Correlation

- What is correlation?
- Variables within a dataset can be related for lot of reasons.
- Types:
  - Pearson's r (Used for Gaussian distribution)
  - Spearman's rho (Used for non-Gaussian distribution)
  - Kendall's tau (Used for ranking)

For example:

1. One variable could cause or depend on the values of another variable.
2. One variable could be lightly associated with another variable.
3. Two variables could depend on a third known variable.

Possitive Correlation: Both variables change in the same direction (Increase in one results increase in others and vice versa).

Negative Correlation: Variables change in opposite directions (Increase in one results decrease in others and vice versa).

Neutral Correlation: No relationship in the change of the variables (Increase or decrease in one results effectless on other).

Covariance

- Variables can be related by a linear relationship. This is a relationship that is consistently additive across the two data samples.
- This relationship can be summarized 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 in different(negative).
- The magnitude of the covariance is not easily interpreted. A covariance value of zero indicates that both variables are completly independent.

In [ ]:
```
# import libraries
import numpy as np
import pandas as pd
import seaborn as sns
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	deck
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	NaN
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	C
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False	NaN
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False	C
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True	NaN

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

Out[ ]:

	sepal_length	sepal_width	petal_length	petal_width	species

	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

To find Covariance first separate the desired variables then convert them into np.array and then apply function np.cov()

In [ ]:
```
ks_age = kashti['age']
ks_age.to_numpy
```

Out[ ]:
```
<bound method IndexOpsMixin.to_numpy of 0      22.0
1      38.0
2      26.0
3      35.0
4      35.0
       ...
886    27.0
887    19.0
888     NaN
889    26.0
890    32.0
Name: age, Length: 891, dtype: float64>
```

In [ ]:
```
ks_fare = kashti['fare']
ks_fare.to_numpy
```

Out[ ]:
```
<bound method IndexOpsMixin.to_numpy of 0       7.2500
1      71.2833
2       7.9250
3      53.1000
4       8.0500
        ...
886    13.0000
887    30.0000
888    23.4500
889    30.0000
890     7.7500
Name: fare, Length: 891, dtype: float64>
```

In [ ]:
```
cov_mat = np.stack((ks_age, ks_fare), axis = 0)
print(np.cov(cov_mat))
```

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

In [ ]:
```
# Another way
np.cov(kashti['age'], kashti['fare'])
```

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

As it is difficult to interpret **Covariance** so use **Correlation** instead of it.

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

```
<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 [ ]:
```
# Simple correlation
kashti.corr()
```

```
<ipython-input-24-d27344ecd7a1>:2: FutureWarning: The default value of numeric_only i
n DataFrame.corr is deprecated. In a future version, it will default to False. Select
only valid columns or specify the value of numeric_only to silence this warning.
  kashti.corr()
```

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 [ ]:
```
# Pearson'
corr_pearson = kashti.corr(method='pearson') # for Gaussian
```

```
<ipython-input-25-0d9f16e57789>:2: FutureWarning: The default value of numeric_only i
n DataFrame.corr is deprecated. In a future version, it will default to False. Select
only valid columns or specify the value of numeric_only to silence this warning.
  corr_pearson = kashti.corr(method='pearson') # for Gaussian
```

In [ ]:
```
# Spearman'
corr_spearman = kashti.corr(method='spearman') # for non Gaussian
```

```
<ipython-input-26-c50da91a56b0>:2: FutureWarning: The default value of numeric_only i
n DataFrame.corr is deprecated. In a future version, it will default to False. Select
only valid columns or specify the value of numeric_only to silence this warning.
  corr_spearman = kashti.corr(method='spearman') # for non Gaussian
```

