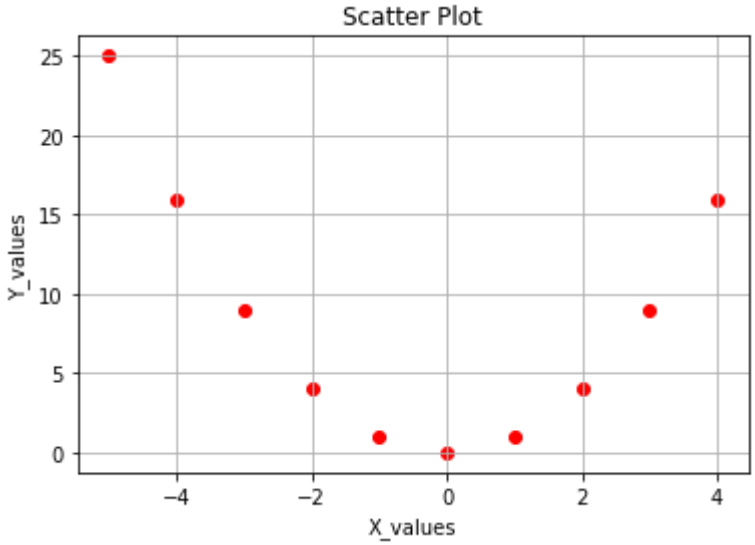
```
x = [1,2,3,4,5]
y = [2,4,6,8,10]

plt.plot(x,y)
plt.title('Line Plot')
plt.xlabel('X_values')
plt.ylabel('Y_values')
plt.grid()
plt.show()
```



Scatter Plot

In [43]:

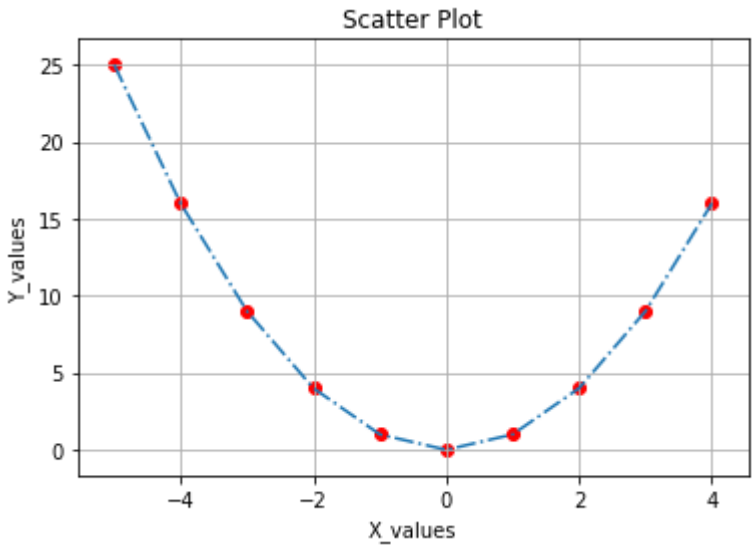
```
import numpy as np
x = np.array([i for i in range(-5,5)])
y = x ** 2

plt.scatter(x,y,color = 'r')
plt.title('Scatter Plot')
plt.xlabel('X_values')
plt.ylabel('Y_values')
plt.grid()
plt.show()
```



Overlapping of two plots

In [44]:

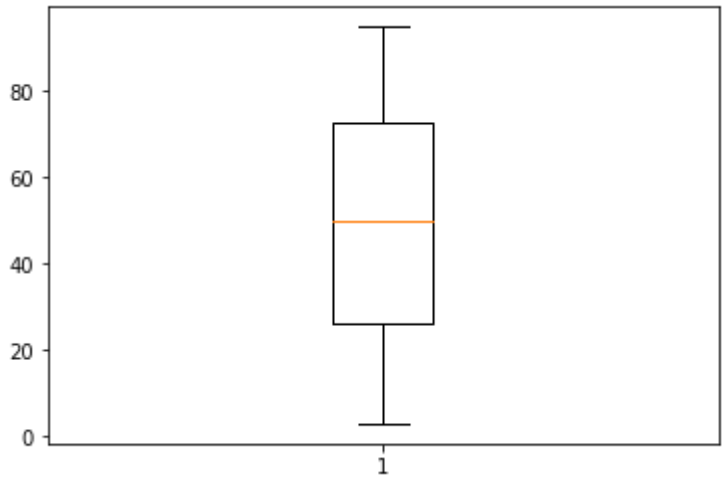
```
import numpy as np
x = np.array([i for i in range(-5,5)])
y = x ** 2

plt.plot(x,y,'-.')
plt.scatter(x,y,color = 'r')
plt.title('Scatter Plot')
plt.xlabel('X_values')
plt.ylabel('Y_values')
plt.grid()
plt.show()
```



