SIT384 Cyber security analytics

Pass Task 5.1P: Data correlation

Task description:

The Pearson’s correlation coefficient is a measure of the strength of the linear relationship between two variables.

Correlation values range 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.

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)). The values of correlation of x with y might be Positive Correlation, Negative Correlation, or No/Weak Correlation.

You are given 1 dataframe which has 4 series:

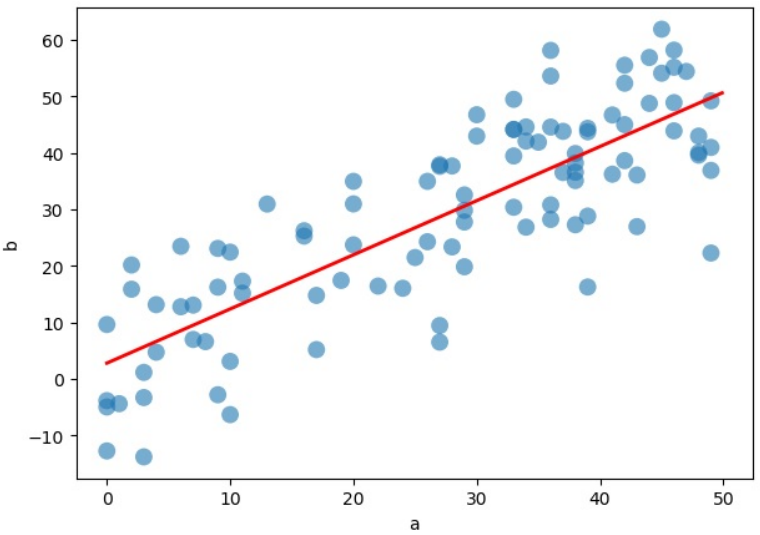
```
df = pd.DataFrame({'a': np.random.randint(0, 50, size=100)})
df['b'] = df['a'] + np.random.normal(0, 10, size=100)
df['c'] = 100 - 2* df['a'] + np.random.normal(0, 10, size=100)
df['d'] = np.random.randint(0, 50, 100)
```

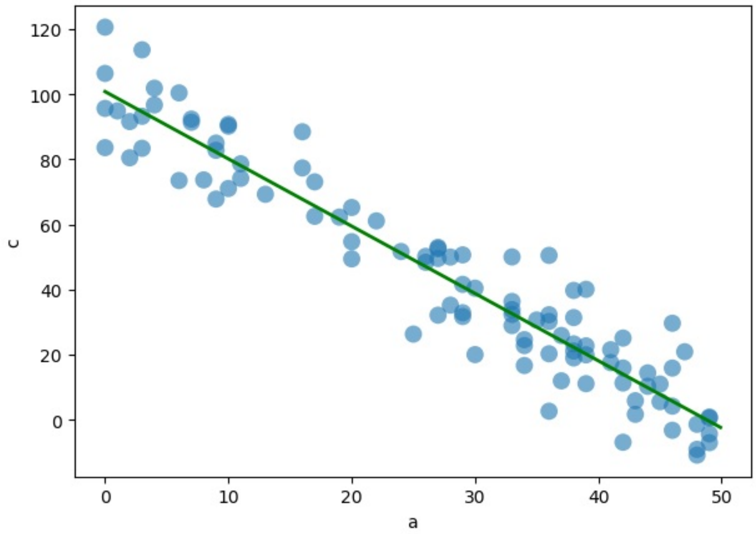
You are asked to :

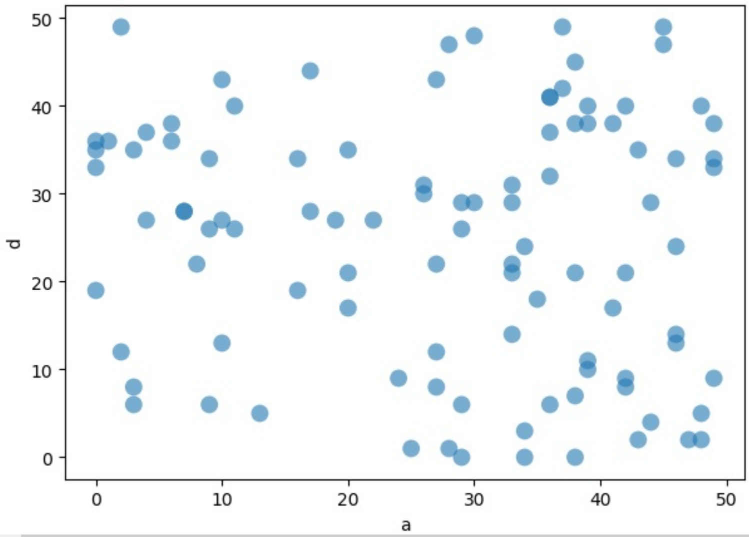
- calculate the Pearson's-r coefficient and corrcoef() for
  - df[‘a’] and df[‘b’],
  - df[‘a’] and df[‘c’], and
  - df[‘a’] and df[‘d’].
- visualize data with positive correlation and data with negative correlation using scatter plot and np.ployfit() where possible.
  - X axis is a
  - Y axis is b, c or d
  - plt.subplots(figsize=(7, 5), dpi=100)

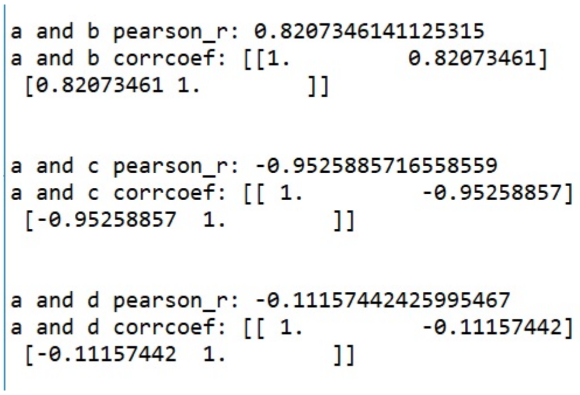
  - line_coef = np.polyfit(x, y, 1)
  - xx = np.arange(0, 50, 0.1)
  - yy = line_coef[0]*xx + line_coef[1]
  - plot(xx, yy, color, lw=2)

Sample output as shown in the following figure is for demonstration purposes only.


b
60
50
40
30
20
10
0
−10
0
10
20
30
40
50
a


c
120
100
80
60
40
20
0
0
10
20
30
40
50
a


d
50
40
30
20
10
0
0
10
20
30
40
50
a



```
a and b pearson_r: 0.8207346141125315
a and b corrcoef: [[1.         0.82073461]
 [0.82073461 1.        ]]


a and c pearson_r: -0.9525885716558559
a and c corrcoef: [[ 1.         -0.95258857]
 [-0.95258857  1.        ]]


a and d pearson_r: -0.11157442425995467
a and d corrcoef: [[ 1.         -0.11157442]
 [-0.11157442  1.        ]]
```

Submission:

Submit the following files to OnTrack:

1. Your program source code (e.g. task5_1.py)
2. A screen shot of your program running

Check the following things before submitting:

1. Add proper comments to your code