See positive Correlation in graph

In [ ]:
```
sns.regplot(kashti['adult_male'], kashti['alone'])
```

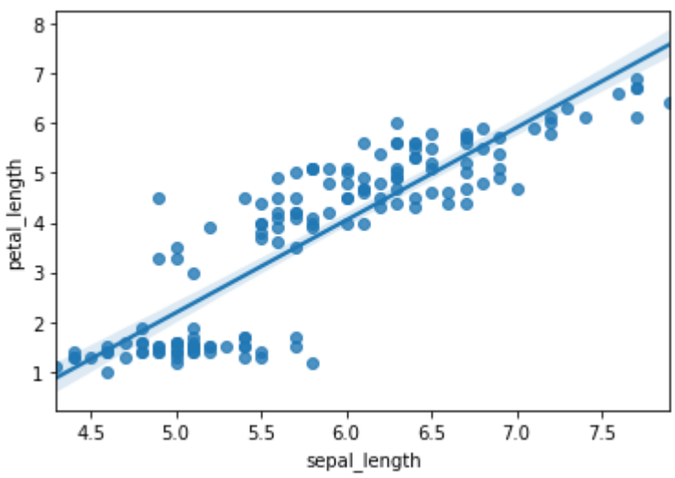
```
c:\Users\kalee\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning:
Pass the following variables as keyword args: x, y. From version 0.12, the only valid
positional argument will be `data`, and passing other arguments without an explicit k
eyword will result in an error or misinterpretation.
  warnings.warn(
```

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

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

```
c:\Users\kalee\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 k
eyword will result in an error or misinterpretation.
  warnings.warn(
```

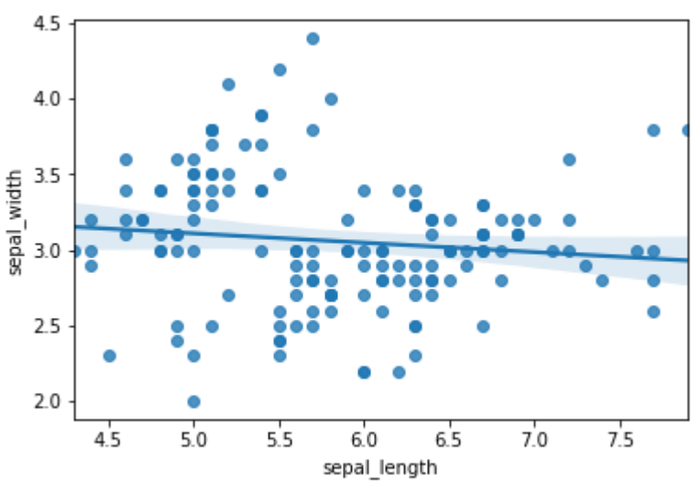
Out[ ]: <AxesSubplot:xlabel='sepal_length', ylabel='petal_length'>



In [ ]:
```
sns.regplot(phool['sepal_length'], phool['sepal_width'])
```

```
c:\Users\kalee\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 k
eyword will result in an error or misinterpretation.
  warnings.warn(
```

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

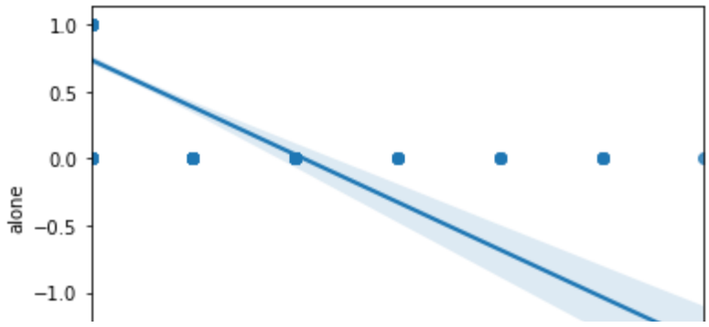


See negative Correlation in graph

In [ ]:
```
sns.regplot(kashti['parch'], kashti['alone'])
```

```
c:\Users\kalee\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 k
eyword will result in an error or misinterpretation.
  warnings.warn(
```

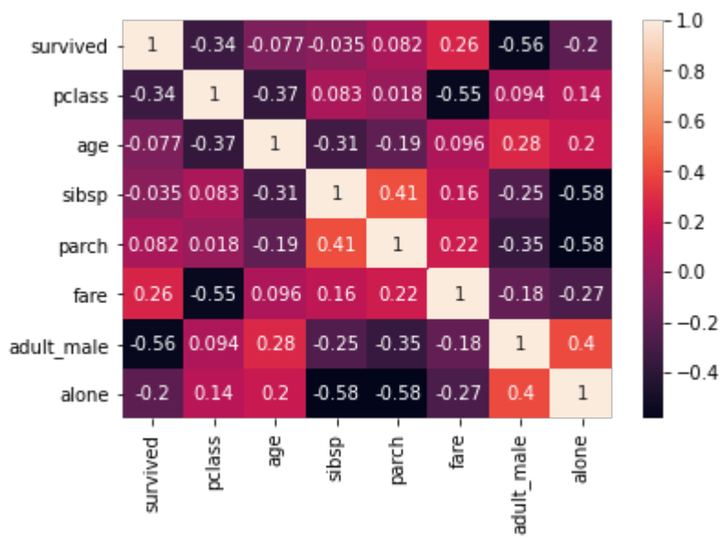
Out[ ]: <AxesSubplot:xlabel='parch', ylabel='alone'>