Box Plot

In [45]:

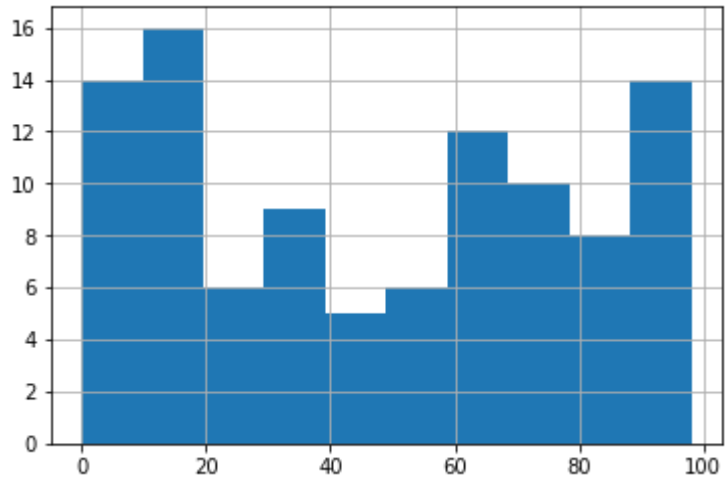
```
import pandas as pd
rdata = pd.read_csv('sdata.csv',index_col = 0)
x = rdata['Python']

plt.boxplot(x)
plt.show()
```



Histogram

In [46]:

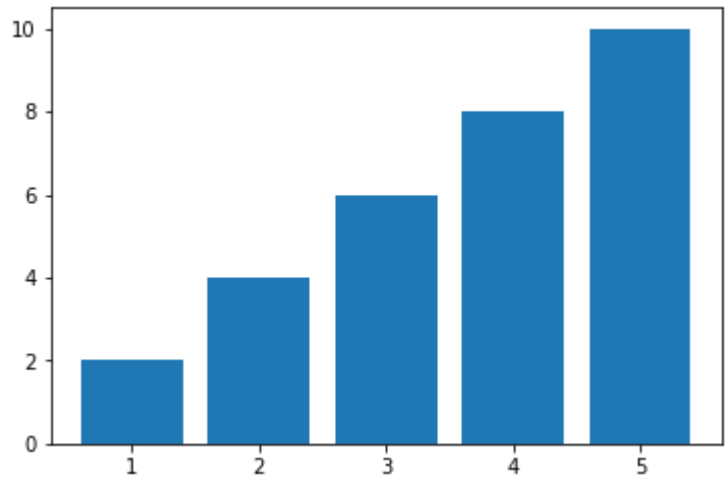
```
x = rdata['Numpy']
plt.hist(x)
plt.grid()
plt.show()
```



Bar Graph

In [47]:

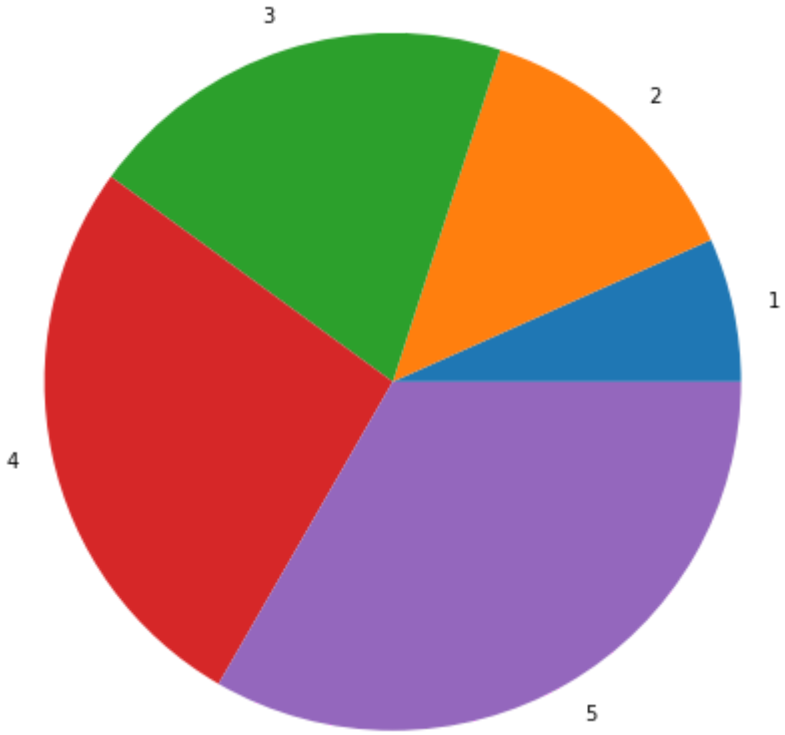
```
x = [1,2,3,4,5]
y = [2,4,6,8,10]
plt.bar(x,y)
plt.show()
```



