Correlation and Co-Variance :
========================

```
import numpy as np

x = np.array([5,3])
y = np.array([6,2])

print("Mean(x)=",x.mean(), "Mean(y)=",y.mean())
print("Variance(x)=",x.var(), "Variance(y)=",y.var())
print("SD(x)=",x.std(), "SD(y)=",y.std())
```

Here is the output of the code :

Mean(x)= 4.0 Mean(y)= 4.0
Variance(x)= 1.0 Variance(y)= 4.0
SD(x)= 1.0 SD(y)= 2.0

Each of these lists has the same mean, namely 4.0. However, they have different standard of deviation. As the standard of deviation is larger, then the data are more spread out. In this case, the second list data is more spread out than first one.

The co-variance determine the direction of the linear relationship between two variables. So the direction could be positive, negative and zero.

$$\text{Co-variance(x,y)} = \frac{1}{n-1}\sum_{i=1}^{n}((x_i - \bar{x})(y_i - \bar{y}))$$

In the below example, there are 6 data points and co-variance is computed in tabular form. The result is 130.2 and indicating positive linear relationship.

Sno	x	y	$x_i - \bar{x}$	$y_i - \bar{y}$	$(x_i - \bar{x})(y_i - \bar{y})$
1	192	218	-6	-7	42
2	218	251	20	26	520
3	197	221	-1	-4	4
4	192	219	-6	-6	36
5	198	223	0	-2	0
6	191	218	-7	-7	49
Sum	1188	1350			651
Mean	198	225			

$\text{Co-variance(x,y)} = \frac{1}{5}\ X\ 651 = 130.2$

3. Comparing Co-variances

Taking the same above example and adding one more feature to it. Lets evaluate the co-variance of (x,y) and (x,z).

Sno	x	y	z	$x_i - \bar{x}$	$y_i - \bar{y}$	$z_i - \bar{z}$	$(x_i - \bar{x})(y_i - \bar{y})$	$(x_i - \bar{x})(z_i - \bar{z})$
1	192	218	6200	-6	-7	-7464	42	-7464
2	218	251	5777	20	26	16420	520	16420
3	197	221	4888	-1	-4	68	4	68
4	192	219	4983	-6	-6	-162	36	-162
5	198	223	5888	0	-2	0	0	0
6	191	218	2000	-7	-7	20692	49	20692
Sum	1188	1350	29736				651	29554
Mean	198	225	4956					

$$\text{Co-variance}(x,z) = \frac{1}{5}\ X\ 29554 = 5910.8$$

The co-variance (x,z) = 5910.8 and which is a very large value than co-variance(x,y) = 130.2.

Does it mean that, the two attributes x and z have better linear relationship than the x and y ?

To answer this question, Let me list out few points —

- The co-variance is a product of 2 units and so, its unit become product of units. The co-variance of (x,y) and (x,z) have different units. So it doesn't make sense to compare the two. Its like comparing two distances whose values are in miles and kms. Of course they need a conversion before a comparison.
- How can we bring the product of 2 units on to a same scale ?. We can make them unit less very easily by dividing them by same product of 2 units. This can be achieved by dividing the co-variance by standard of deviation. For example, the co-variance(x,y) is divided by sd(x) and sd(y) as below. The sd(x) and (xi-x_bar) has same unit. The sd(y) and (yi-y_bar) has same unit.

$$\frac{\frac{1}{n-1}\sum_{i=1}^{n}((x_i - \bar{x})(y_i - \bar{y}))}{\sigma_x \sigma_y}$$

Where σ_x is a standard deviation of x
Where σ_y is a standard deviation of y

Lets go back to our example and calculate standard deviation of x , y , and z.

$$\sigma_x = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}} = \sqrt{\frac{522}{5}} = 10.22$$

$$\sigma_y = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \bar{y})^2}{n-1}} = \sqrt{\frac{830}{5}} = 12.88$$

$$\sigma_z = \sqrt{\frac{\sum_{i=1}^{n}(z_i - \bar{z})^2}{n-1}} = \sqrt{\frac{11833490}{5}} = 1538.41$$