Now see correlation in Heatmap

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

Out[ ]: <AxesSubplot:>



When **Correlation** is more then 0.6 toward 1.0, than it is called highly **+ve Correlated** and below -.06 is called **-ve Correlated** depending on the data

More better representation of Heatmap

In [ ]:
```
corr_pearson.style.background_gradient(cmap='coolwarm')
```

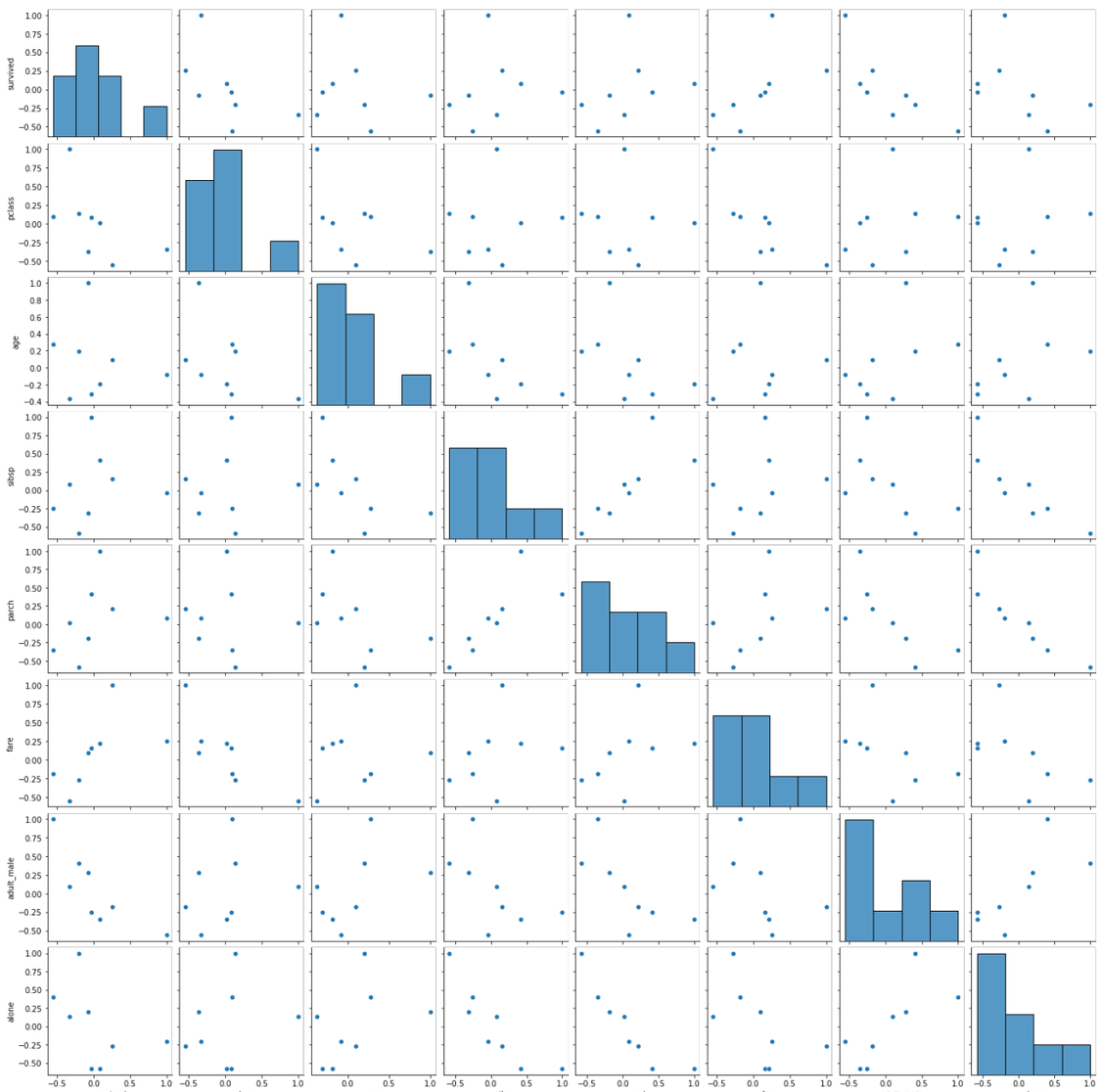
```
---------------------------------------------------------------------------
ImportError                               Traceback (most recent call last)
<ipython-input-32-1926c9d04fbe> in <module>
----> 1 corr_pearson.style.background_gradient(cmap='coolwarm')

c:\Users\kalee\anaconda3\lib\site-packages\pandas\core\frame.py in style(self)
   1262             data with HTML and CSS.
   1263         """
-> 1264         from pandas.io.formats.style import Styler
   1265
   1266         return Styler(self)

c:\Users\kalee\anaconda3\lib\site-packages\pandas\io\formats\style.py in <module>
     54 from pandas.io.formats.format import save_to_buffer
     55
---> 56 jinja2 = import_optional_dependency("jinja2", extra="DataFrame.style requires
jinja2.")
     57
     58 from pandas.io.formats.style_render import (

c:\Users\kalee\anaconda3\lib\site-packages\pandas\compat\_optional.py in import_optio
nal_dependency(name, extra, errors, min_version)
    169                     return None
    170             elif errors == "raise":
--> 171                 raise ImportError(msg)
    172
    173     return module
```

```
ImportError: Pandas requires version '3.0.0' or newer of 'jinja2' (version '2.11.3' c
urrentlv installed).
```

