

# Matplotlib 02\_\_ Pengenalan pyplot

June 4, 2022

## 0.1 1. Pengenalan pyplot

1. pyplot merupakan koleksi atau kumpulan fungsi yang menjadikan Matplotlib dapat bekerja dengan mudah
2. pyplot API secara umum lebih fleksibel bila dibandingkan object-oriented API

```
[1]: %matplotlib inline
```

```
[2]: import matplotlib
import matplotlib.pyplot as plt
import numpy as np

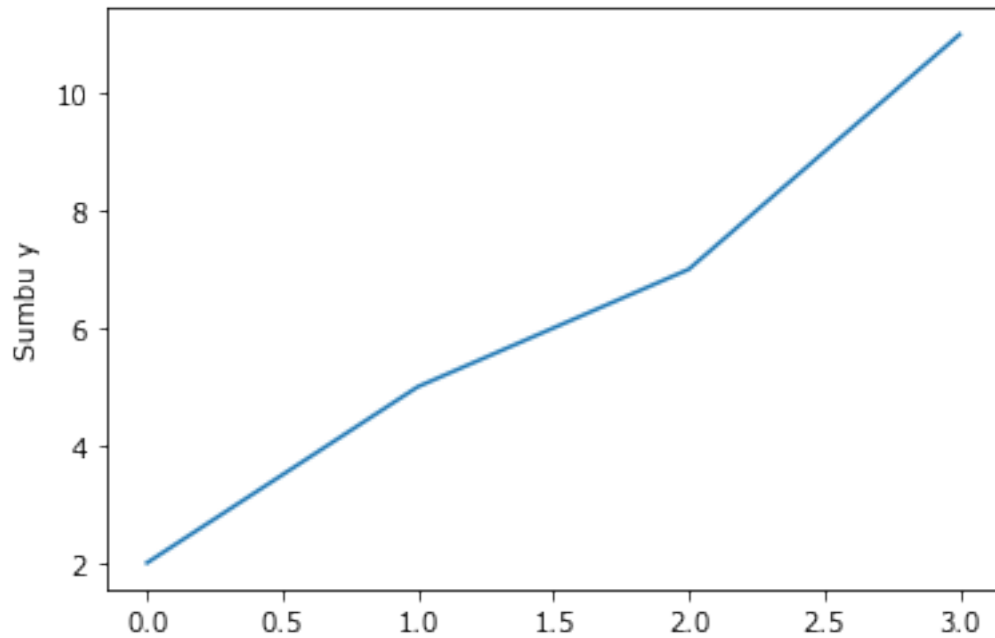
print(matplotlib.__version__)
print(np.__version__)
```

3.3.4

1.20.1

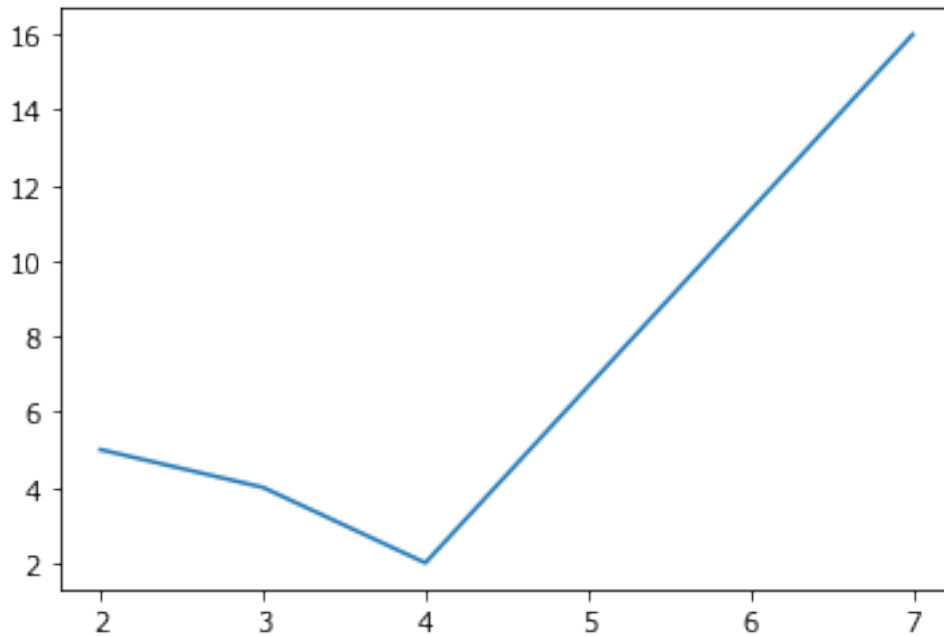
## 0.2 2. Membuat plotting sederhana

```
[3]: plt.plot([2, 5, 7, 11])
plt.ylabel('Sumbu y')
plt.show() # Tidak akan menampilkan teks tambahan diatas figure
```



