How to Add a Fitted Line to a Scatterplot in Python

In the following tutorial, we will discuss how to add a fitted line to a scatterplot in Python. But first of all, what is a scatterplot and when do we use it? To answer this question we will run through our theory first.

Definition

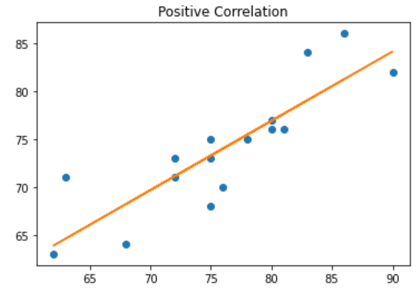
A scatterplot (or scatter plot; scatter graph; scatter chart; scattergram; scatter diagram) is a type of plot to display the relationship between two quantitative variables for a set of data using Cartesian coordinates.

Each data point is represented as a circle (could be any symbol like a cross or an x, but the default out of statistical programs is the circle). Scatterplots are important in statistics because they can show the extent of **correlation**, if any, between the values of observed quantities or phenomena (called variables). If no correlation exists between the variables, the points appear randomly scattered on the coordinate plane. If a large correlation exists, the points concentrate near a straight line. Scatter plots are useful data visualization tools for illustrating a **trend**.

Besides showing the extent of correlation, a scatter plot shows the sense of the correlation:

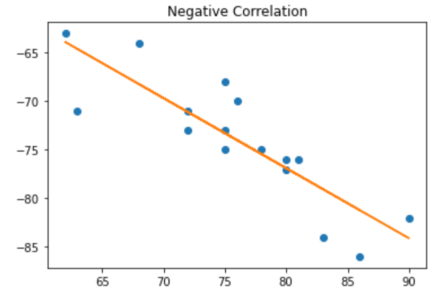
- If the vertical (or y-axis) variable increases as the horizontal (or x-axis) variable increases, the correlation is positive.

Visual Example



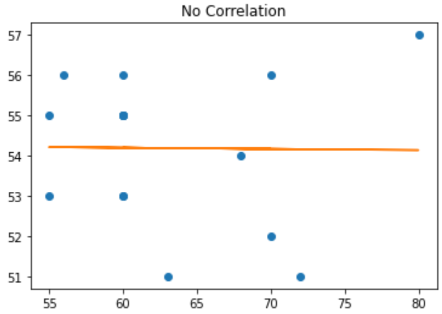
- If the y-axis variable decreases as the x-axis variable increases or vice-versa, the correlation is negative.

Visual Example



- If it is impossible to establish either of the above criteria, then the correlation is zero, similar effect are shown when the correlation is close to zero whether it is negative or positive.

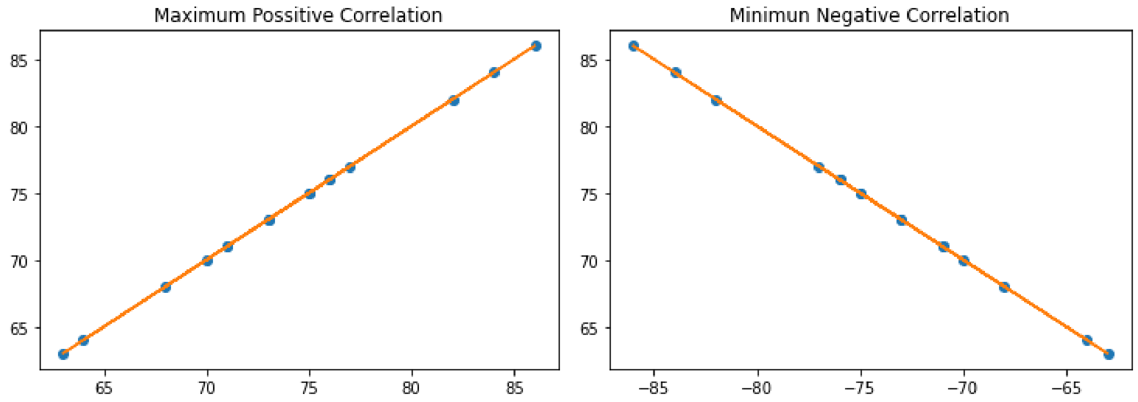
Visual Example



The Range for the correlation is between -1 and+1. The maximum possible positive correlation is +1 (or +100%), when all the points in a scatter plot lie exactly along a straight line with a positive slope.

The minimum possible negative correlation is -1 (or -100%), in which case all the points lie exactly along a straight line with a negative slope.

