

# least\_square\_implementation

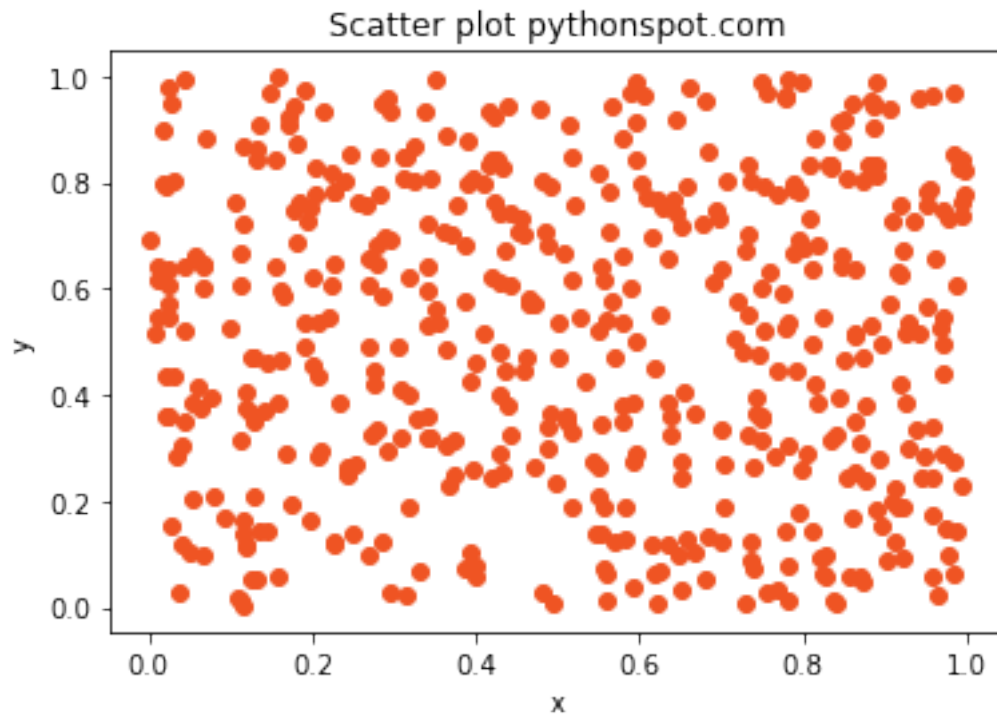
April 26, 2020

```
In [3]: import matplotlib.pyplot as plt
import numpy as np
```

```
In [ ]:
```

```
In [3]: # Create data
N = 500
x1 = np.random.rand(N)
y1 = np.random.rand(N)
colors = (0,0,0)
area = np.pi*3

# Plot
plt.scatter(x1, y1 ,c='#ef5423')
plt.title('Scatter plot pythonspot.com')
plt.xlabel('x')
plt.ylabel('y')
plt.show()
```



In [ ]:

```
In [5]: # Mean X and Y
mean_x = np.mean(x1)
mean_y = np.mean(y1)

# Total number of values
n = len(x1)
```

In [ ]:

```
In [6]: # Using the formula to calculate 'm' and 'c'
numer = 0
denom = 0
for i in range(n):
    numer += (x1[i] - mean_x) * (y1[i] - mean_y)
    denom += (x1[i] - mean_x) ** 2
    m = numer / denom
    c = mean_y - (m * mean_x)

# Printing coefficients
print("Coefficients")
print(m, c)
```

Coefficients

0.006420190628503589 0.5199233693989574

In [ ]:

In [8]:

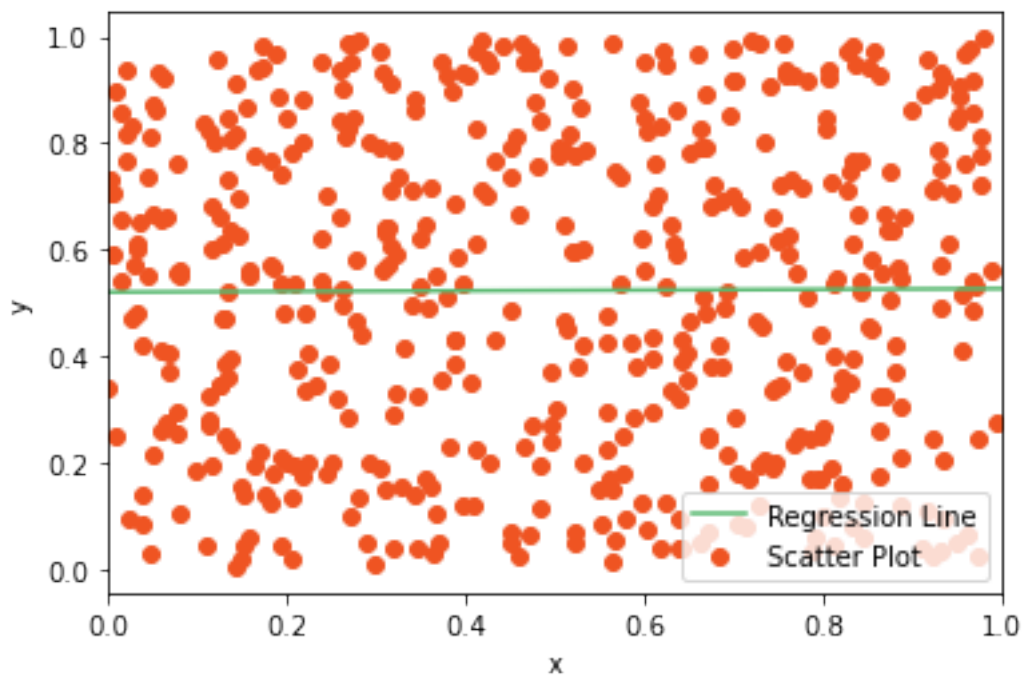
```
max_x = np.max(x1) + 1
min_x = np.min(x1) - 0

# Calculating line values x and y
X1 = np.linspace(min_x, max_x, 1000)
Y1 = c + m * X1

# Plotting Line
plt.plot(X1, Y1, color='#58b970', label='Regression Line')
# Plotting Scatter Points
plt.scatter(x1, y1 ,c='#ef5422',label='Scatter Plot')

axes = plt.gca()
axes.set_xlim([0,1])

plt.xlabel('x')
plt.ylabel('y')
plt.legend()
plt.show()
```



```
In [ ]:
```