DataRitz Technologies
Enhancing Technology Experience

Data Analytics and Machine Learning using Python

Session - 26 June 2020

Multiple Linear Regression & Correlation with Pearson's Correlation Coefficient

Multi Linear Regression

- It is the most common form of Linear Regression Analysis.
- It is used to explain the relationship between one continuous dependent variable and two or more independent variable.
- The independent variable can be continuous or contiguous

You can use Multiple Regression where you want to know

- How strong the relationship between two or more independent variable and one dependent variable
- The value of dependent variable at a certain value of the independent varaibles

Multiple Linear Regression Formula

- *The formula for Multiple Linear Regression is*

$$y = \beta_0 + \beta_1 X_1 + ... + \beta_n X_n + \varepsilon$$

where

- y = predicted value of the dependent variable
- B0 = the y-intercept or constant
- B1 = the regression coefficient of the independent variable
- X = Independent variable
- E = model error

Correlations in Python

- Correlations value ranges between -1 and 1

There are two key components of a Correlation value

- *magnitude - The larger the magnitude (closer to 1 or -1), the stronger the correlation*
- *sign – If negative, there is an inverse correlation. If positive, there is a regular correlation*

Positive Correlation

Let's take a look at a positive correlation. Numpy implements a *corrcoef()* function that returns a matrix of correlations of x with x, x with y, y with x and y with y. We're interested in the values of correlation of x with y (so position (1, 0) or (0, 1))

In [1]:

```
import numpy as np
import matplotlib
import matplotlib.pyplot as plt
%matplotlib inline
import pandas as pd
```

In [2]:

```
np.random.seed(1)
```

In [3]:

```
# 100 random integers between 0 and 50
x = np.random.randint(0,50,100)
```

In [4]:

```
# Positive Correlation with some noise
y = x + np.random.normal(0, 5, 100)
```

In [5]:

```
np.corrcoef(x, y)
```

Out[5]:

```
array([[1.        , 0.93775767],
       [0.93775767, 1.        ]])
```

- This correlation is 0.931, a strong positive correlation

In [6]:

```
matplotlib.style.use('ggplot')
plt.scatter(x,y)
plt.show()
```

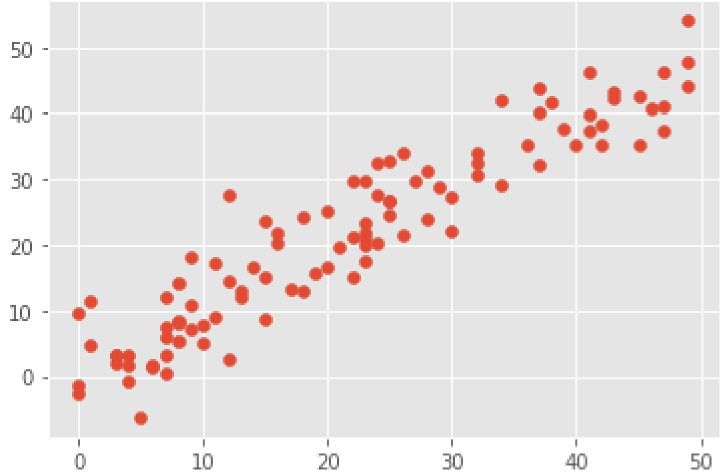


Table Example of Positive Correlation

In [7]:

```
# Read Data from CSV file
positive=pd.read_csv('Positive.csv')
```

In [8]:

```
df=pd.DataFrame(positive)
```

In [9]:

```
TotalExp=df['Years Exp']
```

In [10]:

```
Sal=df['Annual Salary']
```

In [11]:

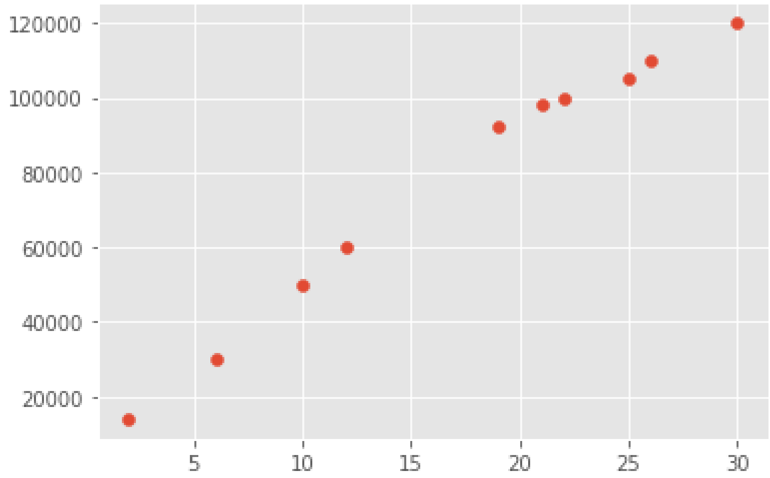
```
np.corrcoef(TotalExp,Sal)
```

Out[11]:

```
array([[1.        , 0.99129685],
       [0.99129685, 1.        ]])
```

In [12]:

```
# Plot the Data
matplotlib.style.use('ggplot')
plt.scatter(TotalExp, Sal)
plt.show()
```



Negative Correlation

In [13]:

```
# 100 random integers between 0 and 50
x = np.random.randint(0, 50, 100)
```

In [14]:

```
# Negative Correlation with some noise
y = 100 - x + np.random.normal(0, 5, 100)
```

In [15]:

```
np.corrcoef(x, y)
```

Out[15]:

```
array([[ 1.        , -0.93646148],
       [-0.93646148,  1.        ]])
```

- Our correlation is now negative and close to 1

In [16]:

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

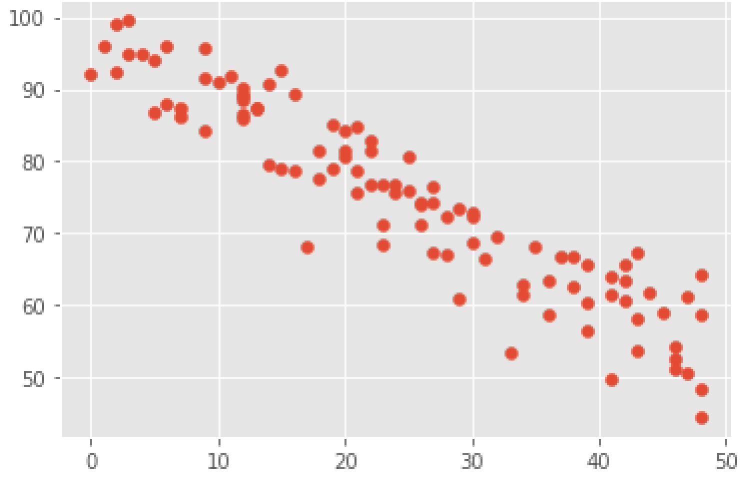


Table Example of Negative Corrleation

In [21]:

```
# load data from CSV file
speed=pd.read_csv('Negative.csv')
```

In [22]:

```
df=pd.DataFrame(speed)
```

In [23]:

```
spe=df['Speed']
```

In [24]:

```
Time=df['Time']
```

In [25]:

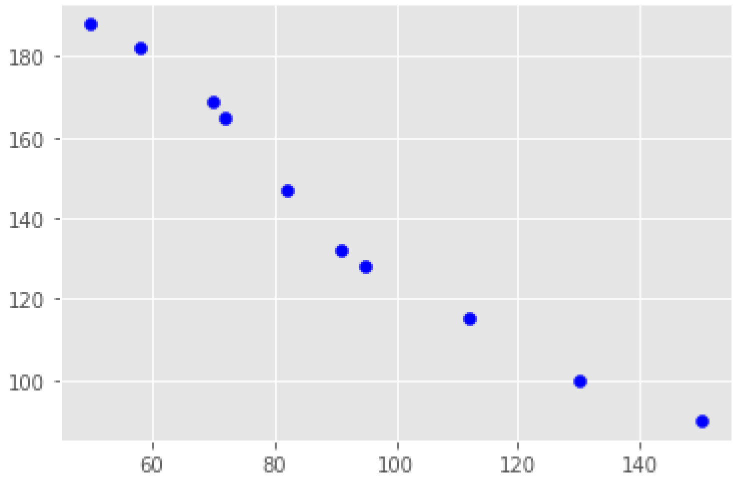
```
np.corrcoef(spe,Time)
```

Out[25]:

```
array([[ 1.        , -0.98026597],
       [-0.98026597,  1.        ]])
```