Now divide the co-variances of (x,y) and (x,z) by standard deviations as below.

$$\frac{Cov(x,y)}{\sigma_x \sigma_y} = \frac{130.2}{10.22 \, * \, 12.88} = 0.98$$

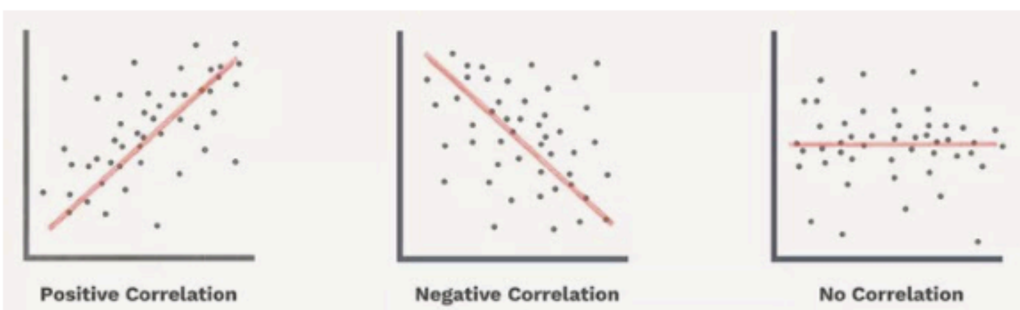
$$\frac{Cov(x,z)}{\sigma_x \sigma_z} = \frac{5910.8}{10.22 \, * \, 1538.41} = 0.36$$

The correlation is the standardized form of co-variance by dividing the co-variance with standard of deviation of each variable. In the previous step, we divided the co-variance(x,y) by sd(x) and sd(y) to get the **correlation coefficient**.

$$\rho_{xy} = \frac{\frac{1}{n-1}\sum_{i=1}^{n}((x_i - \bar{x})(y_i - \bar{y}))}{\sigma_x \sigma_y}$$

As said, the correlation coefficient is unit-less and its range is between -1 and $+1$. It is used to find how strong a relationship is between the attributes. The formulas return a value between -1 and 1, where:

- $+1$ indicates a strong positive relationship.
- -1 indicates a strong negative relationship.
- 0 indicates no relationship at all.

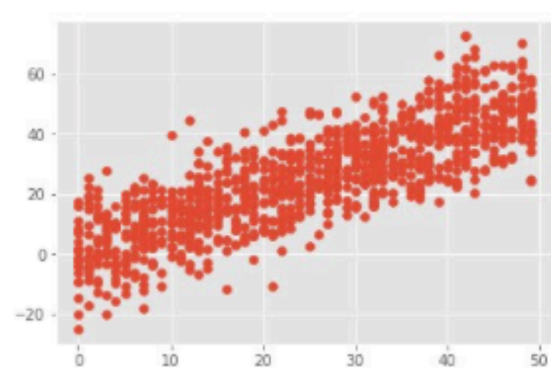


==

```
import numpy as np
import matplotlib
import matplotlib.pyplot as plt
matplotlib.style.use('ggplot')


np.random.seed(1)


x = np.random.randint(0,50,1000)
y = x + np.random.normal(0,10,1000)


print(np.corrcoef(x,y))


plt.scatter(x, y)
plt.show()
```

The output shows that the linear relationship between x and y is stronger and positive direction.

```
[[1.         0.81543901]
 [0.81543901 1.        ]]
```

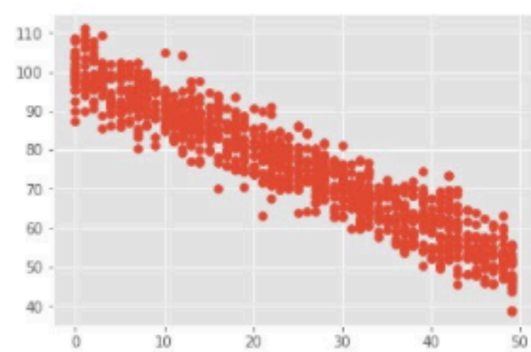


==
======================

```
import numpy as np
import matplotlib
import matplotlib.pyplot as plt
matplotlib.style.use('ggplot')

np.random.seed(1)

x = np.random.randint(0, 50, 1000)
y = 100 - x + np.random.normal(0, 5, 1000)

print(np.corrcoef(x,y))

plt.scatter(x, y)
plt.show()
```

The output shows that the linear relationship between x and y is stronger and negative direction.

```
[[ 1.         -0.94363236]
 [-0.94363236  1.        ]]
```

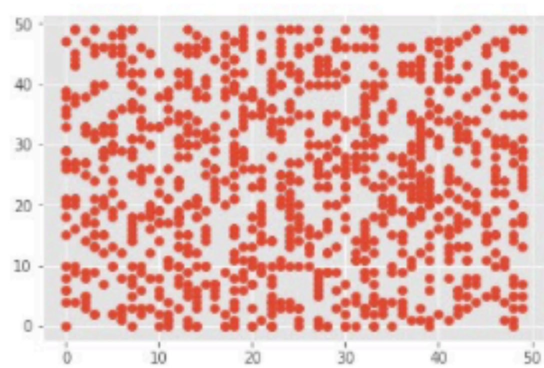


===

```
import numpy as np
import matplotlib
import matplotlib.pyplot as plt
matplotlib.style.use('ggplot')


np.random.seed(1)


x = np.random.randint(0, 50, 1000)
y = np.random.randint(0, 50, 1000)


np.corrcoef(x, y)


print(np.corrcoef(x,y))


plt.scatter(x, y)
plt.show()
```

The output shows that there is NO linear relationship between x and y.

```
[[1.         0.00404702]
 [0.00404702 1.        ]]
```



===
=======================

Correlation matrix

A correlation matrix is a table showing correlation coefficients between variables. A correlation matrix is used to summarize data, as an input into a more advanced analysis, and as a diagnostic for advanced analyses.

Lets compute the correlation matrix for the above data set having 3 attributes as x, y and z.

```
import pandas as pd
dataframe = pd.DataFrame({'X' : [192, 218, 197, 192, 198, 191],
                          'Y' : [218, 251, 221, 219, 223, 218],
                          'Z' : [6200, 5777, 4888, 4983, 5888,
2000]})
print(dataframe.corr())
```

The correlation matrix is printed as below :

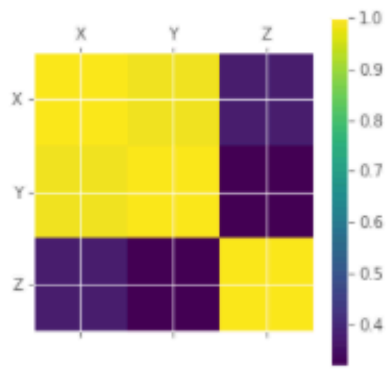
	X	Y	Z
X	1.000000	0.989024	0.376032
Y	0.989024	1.000000	0.318612
Z	0.376032	0.318612	1.000000

===

We can also plot the correlation matrix as below :

```
import pandas as pd
dataframe = pd.DataFrame({'X' : [192, 218, 197, 192, 198, 191],
                          'Y' : [218, 251, 221, 219, 223, 218],
                          'Z' : [6200, 5777, 4888, 4983, 5888,
2000]})
plt.matshow(dataframe.corr())
plt.xticks(range(len(dataframe.columns)), dataframe.columns)
plt.yticks(range(len(dataframe.columns)), dataframe.columns)
plt.colorbar()
plt.show()
```

Here is the output —



==
========================