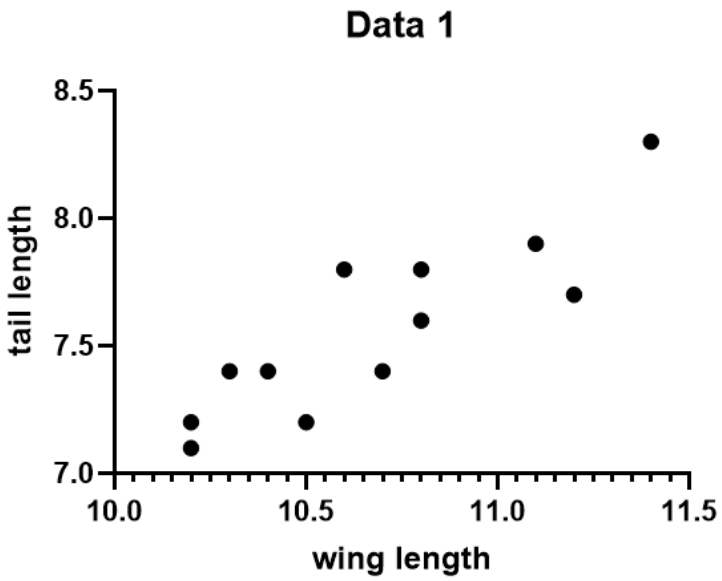
1. Plot X vs Y. Do they look related?

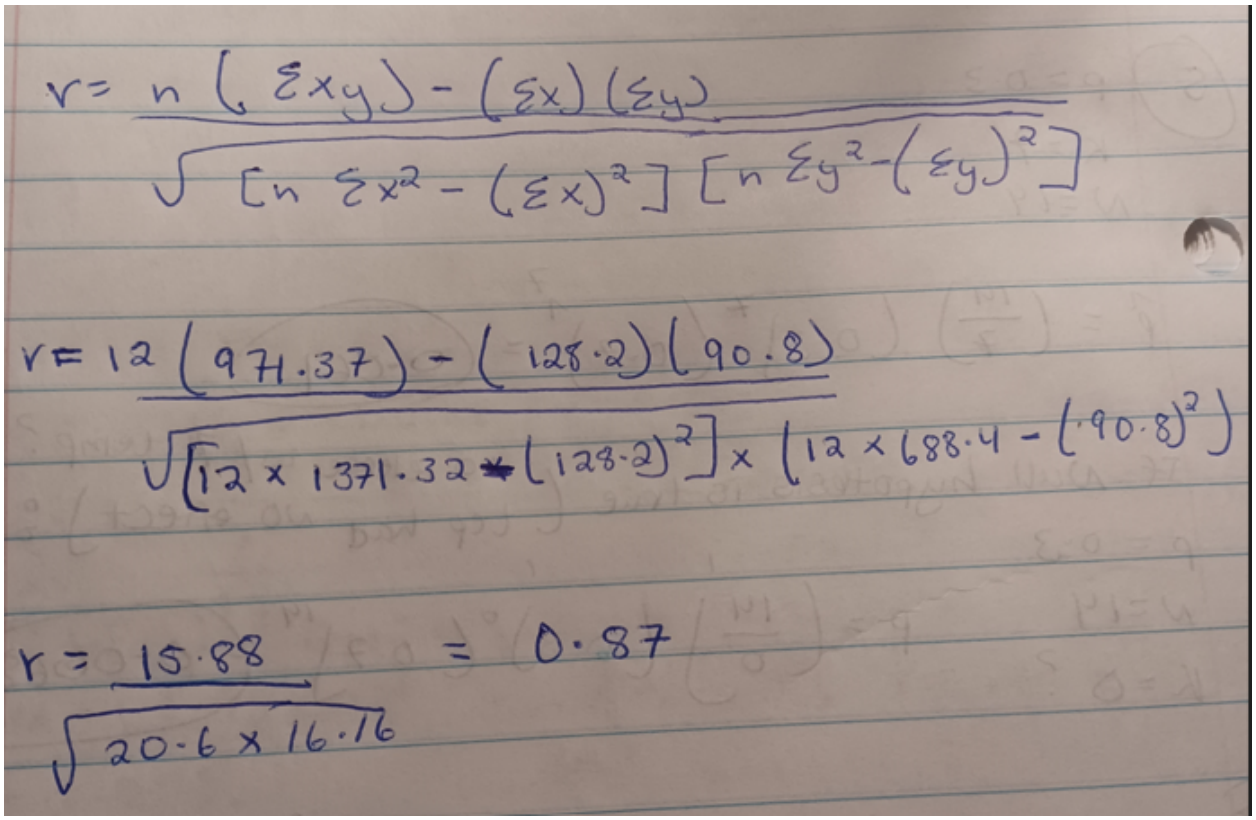


Answer: yes

2. Calculate $r_{X,y}$ and $r_{Y,X}$, first using the equations above and then using either the Python numpy function corrcoef or Matlab's built-in corrcoef. Did you get the same answers?

Answer: yes

x	y	x*y	x^2	y^2
10.4	7.4	76.96	108.16	54.76
10.8	7.6	82.08	116.64	57.76
11.1	7.9	87.69	123.21	62.41
10.2	7.2	73.44	104.04	51.84
10.3	7.4	76.22	106.09	54.76
10.2	7.1	72.42	104.04	50.41
10.7	7.4	79.18	114.49	54.76
10.5	7.2	75.6	110.25	51.84
10.8	7.8	84.24	116.64	60.84
11.2	7.7	86.24	125.44	59.29
10.6	7.8	82.68	112.36	60.84
11.4	8.3	94.62	129.96	68.89
128.2	90.8	971.37	1371.32	688.4



In [1]:
```
import numpy as np
```

In [2]:
```
x_simple = np.array([10.4, 10.8, 11.1, 10.2, 10.3, 10.2, 10.7, 10.5, 10.8, 11.2, 10.6, 11.4])
y_simple = np.array([7.4, 7.6, 7.9, 7.2, 7.4, 7.1, 7.4, 7.2, 7.8, 7.7, 7.8, 8.3])
my_rho = np.corrcoef(x_simple, y_simple)
print(my_rho)
```

```
[[1.         0.87035456]
 [0.87035456 1.        ]]
```

3. What is the standard error of $r_{X,Y}$? The 95% confidence intervals computed from the standard error?

$$s_r = \sqrt{\frac{1-r^2}{n-2}}$$

sr= 0.1559

4. Should the value of $r_{X,Y}$ be considered significant at the *p*<0.05 level, given a two-tailed test (i.e., we reject if the test statistic is too large on either tail of the null distribution) for $H0: r_{X,Y}=0$?

Answer: yes

5. Yale does the exact same study and finds that his correlation value is 0.75. Is this the same as yours? That is, evaluate $H0: r=0.75$.

Answer: no, mine was 0.87

6. Finally, calculate the statistical power and sample size needed to reject $H0:r=0$ when $r \geq 0.5$

Answer: to reject the null hypothesis, an effect size of 0.87 would need about 21 as a sample size to get 80% power, which is really good.

In [3]:
```
from math import sqrt
from statsmodels.stats.power import TTestIndPower
```

In [4]:
```
# factors for power analysis
alpha = 0.05
power = 0.8
effect_size= 0.87

# perform power analysis to find sample size
# for given effect
obj = TTestIndPower()
n = obj.solve_power(effect_size=effect_size, alpha=alpha, power=power,
                    ratio=1, alternative='two-sided')

print('Sample size/Number needed in each group: {:.3f}'.format(n))
```

```
Sample size/Number needed in each group: 21.743
```