In [26]:

```
plt.scatter(spe,Time,color='Blue')
plt.show()
```



No/Weak Correlation

What if there is no correlation between x and y ?

In [27]:

```
x = np.random.randint(0, 50, 100)
y = np.random.randint(0, 50, 100)

np.corrcoef(x, y)
```

Out[27]:

```
array([[ 1.        , -0.09762036],
       [-0.09762036,  1.        ]])
```

In [28]:

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

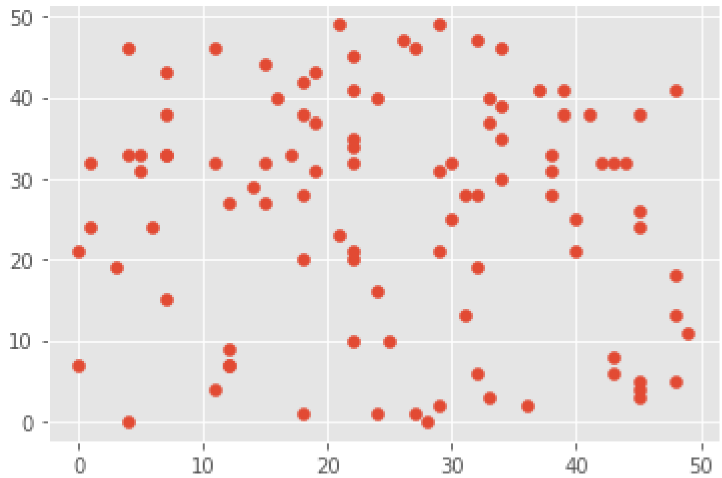


Table Example of Zero Correlation

In [29]:

```
zero=pd.read_csv('zero.csv')
```

In [30]:

```
df=pd.DataFrame(zero)
```

In [31]:

```
temp=df['Temp']
```

In [32]:

```
price=df['Stock Price']
```

In [33]:

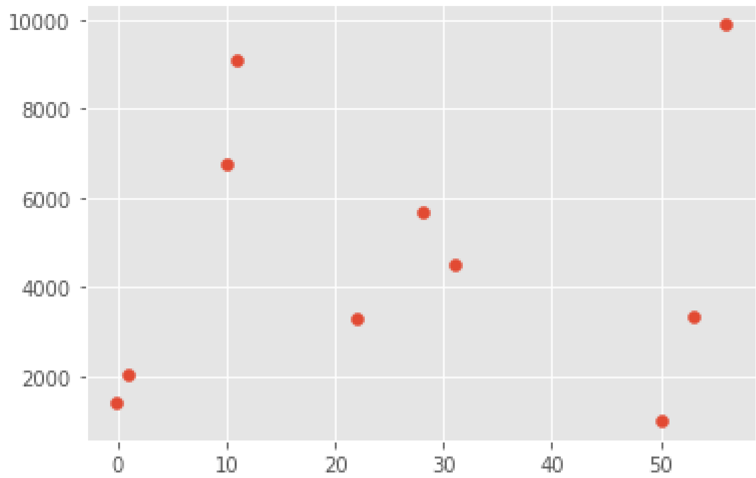
```
np.corrcoef(temp,price)
```

Out[33]:

```
array([[1.        , 0.15478381],
       [0.15478381, 1.        ]])
```

In [34]:

```
plt.scatter(temp,price)
plt.show()
```



Correlation Matrix

In [35]:

```
df=pd.DataFrame({'a': np.random.randint(0, 20, 1000)})
```

In [36]:

```
df['b'] = df['a'] + np.random.normal(0, 10, 1000) # positively correlated with 'a'
```