1. Pemanggilan fungsi `plot()` dapat dilakukan dengan menyertakan hanya sebuah deret bilangan (list/array) ataupun dua buah deret bilangan (list/array).
2. Bila pemanggilan hanya menyertakan sebuah deret bilangan, maka nilai pada deret bilangan tersebut akan dijadikan data point untuk nilai pada sumbu y; sedangkan nilai pada sumbu x akan secara otomatis dibuatkan sebuah deret bilangan terurut dengan nilai dimulai dari nol.

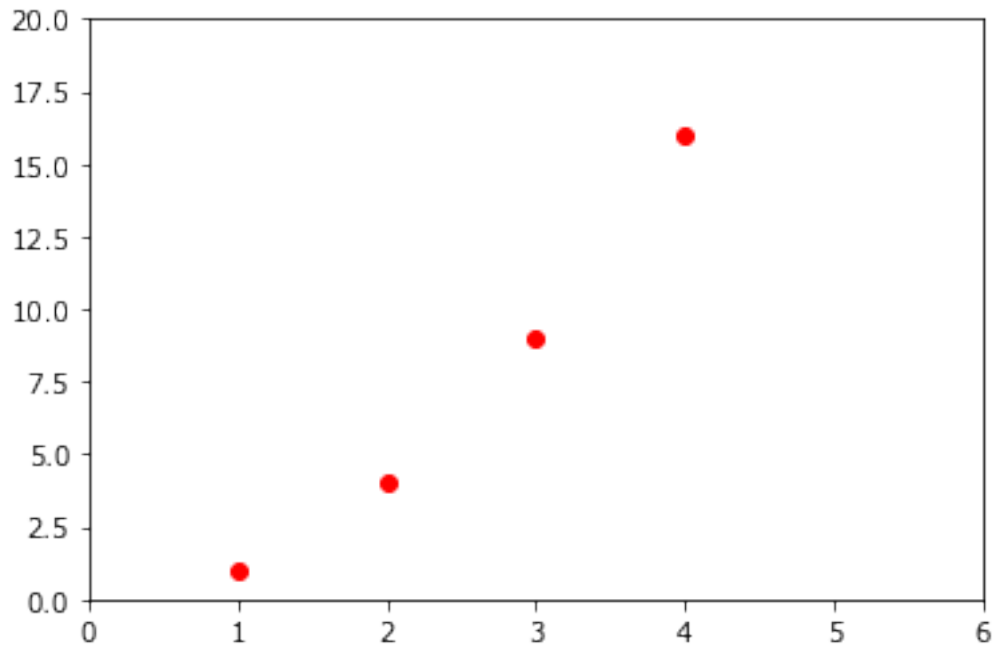
```
[4]: plt.plot([2, 3, 4, 7], [5, 4, 2, 16])  
plt.show()
```



### 0.3 3. Pengaturan Format pada plot

1. Selain menyertakan dua buah deret bilangan sebagai parameter untuk sumbu x dan y, terdapat parameter ketiga yang bisa kita sertakan untuk mengatur warna dan jenis plotting.
2. Pengaturan parameter ketiga ini sepenuhnya mengadopsi gaya formatting pada MATLAB.
3. Nilai default untuk parameter ketiga ini adalah 'b-'.
4. Pemanggilan fungsi `axis()` menyertakan sebuah list `[xmin, xmax, ymin, ymax]`.

```
[5]: plt.plot([1, 2, 3, 4], [1, 4, 9, 16], 'ro')  
plt.axis([0, 6, 0, 20])  
plt.show()
```