Draw the pairplot now

In [ ]:
```
sns.pairplot(corr_pearson)
```

Out[ ]: <seaborn.axisgrid.PairGrid at 0x20118aae8b0>



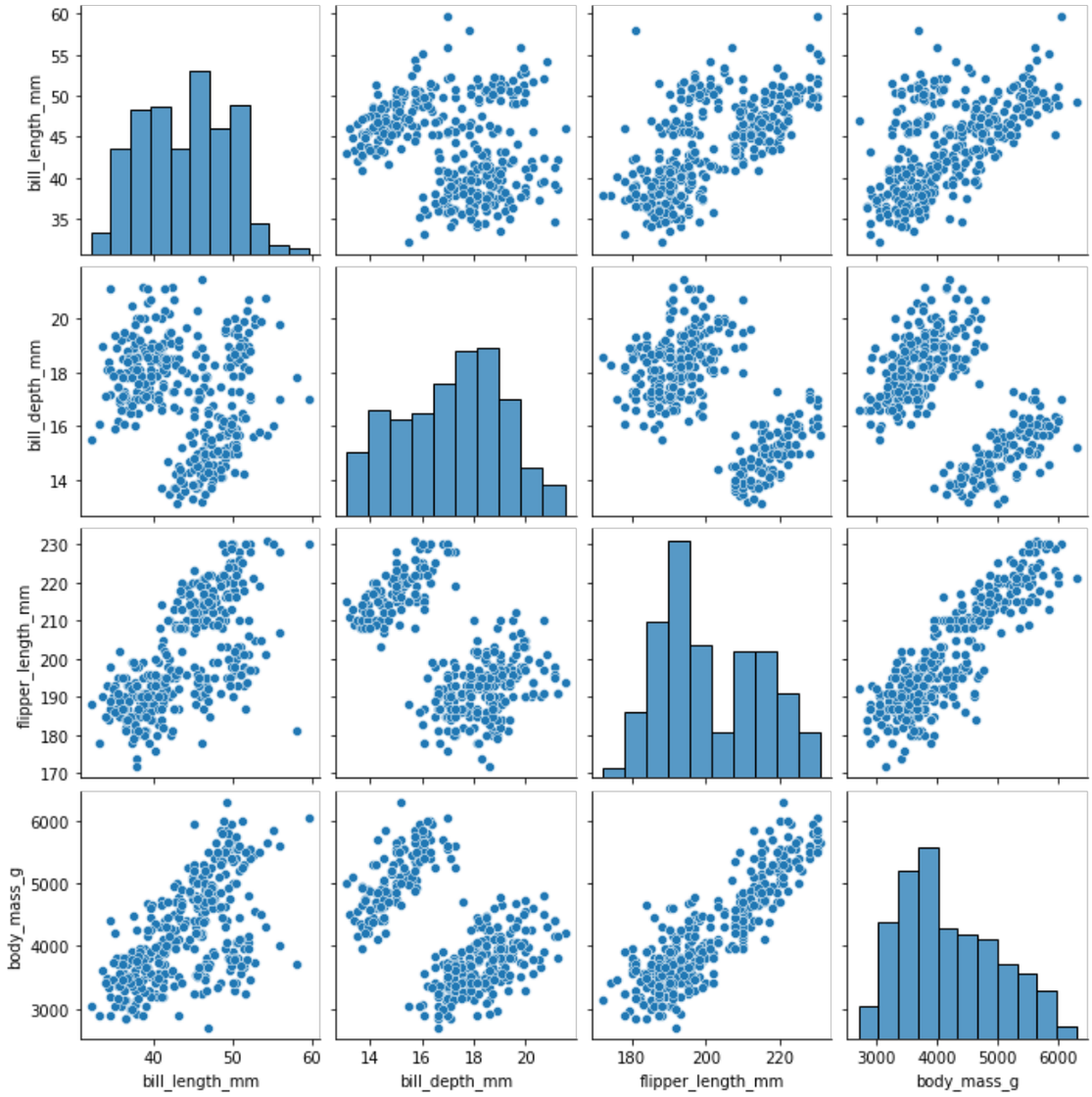
In [ ]:
```
# We can change the points in pairplot based on category
# import a new dataset
penguins = sns.load_dataset('penguins')
penguins.head()
```