Visual Example



Correlation is often confused with causation, in the pure sense, while a scatterplot can reveal the nature and extent of correlation, it implies nothing about causation.

Example: Add a Fitted Line to a Scatterplot in Python

The following python syntax shows how to draw a basic scatterplot in Python. The x-variable is Height (in cm) and the y-variable is Weight (in Kg).

```
import numpy as np

#create data
x = np.array([183, 162, 172, 181, 180, 168, 176, 180, 190, 175,
178, 175, 186, 172, 175, 163])
y = np.array([84, 63, 71, 76, 77, 64, 70, 76, 82, 68, 75, 73, 86,
73, 75, 71])
```

Method 1: Matplotlib

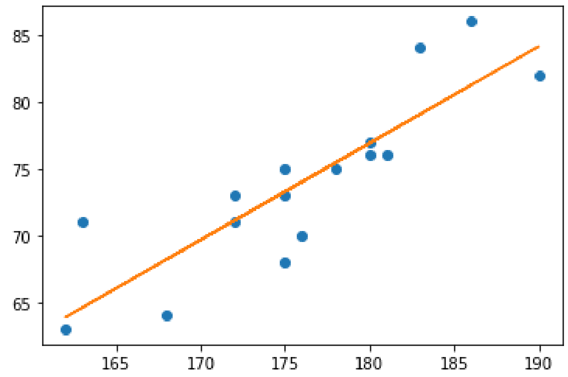
```
import numpy as np
import matplotlib.pyplot as plt

#create basic scatterplot
plt.plot(x, y, 'o')

#obtain m (Slope) and b(Constant) of the linear line
m, b = np.polyfit(x, y, deg=1) #deg=1 means: linear line (i.e.
polynomial of degree 1)

#add line to scatterplot
```

```
plt.plot(x, m*x+b)
plt.show()
```

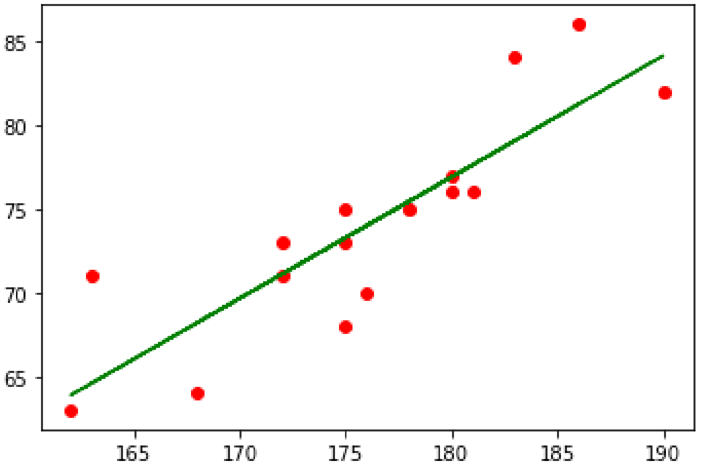


You can change the color of the graph and the line color as well by adding the following which is in red color:

```
#use green as color for individual points
plt.plot(x, y, 'o', color='red')

#obtain m (slope) and b(Constant) of the linear line
m, b = np.polyfit(x, y, deg=1)

#use red as color for regression line
plt.plot(x, m*x+b, color='green')
plt.show()
```



**Note

We use plt.show() in order not to get this message under our scatterplot:

```
[<matplotlib.lines.Line2D at 0x137710396a0>]
```

If you want to know the fitting line's equation (as well as known as Linear Regression Line), a little reminder:

- m : the slope of the line
- b : the constant value of the line

```
print("Regression Line: y = ",np.round(m,4),"x ",np.round(b,4))
```

The **`np.round(b,4)`** will round the number to 4-digits. The output is:

```
Regression Line: y = 0.723 x - 53.246
```

As you have noticed, m = 0.723 which is positive, this determines the slope of the line and in our example, we can see that the scatterplot displays that there is a positive correlation between Height and Weight. We can measure it with Pearson correlation coefficient formula. In python you can with the help of **`corrcoef`** function, write the following code:

```
rho = np.corrcoef(x,y)
rho = np.round(rho.flat[1],4)

print("Pearson's rho =",rho)
```

The output would therefore be:

```
Pearson's rho = 0.8564
```

If you want to create the percentage of Pearson's rho then you add the following:

```
print("Pearson's rho =",rho ,"or",100*rho,"%")
```

And then the output would be like:

```
Pearson's rho = 0.8564 or 85.64 %
```

In conclusion, we know that the Pearson's coefficient is 0.8564 or 85.64 %, which is positive and close to +1, which means that there is a positive correlation between Height and Weight in our sample data. By definition, the values that the Pearson's coefficient can take lies between -1 and +1

Method 2: Seaborn

We will use the same data as before with x-variable being the Height (in cm) and the y-variable to be the Weight (in Kg), but we will use a different graphic library in Python to help us create a scatterplot with a fitting line.

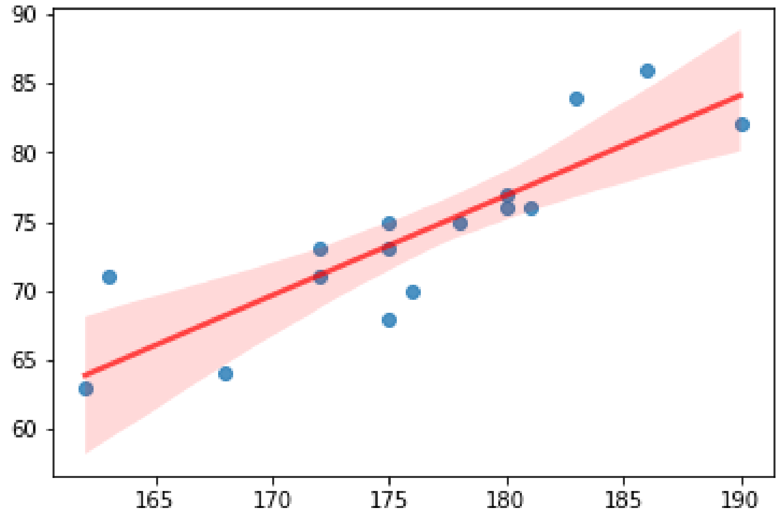
First of all we have to import the graphic library and we are going to need the **seaborn** graphic library:

```
import seaborn as sns

# The plot
sns.regplot(x=x, y=y, line_kws={"color":"r","alpha":0.7,"lw":2.5})
plt.show()
```

- color: Color of the line
- alpha: Opacity value of the line, the values lies between 0 and 1, with 0 meaning that having no line at all and with 1 having the full line appeared with no transparency.
- lw: Length width of the line

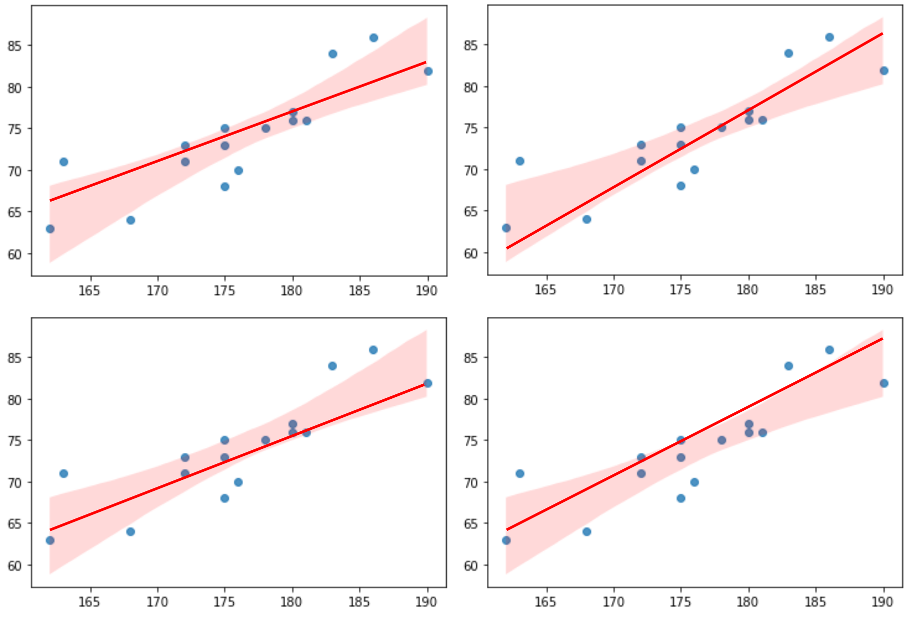
Output



The more transparent red area represents the 95% Confidence Interval for the line we draw.

The interpretation is, that on average, if we conduct the same experiment 100 times, we expect the fitting line to be around the red area 95 times or else 95% of the time.

Some possible outcomes for the fitting line that are into the 95% Confidence Interval:


85
80
75
70
65
60
165
170
175
180
185
190