Pie Charts

In [48]:

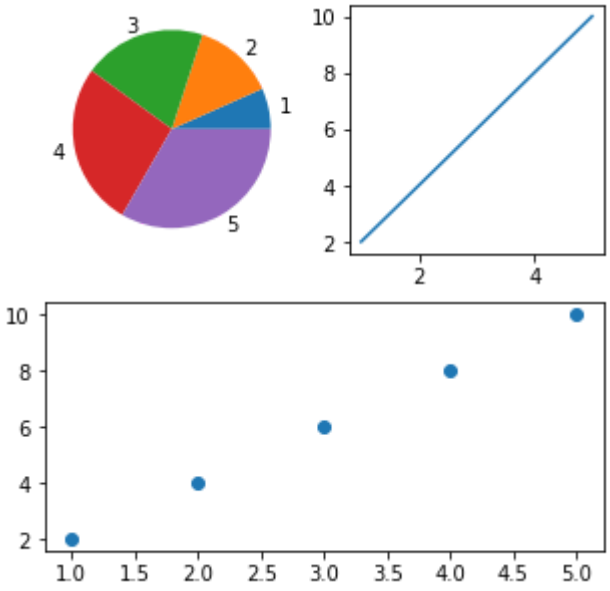
```
x = [1,2,3,4,5]
plt.pie(x,labels=x,radius=2)
plt.savefig('pie.png')
```



Sub Plots

In [49]:

```
x = [1,2,3,4,5]
y = [2,4,6,8,10]
plt.figure(figsize=(5,5))
plt.subplot(2,2,1)
plt.pie(x,labels=x)

plt.subplot(2,2,2)
plt.plot(x,y)

plt.subplot(2,1,2)
plt.scatter(x,y)

plt.show()
```

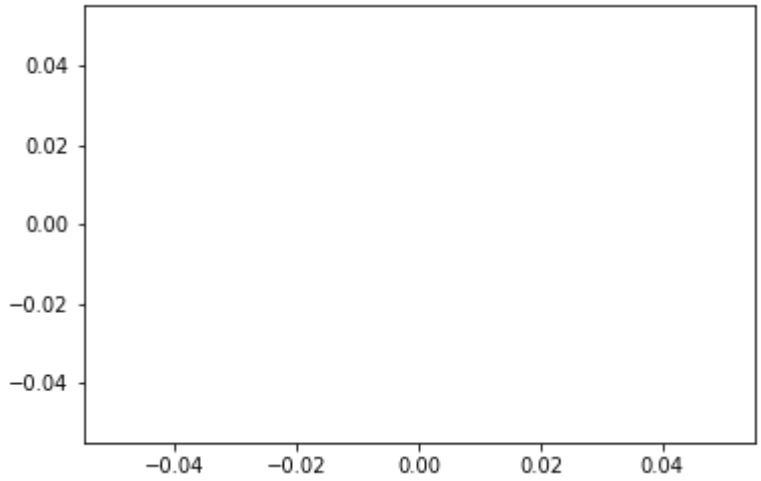


In [50]:

```
# only canvas
plt.plot()
```

Out[50]:

```
[]
```



http://bit.ly/AITAM-Python-Adv-Feb-2020 (http://bit.ly/AITAM-Python-Adv-Feb-2020)

http://bit.ly/APSSDC-Feedback (http://bit.ly/APSSDC-Feedback)

www.awseducate.com