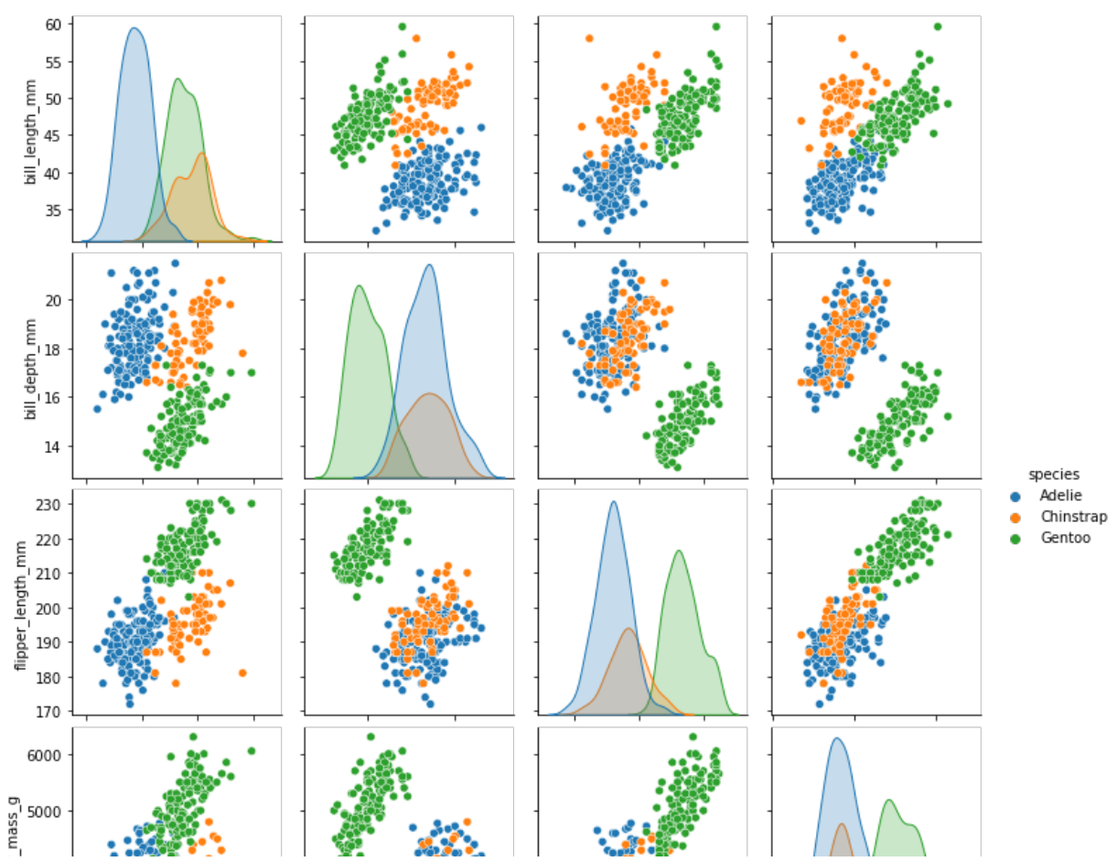
Out[ ]:

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

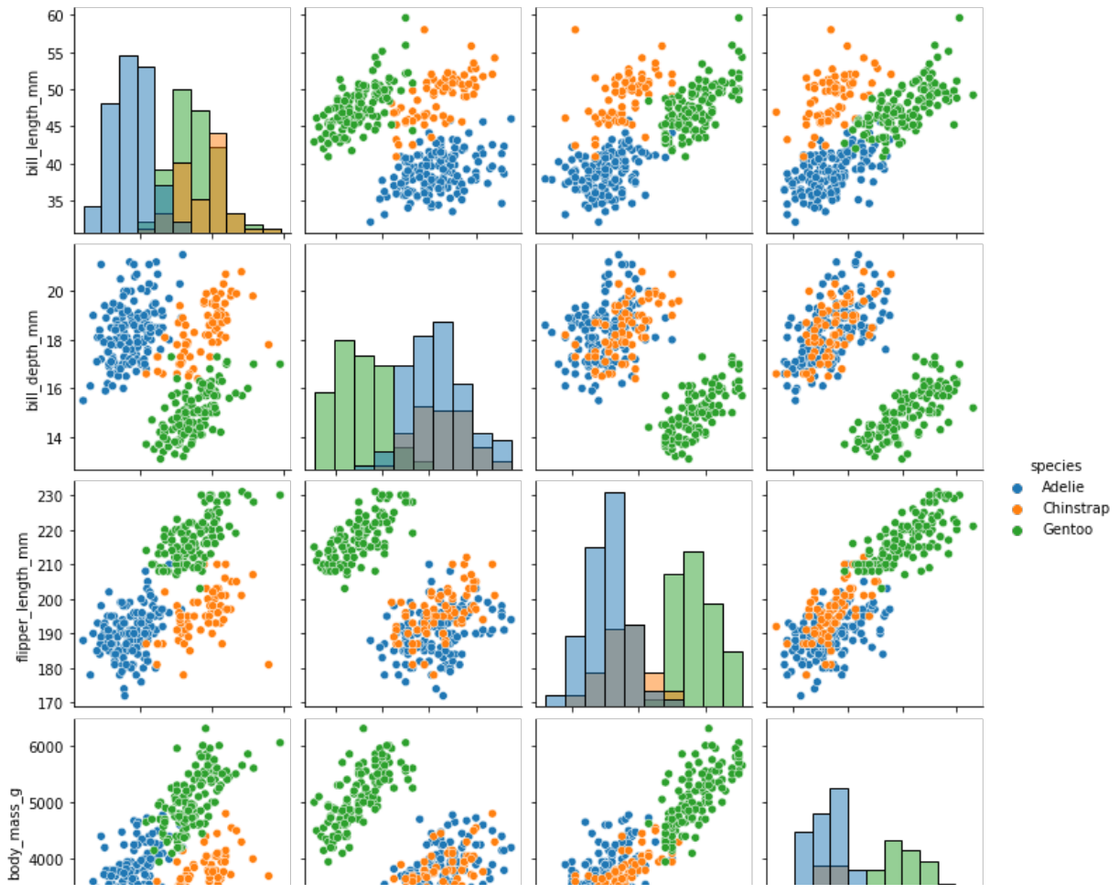
Out[ ]: <seaborn.axisgrid.PairGrid at 0x2011b817670>





