```
import numpy as np
import matplotlib.pyplot as plt
```

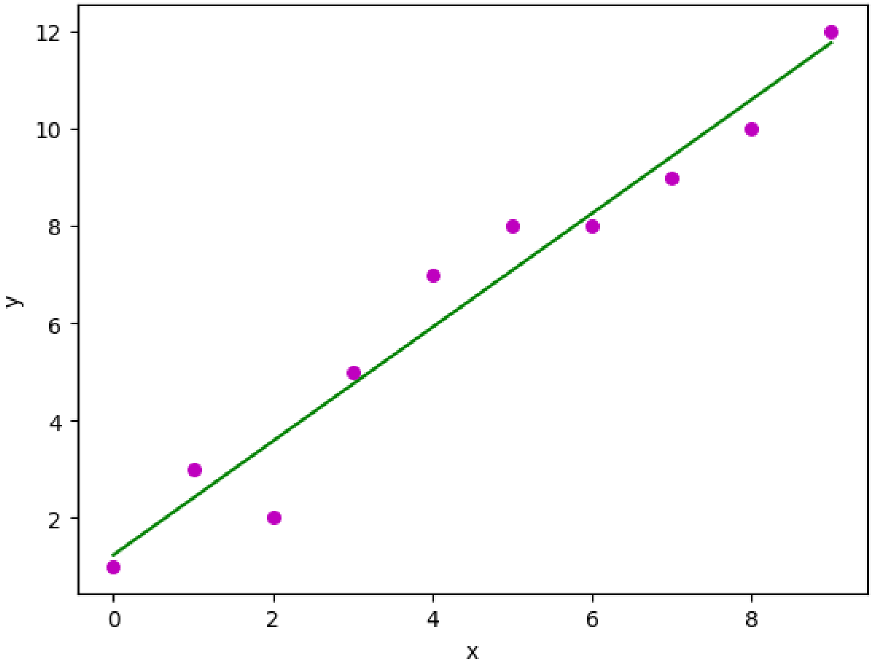
```
def estimate_coef(x, y):
    n = np.size(x) # number of observations/points
    m_x = np.mean(x)  # mean of x and y vector
    m_y = np.mean(y)
    SS_xy = np.sum(y*x) - n*m_y*m_x   # calculating cross-deviation and deviation about x
    SS_xx = np.sum(x*x) - n*m_x*m_x
    b_1 = SS_xy / SS_xx
    b_0 = m_y - b_1*m_x  # calculating regression coefficients
    return (b_0, b_1)
```

```
def plot_regression_line(x, y, b):
    plt.scatter(x, y, color="m", marker="o", s=30)  # plotting the actual points as a scatter plot
    y_pred = b[0] + b[1]*x   # predicted response vector
    plt.plot(x, y_pred, color="g")      # plotting the regression line
    plt.xlabel('x')      # putting labels
    plt.ylabel('y')
    plt.show()     # function to show plot
```

```
def main():
    x = np.array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])   # observations / data
    y = np.array([1, 3, 2, 5, 7, 8, 8, 9, 10, 12])
    b = estimate_coef(x, y)      # estimating coefficients
    print("Estimated coefficients:\nb_0 = {} \nb_1 = {}".format(b[0], b[1]))
    plot_regression_line(x, y, b)      # plotting regression line
```

```
if __name__ == "__main__":
    main()
```

```
Estimated coefficients:
b_0 = 1.2363636363636363
b_1 = 1.1696969696969697
```


12
10
8
6
4
2
y
0
2
4
6
8
x