In [37]:

```
df['c'] = 100 - df['a'] + np.random.normal(0, 5, 1000) # negatively correlated with 'a'
```

In [38]:

```
df['d'] = np.random.randint(0, 20, 1000) # not correlated with 'a'
```

In [39]:

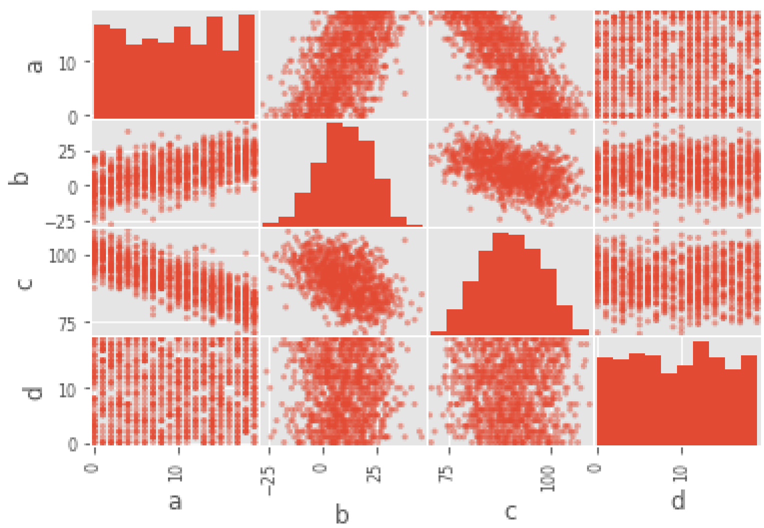
```
df.corr()
```

Out[39]:

	a	b	c	d
a	1.000000	0.530994	-0.763981	0.006481
b	0.530994	1.000000	-0.436875	0.015120
c	-0.763981	-0.436875	1.000000	0.017810
d	0.006481	0.015120	0.017810	1.000000

In [40]:

```
from pandas.plotting import scatter_matrix
scatter_matrix(df)
plt.show()
```



Pearson's Correlation Formula

- The name correlation suggests the relationship between two variables as their Co-relation. The correlation coefficient is the measurement of correlation
- The linear dependency between the data set is done by the Pearson Correlation coefficient. It is also known as Pearson product moment correlation coefficient

- When the correlation coefficient comes down to zero, then the data is said to be not related. While, if we are getting the value of +1, then the data are positively correlated and -1 has a negative correlation

- The Pearson correlation coefficient is denoted by the letter “r”
- *The formula for Pearson correlation coefficient r is given by :*

$$r = \frac{n(\sum xy) - (\sum x)(\sum y)}{\sqrt{[n\sum x^2 - (\sum x)^2][n\sum y^2 - (\sum y)^2]}}$$

Where,

- r = Pearson correlation coefficient
- x = Values in the first set of data
- y = Values in the second set of data
- n = Total number of values.

Example

S.No	Age (x)	Income (y)	(xy)	(x^2)	(y^2)
1	10	1500	15000	100	2250000
2	20	2000	40000	400	4000000
3	30	2500	75000	900	6250000
4	40	3000	120000	1600	9000000
5	50	3500	175000	2500	12250000

S.No	Age (x)	Income (y)	(xy)	(x^2)	(y^2)
1	10	1500	15000	100	2250000
2	20	2000	40000	400	4000000
3	30	2500	75000	900	6250000
4	40	3000	120000	1600	9000000
5	50	3500	175000	2500	12250000
Total	150	12500	425000	5500	33750000

- On solving this, r comes out to be '1', which means the data has a "Positive Correlation"

Interpretation of Pearson's Correlation Coefficient

- The sign of the correlation coefficient determines whether the correlation is positive of negative.
- The magnitude determined the strength of correlation.

Coefficient of Determination

- The coefficient of determination is the square of the correlation coefficient.
- The statistics quantifies the proportion of the variance of one variable by the other

In [ ]:

```

```