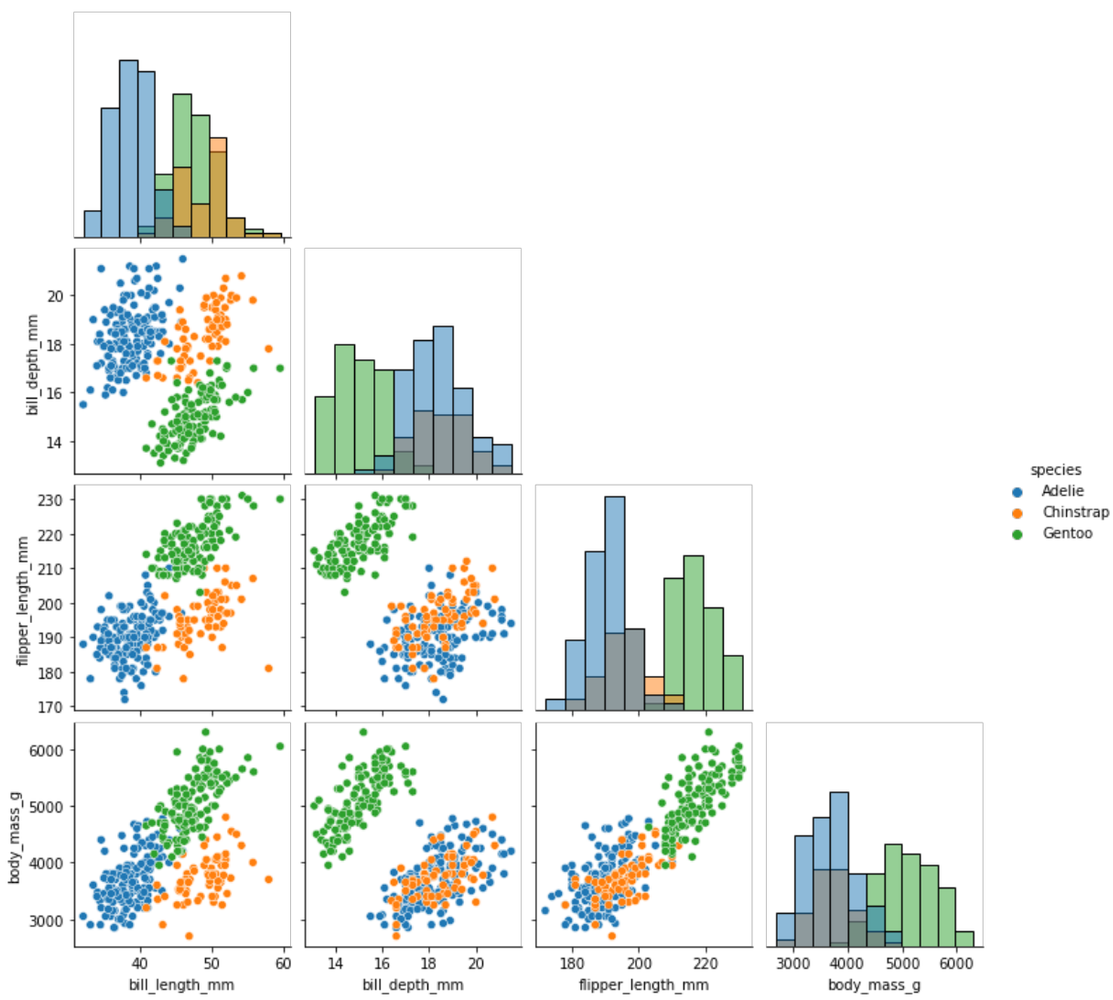
In [ ]:
```
# we can convert this into histogram
sns.pairplot(penguins, hue = 'species', diag_kind='hist')
```

Out[ ]: <seaborn.axisgrid.PairGrid at 0x2011dce36a0>



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

Out[ ]: <seaborn.axisgrid.PairGrid at 0x2011b684250>



Now use of the scipy stats library for pearson's correlation

In [ ]:
```
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 [ ]:
```
# calculate pearson's correlation
from scipy.stats import pearsonr # and spearsman
corr, _ = pearsonr(phool['sepal_length'], phool['petal_length'])
print('Pearsons Correlation: %.3f' % corr)
```

```
Pearsons Correlation: 0.872
```

ASSIGNMENTS

four types of graphs:

1. +ve correlation
2. -ve correlation
3. 0 correlation
4. slightly +ve correlation

Sollutions

Load a new dataset

In [ ]:
```
df = pd.read_csv('D:/Python_Ka_Chilla_Data/Data_Sets/House_Data1.csv')
df.head()
```

Out[ ]:

	Month	E. Bill	Gas Bill	Grocery	Milk	Fruit	Meet	Wegetables	Medicines	Eid_guests_pocketmonies
0	May	1287	250	26000	7200	4580	3800	2310	6580	11000
1	Jun	7309	280	7200	7000	3400	4100	2500	9340	100500
2	Jul	11490	270	11500	7350	4800	3650	2700	2960	52710
3	Aug	19184	250	5400	7200	4100	4000	2200	2000	70650
4	Sep	20065	260	12310	6900	5200	4700	2400	3450	34560

In [ ]:
```
df.columns
```

Out[ ]:
```
Index(['Month', 'E. Bill', 'Gas Bill', 'Grocery', 'Milk', 'Fruit', 'Meet',
       'Wegetables', 'Medicines', 'Eid_guests_pocketmonies', 'Maintenance',
       'house_wares', 'