

Correlation

- variable within dataset can be related for lots of reasons
- Types
 - Pearson's r
 - spearman,s rho
 - kendall's tau

For Example:

1. One variable could cause or depend in the value of another variable.
2. one variable could be lightly associated with another variable.
3. Two variable could depend on a third unknown variable.

Positive correlation: both variables change in same direction\ **Neutral correlation:** no relationship in the change of variable\ **Negative correlation:** variables change in opposite direction

Covariance

- Variables can be related by a linear relationship. this is a relationship that is constantly additive across the two data sample.
- This relationship can be summerized between two variables, called the covariance.
- The sign of the covariance can be interpreted as whether the two variables change in the same direction (positive) or change in the different direct (negative).
- The magnitude of the covariance is not easily interpreted. A covariance value of zero indicates that both variables are completely independent.

```
In [ ]: # import library
import numpy as np
import seaborn as sns
import pandas as pd
import matplotlib.pyplot as plt
# import dataset
kashti = sns.load_dataset('titanic')
phool = sns.load_dataset('iris')
```

```
In [ ]: kashti.head()
```

```
Out[ ]:
   survived  pclass    sex  age  sibsp  parch    fare  embarked  class  who  adult_male  d
0         0      3   male  22.0     1     0   7.2500         S   Third   man         True  1
1         1      1  female  38.0     1     0  71.2833         C    First  woman        False
2         1      3  female  26.0     0     0   7.9250         S   Third  woman        False  1
3         1      1  female  35.0     1     0  53.1000         S    First  woman        False
4         0      3   male  35.0     0     0   8.0500         S   Third   man         True  1
```

```
In [ ]: np.cov(kashti['age'], kashti['fare']) # numpy fucntion tahts why we use np.cov
# also can find 'google' more libraries of covariance
```

```
Out[ ]: array([[          nan,           nan],
               [          nan, 2469.43684574]])
```

```
In [ ]: # ANTOHER Example
import numpy as np

x= kashti['age']
y= kashti['fare']

# find out covariance with respect to columns
cov_mat = np.stack((x,y), axis = 0)
cov_mat
```

```
Out[ ]: array([[22.      , 38.      , 26.      , ...,          nan, 26.      , 32.      ],
               [ 7.25     , 71.2833,  7.925     , ..., 23.45     , 30.      ,  7.75     ]])
```

```
In [ ]: print(np.cov(cov_mat))
# nan because of null values
```

```
[[          nan          nan]
 [          nan 2469.43684574]]
```

- Corelation istead of COV

```
In [ ]: kashti.columns
```

```
Out[ ]: Index(['survived', 'pclass', 'sex', 'age', 'sibsp', 'parch', 'fare',
              'embarked', 'class', 'who', 'adult_male', 'deck', 'embark_town',
              'alive', 'alone'],
              dtype='object')
```

```
In [ ]: kashti.info()
# corelation always lagta ha against int or float dtypes
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 15 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   survived    891 non-null    int64
1   pclass      891 non-null    int64
2   sex         891 non-null    object
3   age         714 non-null    float64
4   sibsp       891 non-null    int64
5   parch       891 non-null    int64
6   fare        891 non-null    float64
7   embarked    889 non-null    object
8   class       891 non-null    category
9   who         891 non-null    object
10  adult_male   891 non-null    bool
11  deck        203 non-null    category
12  embark_town  889 non-null    object
13  alive       891 non-null    object
14  alone       891 non-null    bool
dtypes: bool(2), category(2), float64(2), int64(4), object(5)
memory usage: 80.7+ KB
```

```
In [ ]: # correlation at kashti dataset
kashti.corr()
# corr function is builtin
# 1 highly +ve, -1 highly -ve, 0 is no correlation
```

```
Out[ ]:
```

	survived	pclass	age	sibsp	parch	fare	adult_male	alone
survived	1.000000	-0.338481	-0.077221	-0.035322	0.081629	0.257307	-0.557080	-0.203367
pclass	-0.338481	1.000000	-0.369226	0.083081	0.018443	-0.549500	0.094035	0.135207
age	-0.077221	-0.369226	1.000000	-0.308247	-0.189119	0.096067	0.280328	0.198270
sibsp	-0.035322	0.083081	-0.308247	1.000000	0.414838	0.159651	-0.253586	-0.584471
parch	0.081629	0.018443	-0.189119	0.414838	1.000000	0.216225	-0.349943	-0.583398
fare	0.257307	-0.549500	0.096067	0.159651	0.216225	1.000000	-0.182024	-0.271832
adult_male	-0.557080	0.094035	0.280328	-0.253586	-0.349943	-0.182024	1.000000	0.404744
alone	-0.203367	0.135207	0.198270	-0.584471	-0.583398	-0.271832	0.404744	1.000000

```
In [ ]: kashti.corr(method='pearson') # for normal data
```

```
Out[ ]:
```

	survived	pclass	age	sibsp	parch	fare	adult_male	alone
survived	1.000000	-0.338481	-0.077221	-0.035322	0.081629	0.257307	-0.557080	-0.203367
pclass	-0.338481	1.000000	-0.369226	0.083081	0.018443	-0.549500	0.094035	0.135207
age	-0.077221	-0.369226	1.000000	-0.308247	-0.189119	0.096067	0.280328	0.198270
sibsp	-0.035322	0.083081	-0.308247	1.000000	0.414838	0.159651	-0.253586	-0.584471
parch	0.081629	0.018443	-0.189119	0.414838	1.000000	0.216225	-0.349943	-0.583398
fare	0.257307	-0.549500	0.096067	0.159651	0.216225	1.000000	-0.182024	-0.271832
adult_male	-0.557080	0.094035	0.280328	-0.253586	-0.349943	-0.182024	1.000000	0.404744
alone	-0.203367	0.135207	0.198270	-0.584471	-0.583398	-0.271832	0.404744	1.000000

```
In [ ]: kashti.corr(method='spearman') # for non gaussian distrubution data
```

```
Out[ ]:
```

