linear-regression-ols

November 10, 2024

[1]:
```
# s1:calculate the mean of x and y
# s2:calculate the error of x and y
# s3:get the product
# s4:get the summation of the product
# s5:square the difference of the x
# s6:get the sum of the squared difference
# s7:divide output of step 4 by the output of step 6
# s8:calculate 'a' using the value of 'b'
```

[1]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

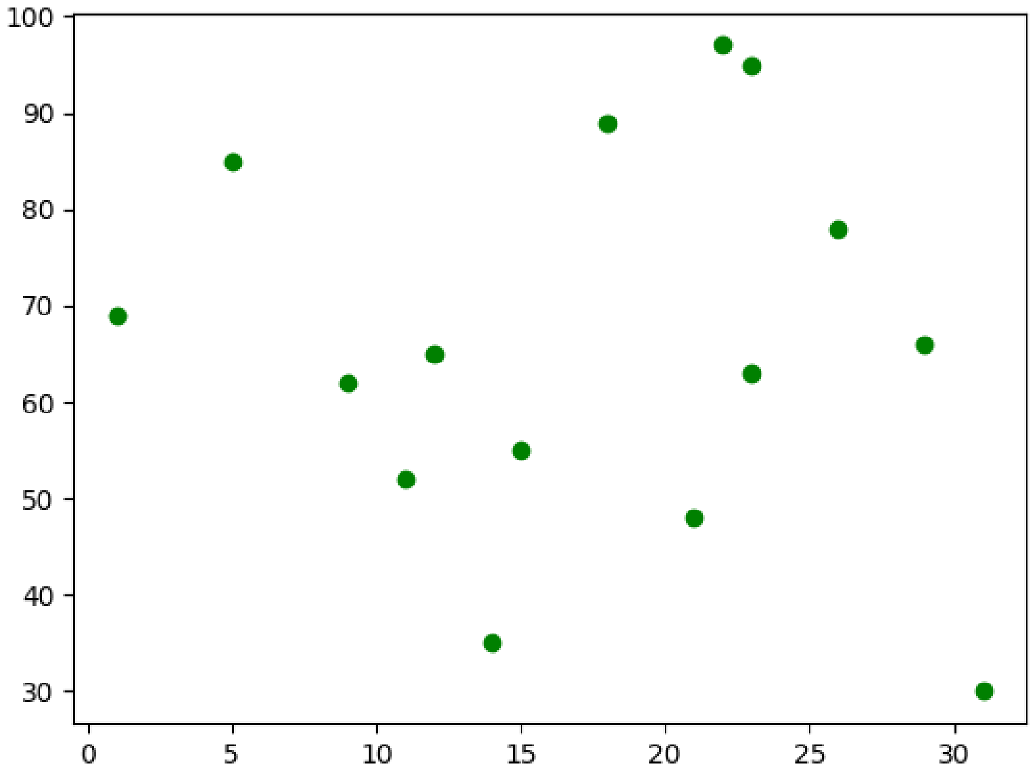
[6]:
```
data = pd.read_excel("Book1.xlsx")
```

[7]:
```
data
```

[7]:
```
     x   y
0   23  63
1    1  69
2   15  55
3   18  89
4   22  97
5   11  52
6    9  62
7   21  48
8   26  78
9    5  85
10  12  65
11  23  95
12  29  66
13  31  30
14  14  35
```

[12]:
```
plt.plot(data.x,data.y,'o',color='g')
```

[12]:
```
[<matplotlib.lines.Line2D at 0x26ae9518140>]
```



```
[13]: x_mean = np.mean(data.x)
```

```
[14]: print(x_mean)
```

```
17.333333333333332
```

```
[15]: y_mean = np.mean(data.y)
```

```
[16]: print(y_mean)
```

```
65.93333333333334
```

```
[18]: numerator = np.sum((data.x-x_mean)*(data.y-y_mean))
      print(numerator)
```

```
-173.66666666666669
```

```
[19]: denominator = np.sum((data.x-x_mean)**2)
      print(denominator)
```

```
1071.3333333333333
```

[20]:

```
b = numerator/denominator
print(b)
```

```
-0.16210329807093968
```

[21]:

```
a = y_mean - (b * x_mean)
print(a)
```

```
68.74312383322962
```

[24]:

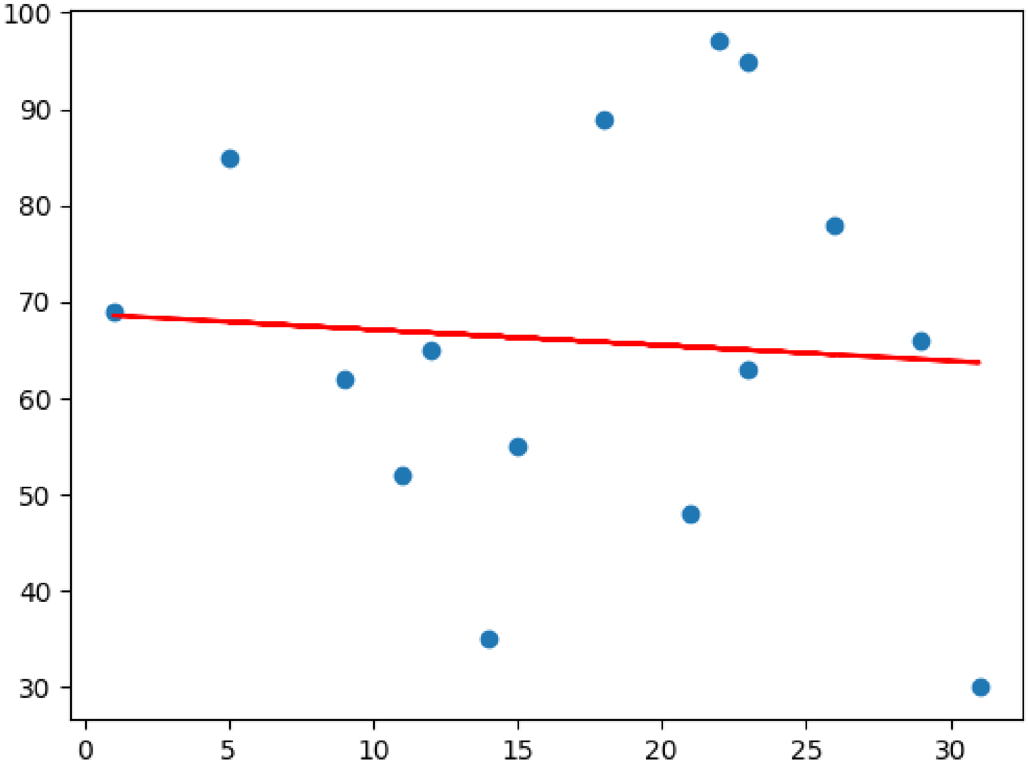
```
def BestFitLine(x,a,b):
    return (b*x + a)
```

[25]:

```
yPred = []
for i in range(len(data['x'])):
    yPred.append(BestFitLine(data.x[i],a,b))
```

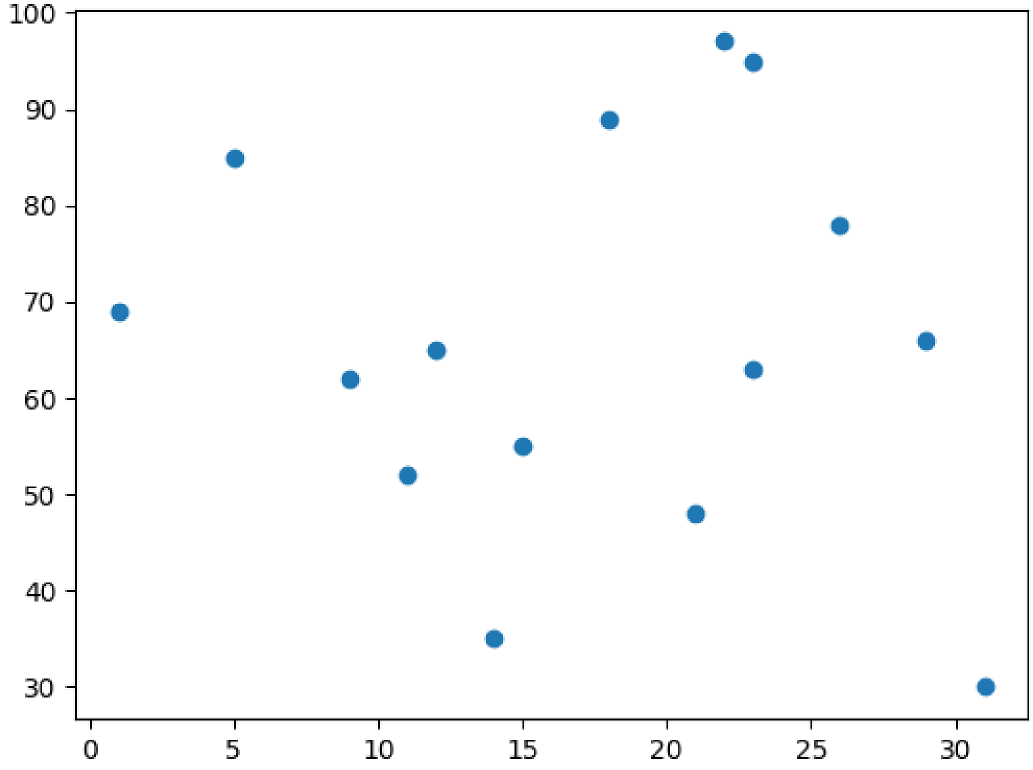
[26]:

```
plt.plot(data['x'],yPred,'r')
plt.scatter(data['x'],data['y'])
plt.show()
```



[27]:

```python
plt.plot(data['x'],data['y'],'o')
plt.show()
```



[ ]:

```python

```