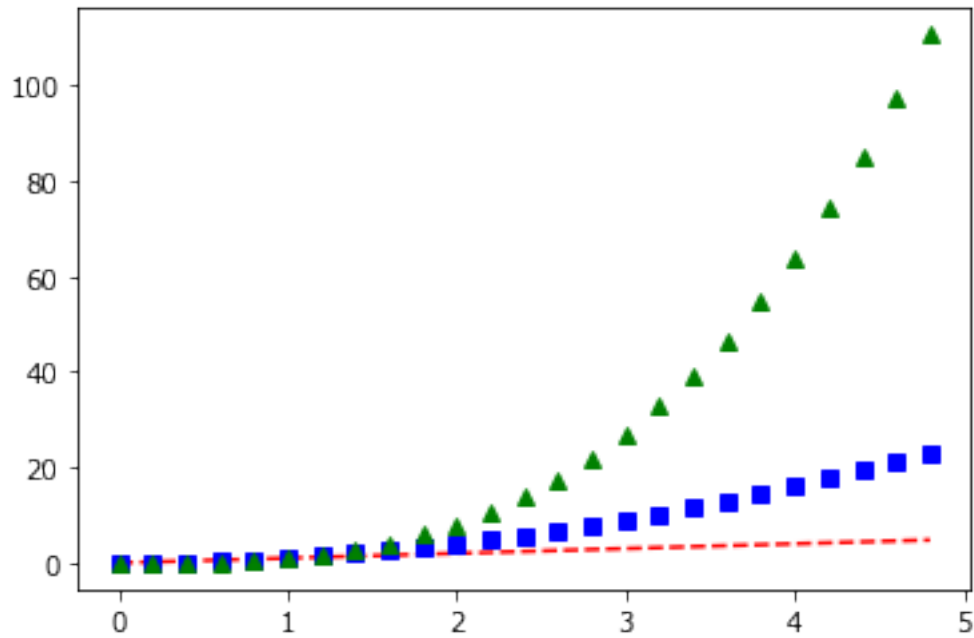
Catatan Penting: 1. 'ro' artinya 'r' menyatakan warna merah (red), 'o' menyatakan jenis plotting

#### 0.4 4. Multi plot dalam satu pemanggilan fungsi plot()

```
[6]: t = np.arange(0., 5., 0.2)
t
```

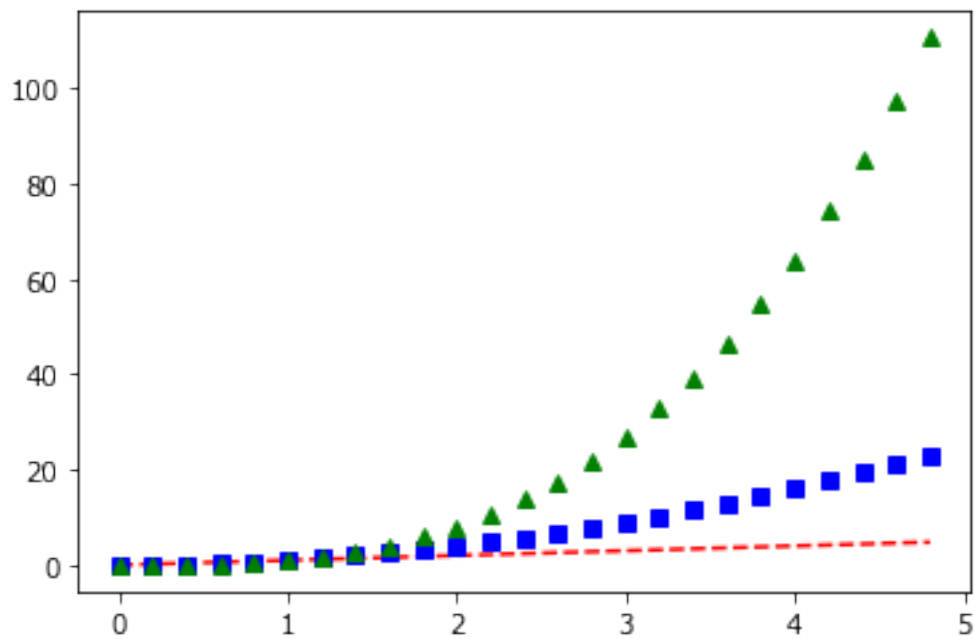
```
[6]: array([0. , 0.2, 0.4, 0.6, 0.8, 1. , 1.2, 1.4, 1.6, 1.8, 2. , 2.2, 2.4,
        2.6, 2.8, 3. , 3.2, 3.4, 3.6, 3.8, 4. , 4.2, 4.4, 4.6, 4.8])
```

```
[7]: plt.plot(t, t, 'r--') # warna merah dan garis putus-putus
plt.plot(t, t**2, 'bs') # warnah hitam dan berbentuk kotak (square)
plt.plot(t, t**3, 'g^') # warna hijau dan berbentuk segitiga
plt.show()
```



Kita juga dapat melakukan multi plot dalam satu pemanggilan fungsi plot()

```
[8]: plt.plot(t, t, 'r--',
             t, t**2, 'bs',
             t, t**3, 'g^')
plt.show()
```



## 0.5 5. Plotting dengan keywords

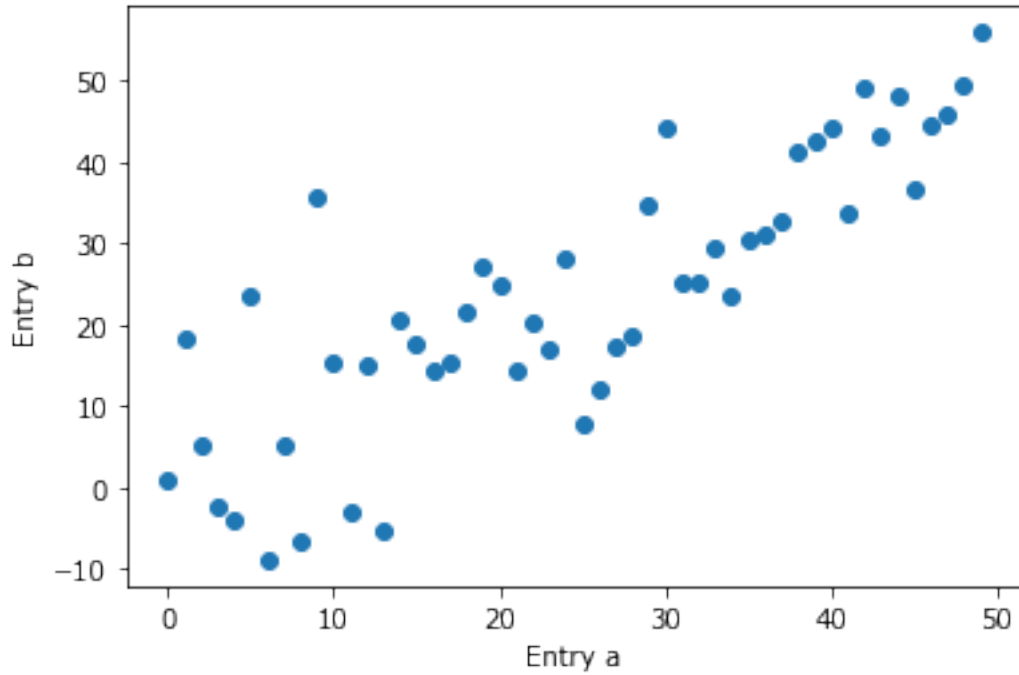
Matplotlib juga memungkinkan kita untuk melakukan plotting sekumpulan nilai yang tersimpan dalam struktur data yang disertai keywords argument (kwargs) seperti pada dictionary dan Pandas Dataframe.

```
[9]: data = {'a': np.arange(50),
            'c': np.random.randint(0, 50, 50),
            'd': np.random.randn(50)}
data['b'] = data['a'] + 10 * np.random.randn(50)
data['d'] = np.abs(data['d']) * 100

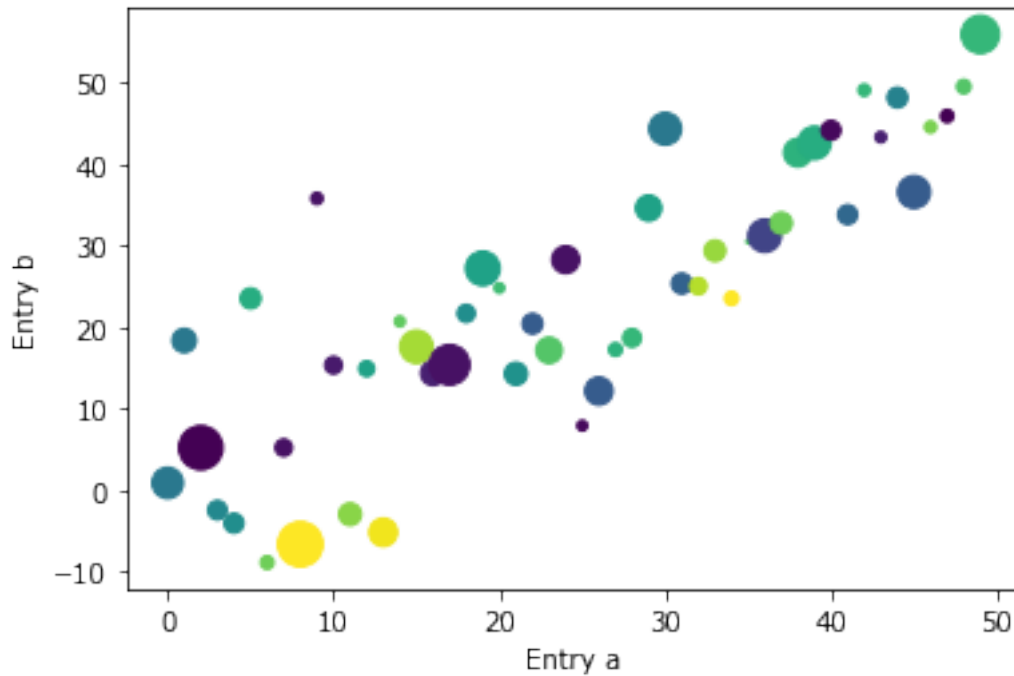
data

[9]: {'a': array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
                17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
                34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49]),
      'c': array([19, 19,  1, 22, 23, 29, 36,  3, 46,  3,  4, 38, 27, 45, 35, 40,  5,
                3, 23, 27, 32, 24, 14, 34,  3,  2, 14, 30, 31, 27, 19, 15, 41, 39,
                46, 34, 10, 36, 30, 29,  2, 16, 31,  5, 21, 14, 37,  1, 34, 31]),
      'd': array([132.55506179,  78.73963144, 267.5107938 ,  49.42353947,
                51.78256026,  57.65053776,  23.75235603,  39.45317953,
                282.3290094 ,  19.08828435,  40.50605828,  68.59759298,
                32.7221886 , 109.09285571,  14.92903946, 150.37995415,
                82.60392719, 224.79205894,  42.70739622, 162.54943621,
                13.63400917,  70.56102772,  56.19909721,  93.27586248,
                103.68038761,  14.36628874, 106.57275764,  23.96932332,
                45.8440484 ,  91.46837385, 144.48982048,  57.68970851,
                38.85216129,  61.31679924,  26.02591279,   1.9244637 ,
                150.78176642,  64.06867893, 109.87447868, 150.59229779,
                50.9214952 ,  51.98482933,  19.4534196 ,  16.47420743,
                56.21994043, 146.67890802,  18.77420482,  22.60021063,
                27.00952812, 200.76963841]),
      'b': array([ 0.87412564, 18.36714589,  5.19337459, -2.46970761, -4.07147685,
                23.54379586, -8.90598751,  5.2062485 , -6.65872644, 35.81823555,
                15.33206276, -2.97075258, 14.90722435, -5.19379152, 20.70751637,
                17.59234688, 14.35404602, 15.37672243, 21.69618588, 27.22364881,
                24.81388566, 14.27857821, 20.42741498, 17.15698735, 28.29918277,
                 7.91004966, 12.15986827, 17.2588187 , 18.68673515, 34.63531981,
                44.37501448, 25.35979184, 25.04141266, 29.39638433, 23.53997123,
                30.52397516, 31.24627137, 32.78829046, 41.44492281, 42.6442848 ,
                44.19837729, 33.82735414, 49.12499041, 43.3524414 , 48.20759726,
                36.60982467, 44.59406069, 45.94217351, 49.5582645 , 55.97193798])}
```

```
[10]: plt.scatter('a', 'b', data = data)
plt.xlabel('Entry a')
plt.ylabel('Entry b')
plt.show()
```



```
[11]: plt.scatter('a', 'b', c = 'c', s = 'd', data = data)
plt.xlabel('Entry a')
plt.ylabel('Entry b')
plt.show()
```



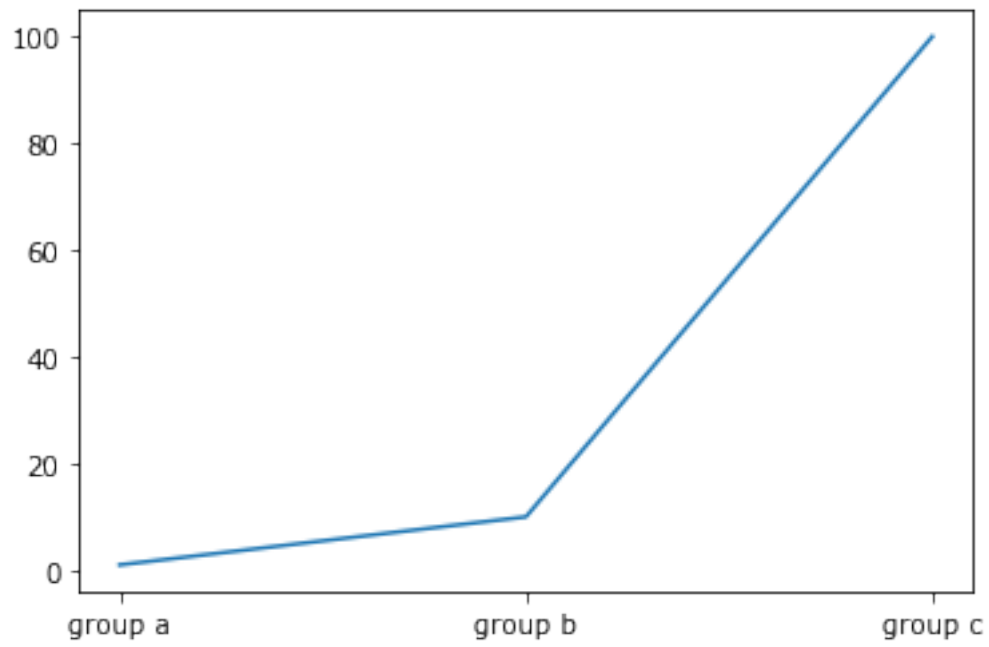
## 0.6 6. Plotting untuk tipe data categorical

Matplotlib juga mendukung plotting untuk data dengan tipe data kategori atau biasa dikenal dengan istilah categorical data categorical variables.

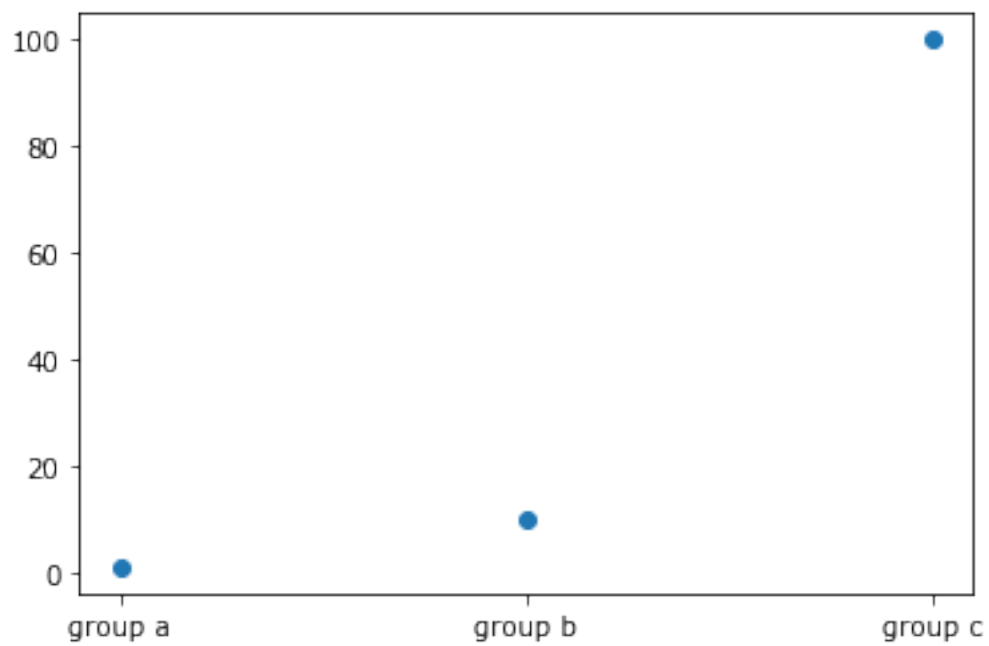
```
[12]: names = ['group a', 'group b', 'group c']  
      values = [1, 10, 100]
```

```
[13]: plt.plot(names, values)  
      plt.show()
```





```
[14]: plt.scatter(names, values)  
plt.show()
```



```
[15]: plt.bar(names, values)  
plt.show()
```