	survived	pclass	age	sibsp	parch	fare	adult_male	alone
survived	1.000000	-0.339668	-0.052565	0.088879	0.138266	0.323736	-0.557080	-0.203367
pclass	-0.339668	1.000000	-0.361666	-0.043019	-0.022801	-0.688032	0.099351	0.135896
age	-0.052565	-0.361666	1.000000	-0.182061	-0.254212	0.135051	0.257641	0.167293
sibsp	0.088879	-0.043019	-0.182061	1.000000	0.450014	0.447113	-0.307948	-0.828215
parch	0.138266	-0.022801	-0.254212	0.450014	1.000000	0.410074	-0.397286	-0.683719
fare	0.323736	-0.688032	0.135051	0.447113	0.410074	1.000000	-0.308894	-0.531472
adult_male	-0.557080	0.099351	0.257641	-0.307948	-0.397286	-0.308894	1.000000	0.404744
alone	-0.203367	0.135896	0.167293	-0.828215	-0.683719	-0.531472	0.404744	1.000000

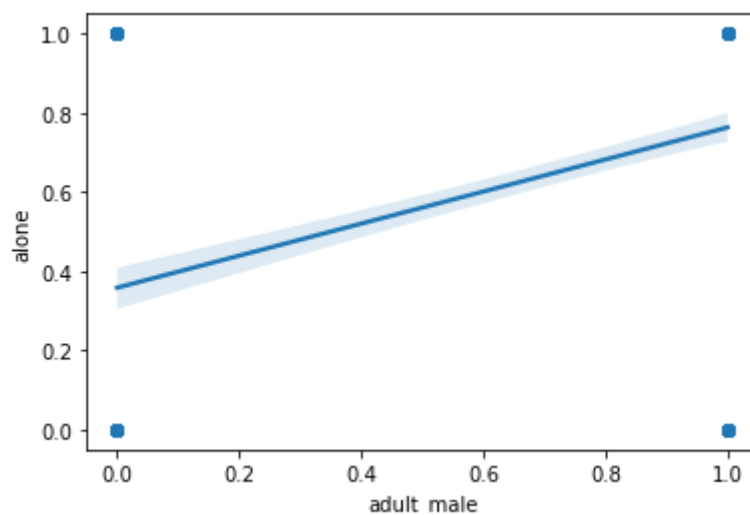
```
In [ ]: corrp = kashti.corr(method='pearson') # store command as corrp
```

```
In [ ]: # +ve corealtion
sns.regplot(kashti['adult_male'], kashti['alone'], data=kashti)
```

C:\Users\Junaid\anaconda3\lib\site-packages\seaborn_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

```
warnings.warn(
```

```
Out[ ]: <AxesSubplot:xlabel='adult_male', ylabel='alone'>
```

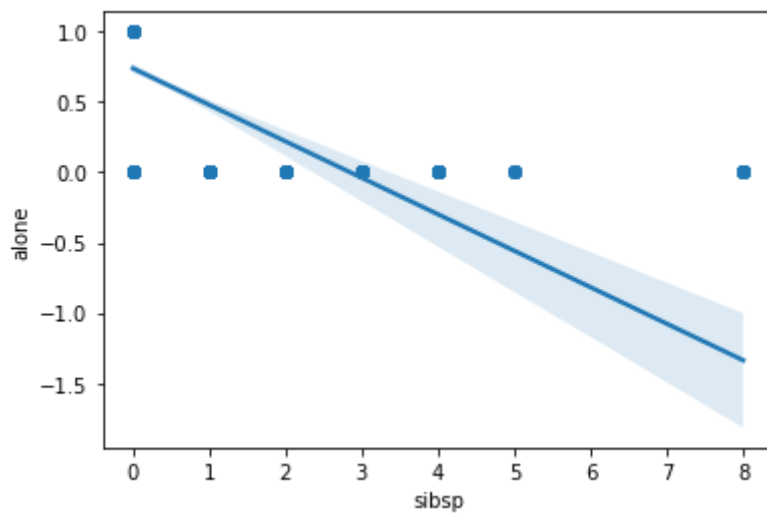


```
In [ ]: # -ve corealtion
sns.regplot(kashti['sibsp'], kashti['alone'], data=kashti)
```

C:\Users\Junaid\anaconda3\lib\site-packages\seaborn_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

```
warnings.warn(
```

```
Out[ ]: <AxesSubplot:xlabel='sibsp', ylabel='alone'>
```



```
In [ ]: # at phool dataset
        phool.head()
```

```
Out [ ]:
```

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

```
In [ ]: phool.corr(method='spearman')
```

```
Out [ ]:
```

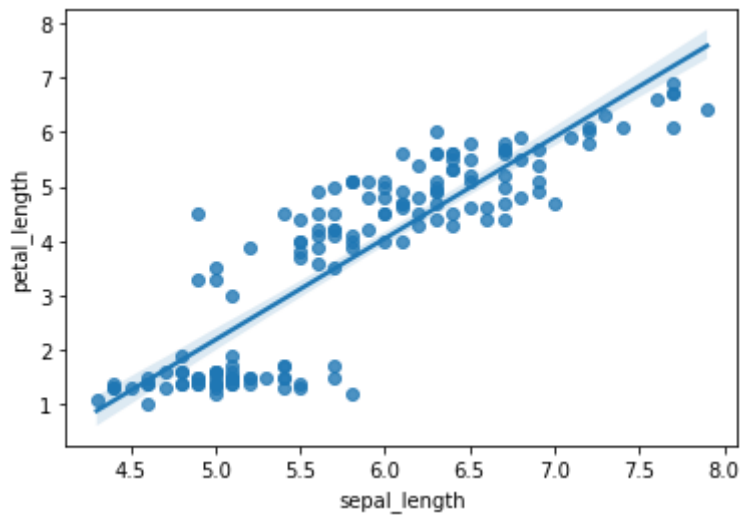
	sepal_length	sepal_width	petal_length	petal_width
sepal_length	1.000000	-0.166778	0.881898	0.834289
sepal_width	-0.166778	1.000000	-0.309635	-0.289032
petal_length	0.881898	-0.309635	1.000000	0.937667
petal_width	0.834289	-0.289032	0.937667	1.000000

```
In [ ]: # +ve corealtion
        sns.regplot(phool['sepal_length'], phool['petal_length'], data=phool)
```

C:\Users\Junaid\anaconda3\lib\site-packages\seaborn_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

warnings.warn(

```
Out [ ]: <AxesSubplot:xlabel='sepal_length', ylabel='petal_length'>
```

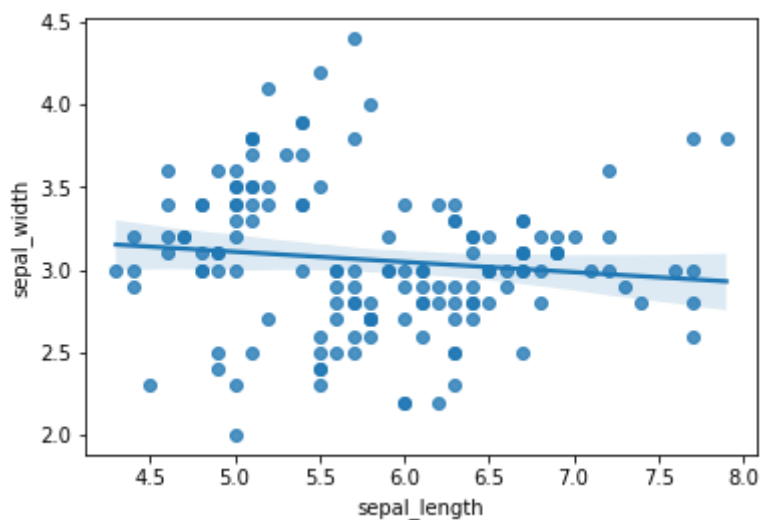


```
In [ ]: sns.regplot(phool['sepal_length'], phool['sepal_width'], data=phool)
# almost 0 correlation
```

C:\Users\Junaid\anaconda3\lib\site-packages\seaborn_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

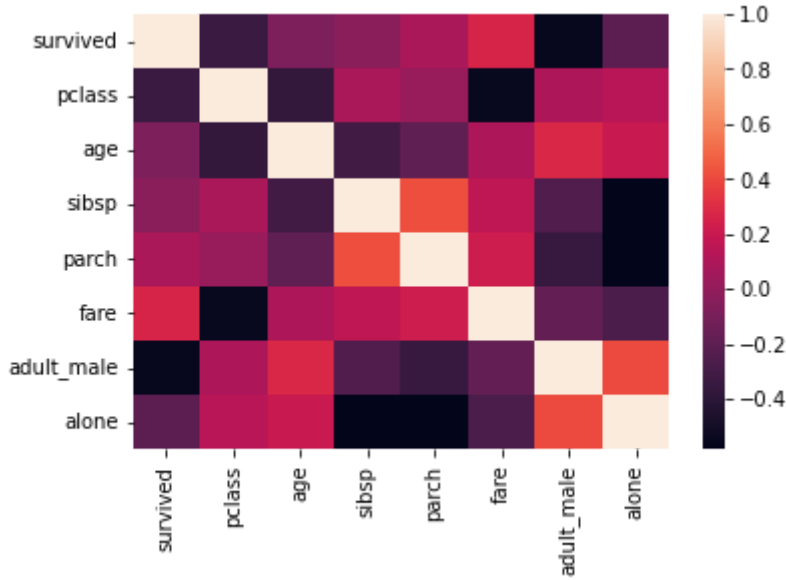
warnings.warn(

```
Out[ ]: <AxesSubplot:xlabel='sepal_length', ylabel='sepal_width'>
```



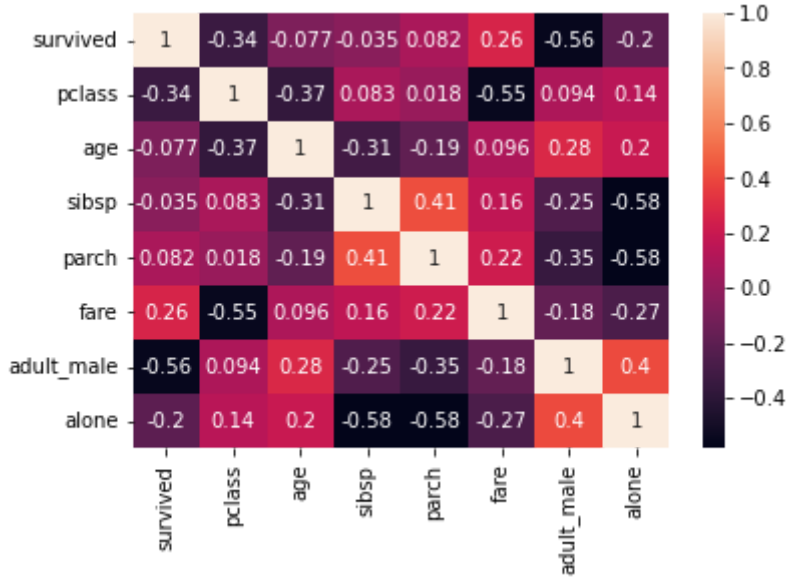
```
In [ ]: # apply corr function
corr = kashti.corr(method='pearson')
sns.heatmap(corr)
```

```
Out[ ]: <AxesSubplot:>
```



```
In [ ]: sns.heatmap(corr, annot=True)

Out[ ]: <AxesSubplot:>
```



```
In [ ]: # ANOTHER STYLE
corr.style.background_gradient(cmap='coolwarm')
```

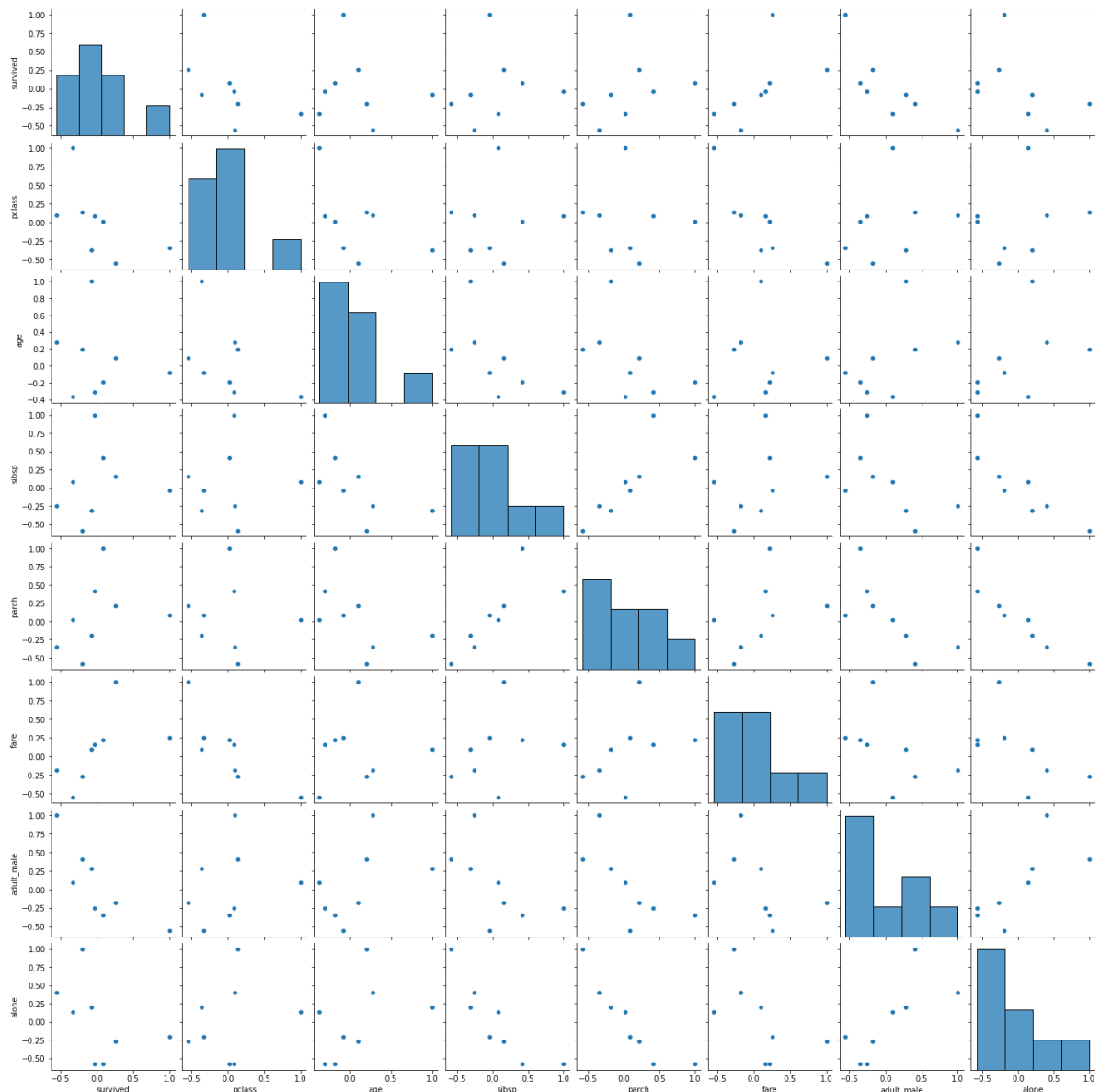
Out[]:

	survived	pclass	age	sibsp	parch	fare	adult_male	alone
survived	1.000000	-0.338481	-0.077221	-0.035322	0.081629	0.257307	-0.557080	-0.203367
pclass	-0.338481	1.000000	-0.369226	0.083081	0.018443	-0.549500	0.094035	0.135207
age	-0.077221	-0.369226	1.000000	-0.308247	-0.189119	0.096067	0.280328	0.198270
sibsp	-0.035322	0.083081	-0.308247	1.000000	0.414838	0.159651	-0.253586	-0.584471
parch	0.081629	0.018443	-0.189119	0.414838	1.000000	0.216225	-0.349943	-0.583398
fare	0.257307	-0.549500	0.096067	0.159651	0.216225	1.000000	-0.182024	-0.271832
adult_male	-0.557080	0.094035	0.280328	-0.253586	-0.349943	-0.182024	1.000000	0.404744
alone	-0.203367	0.135207	0.198270	-0.584471	-0.583398	-0.271832	0.404744	1.000000

```
In [ ]: # pairplot
```

```
sns.pairplot(corr)
```

```
Out[ ]: <seaborn.axisgrid.PairGrid at 0x1c145e0c310>
```



```
In [ ]: # new dataset penguins
penguins = sns.load_dataset('penguins')
penguins.head()
```

```
Out[ ]:
```

	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex
0	Adelie	Torgersen	39.1	18.7	181.0	3750.0	Male
1	Adelie	Torgersen	39.5	17.4	186.0	3800.0	Female
2	Adelie	Torgersen	40.3	18.0	195.0	3250.0	Female
3	Adelie	Torgersen	NaN	NaN	NaN	NaN	NaN
4	Adelie	Torgersen	36.7	19.3	193.0	3450.0	Female

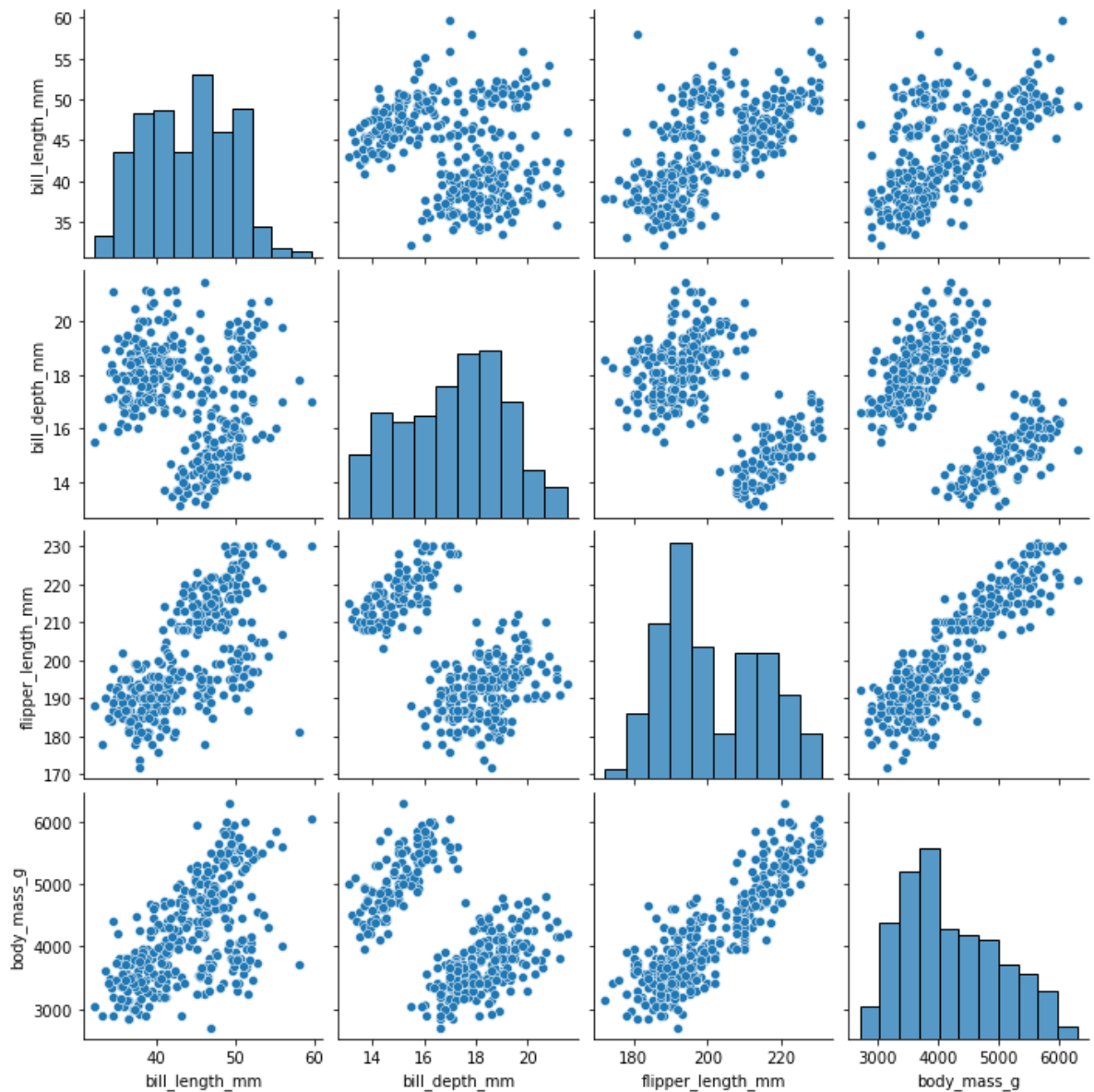
```
In [ ]: penguins.corr(method='spearman')
```

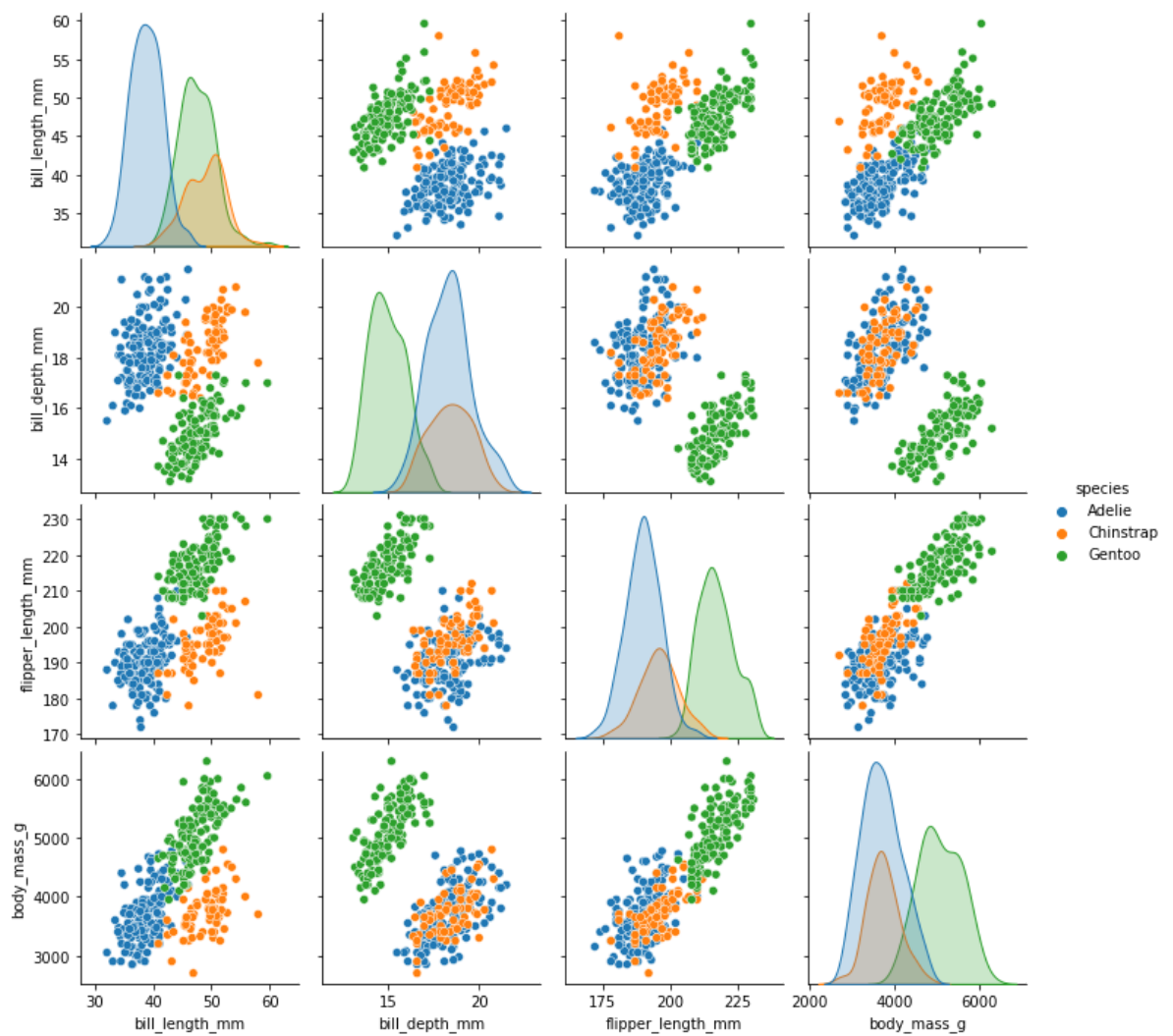

Out[]:

	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g
bill_length_mm	1.000000	-0.221749	0.672772	0.583800
bill_depth_mm	-0.221749	1.000000	-0.523267	-0.432372
flipper_length_mm	0.672772	-0.523267	1.000000	0.839974
body_mass_g	0.583800	-0.432372	0.839974	1.000000

```
In [ ]: sns.pairplot(penguins)
sns.pairplot(penguins, hue='species')
```

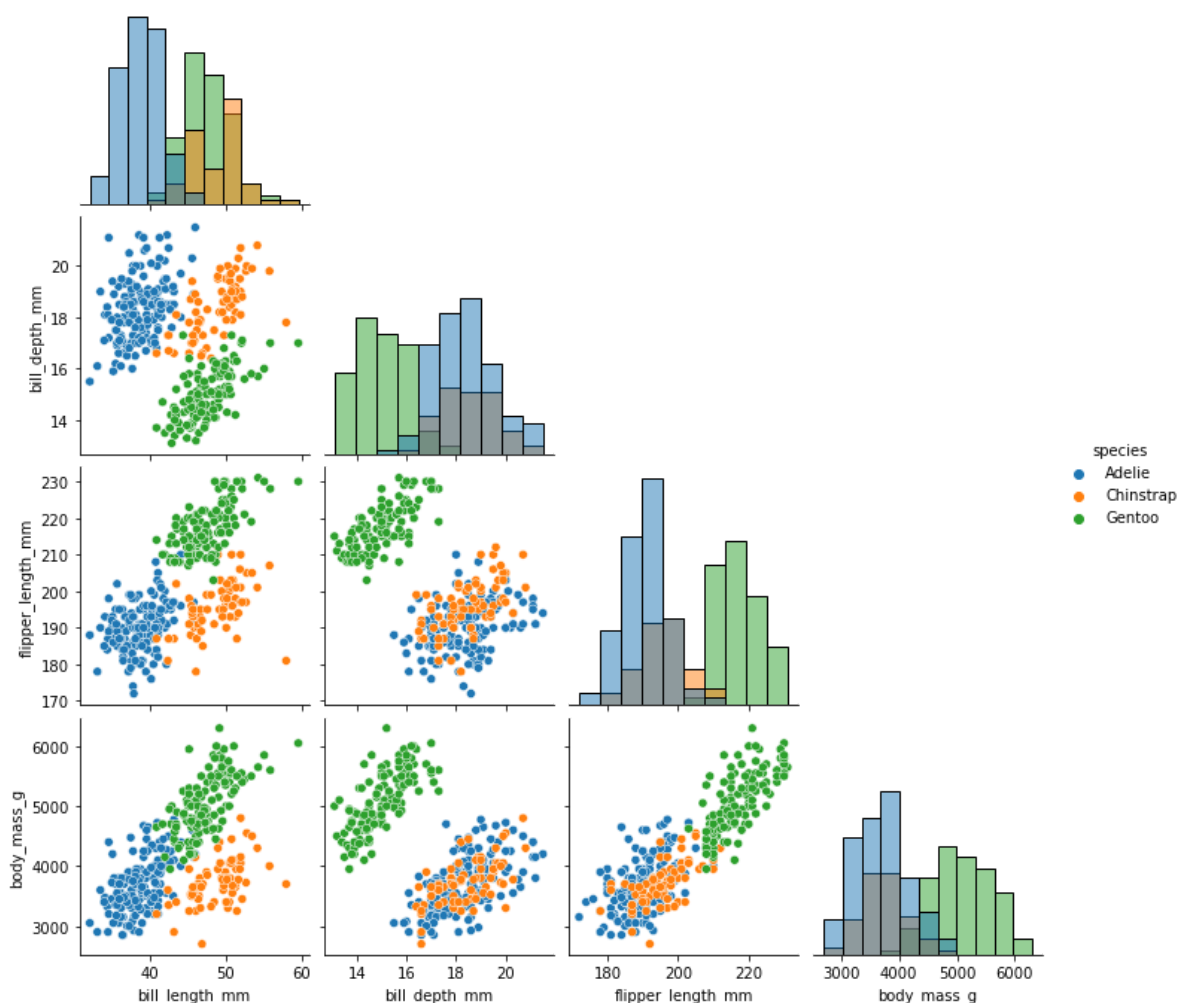
Out[]: <seaborn.axisgrid.PairGrid at 0x1c149f0a280>





```
In [ ]: # make one side
sns.pairplot(penguins, hue='species', diag_kind='hist', corner=True)
```

```
Out[ ]: <seaborn.axisgrid.PairGrid at 0x1c14ba9ef70>
```



```
In [ ]: # calculate pearson's correlation
phool.head()
```

```
Out [ ]:
```

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

```
In [ ]: from scipy.stats import pearsonr
corr, _ = pearsonr(phool['sepal_length'], phool['petal_length'])
print('pearson correlation: %.3f' % corr)

pearson correlation: 0.872
```

```
In [ ]: # same as spearman
from scipy.stats import spearmanr
corr, _ = spearmanr(phool['sepal_length'], phool['petal_length'])
print('pearson correlation: %.3f' % corr)

pearson correlation: 0.882
```

```
In [ ]: # Assignment
# 4 types of plots with all types of correlation
```

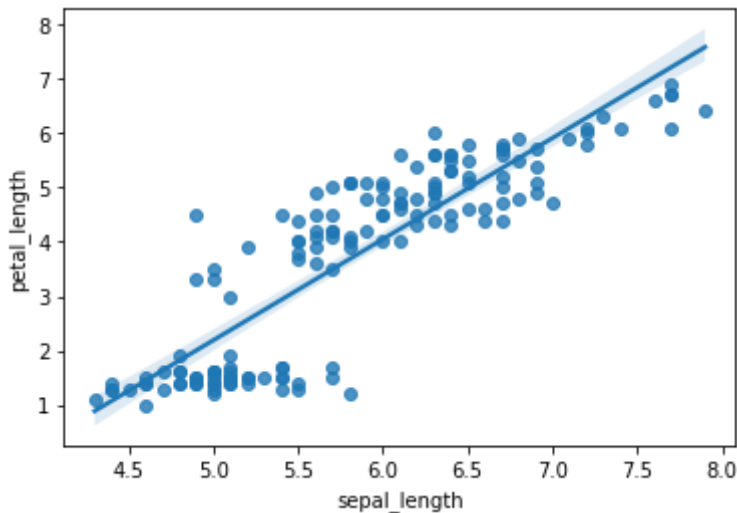
- +ve correlation in phool dataset

```
In [ ]: sns.regplot(phool['sepal_length'], phool['petal_length'], data=phool)
```

C:\Users\Junaid\anaconda3\lib\site-packages\seaborn_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

```
warnings.warn(
```

```
Out[ ]: <AxesSubplot:xlabel='sepal_length', ylabel='petal_length'>
```

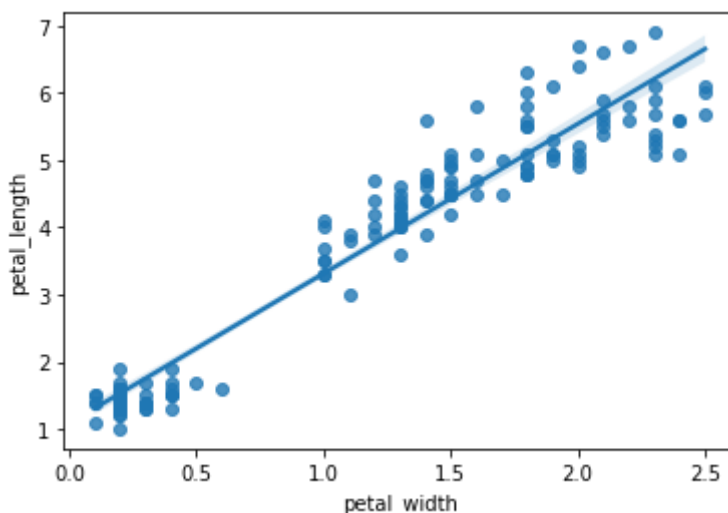


```
In [ ]: sns.regplot(phool['petal_width'], phool['petal_length'], data=phool)
```

C:\Users\Junaid\anaconda3\lib\site-packages\seaborn_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

```
warnings.warn(
```

```
Out[ ]: <AxesSubplot:xlabel='petal_width', ylabel='petal_length'>
```



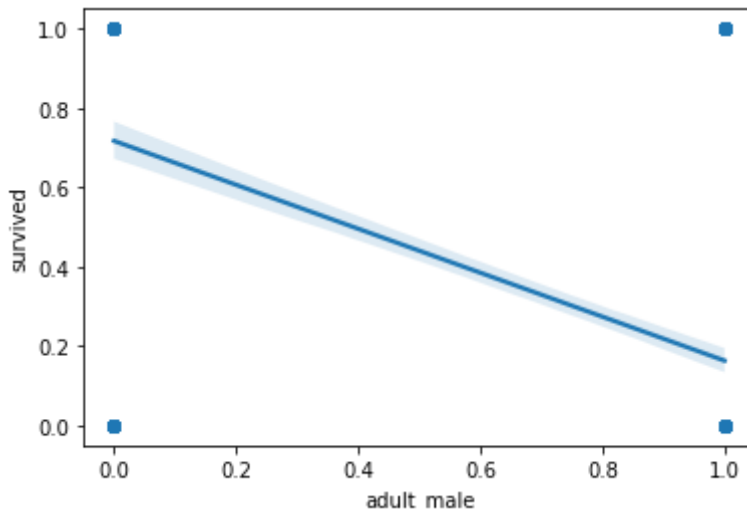
- -ve correlation in kashti dataset

```
In [ ]: sns.regplot(kashti['adult_male'], kashti['survived'], data=kashti)
```

C:\Users\Junaid\anaconda3\lib\site-packages\seaborn_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

```
warnings.warn(
```

Out[]: <AxesSubplot:xlabel='adult_male', ylabel='survived'>

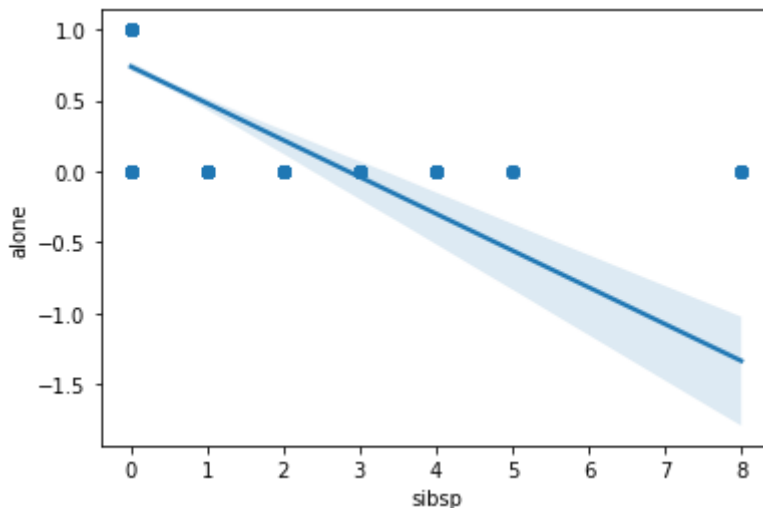


In []: `sns.regplot(kashti['sibsp'], kashti['alone'], data=kashti)`

C:\Users\Junaid\anaconda3\lib\site-packages\seaborn_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

```
warnings.warn(
```

Out[]: <AxesSubplot:xlabel='sibsp', ylabel='alone'>



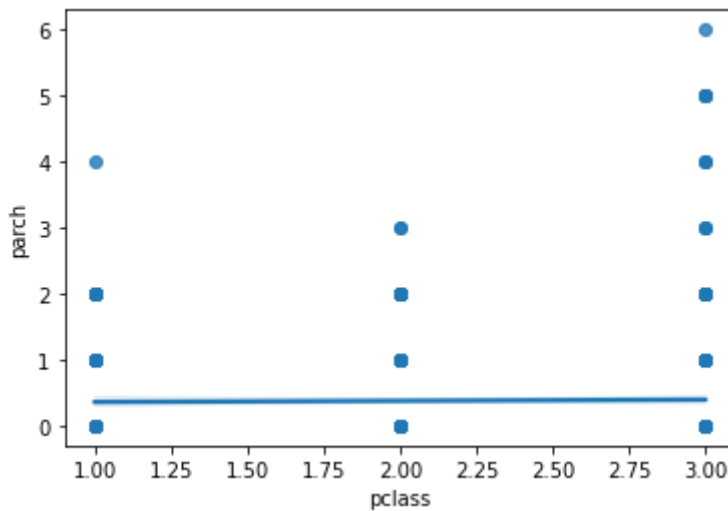
- neutral correlation in kashti dataset

In []: `sns.regplot(kashti['pclass'], kashti['parch'], data=kashti)`

C:\Users\Junaid\anaconda3\lib\site-packages\seaborn_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

```
warnings.warn(
```

Out[]: <AxesSubplot:xlabel='pclass', ylabel='parch'>

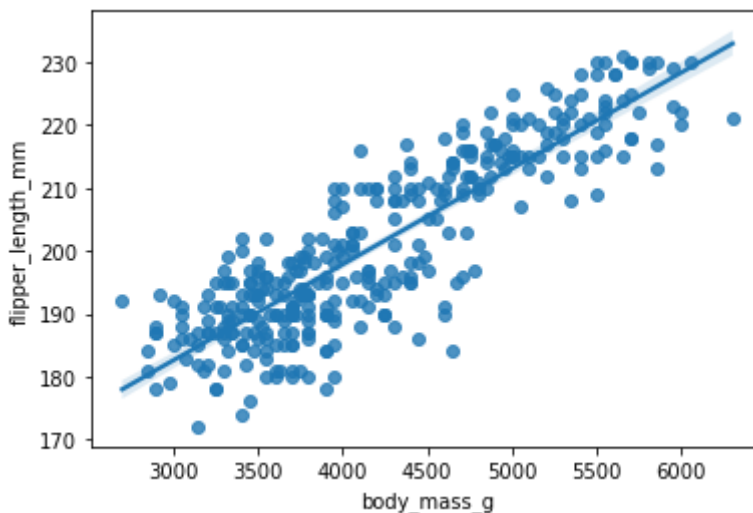


- correlations in penguin dataset

```
In [ ]: # +ve correlation
sns.regplot(penguins['body_mass_g'], penguins['flipper_length_mm'], data=penguins)
```

C:\Users\Junaid\anaconda3\lib\site-packages\seaborn_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

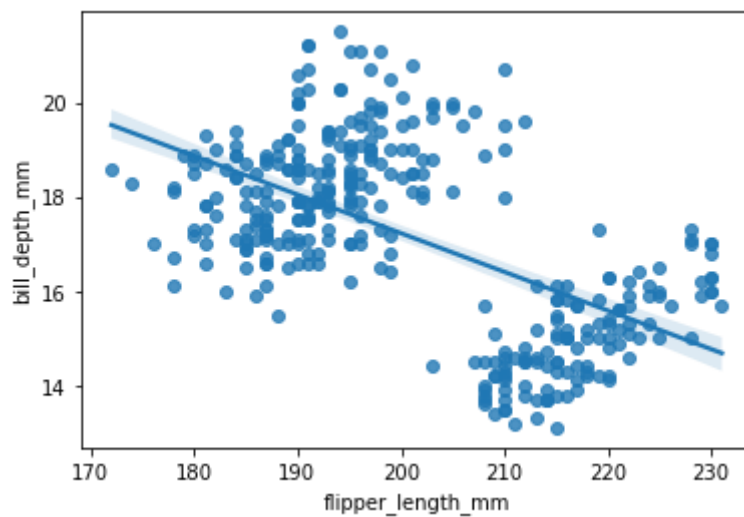
```
warnings.warn(
Out[ ]: <AxesSubplot:xlabel='body_mass_g', ylabel='flipper_length_mm'>
```



```
In [ ]: # -ve correlation
sns.regplot(penguins['flipper_length_mm'], penguins['bill_depth_mm'], data=penguins)
```

C:\Users\Junaid\anaconda3\lib\site-packages\seaborn_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

```
warnings.warn(
Out[ ]: <AxesSubplot:xlabel='flipper_length_mm', ylabel='bill_depth_mm'>
```



```
In [ ]: # zero correlation
sns.regplot(penguins['bill_depth_mm'], penguins['bill_length_mm'], data=penguins)
```

C:\Users\Junaid\anaconda3\lib\site-packages\seaborn_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

warnings.warn(

```
Out[ ]: <AxesSubplot:xlabel='bill_depth_mm', ylabel='bill_length_mm'>
```

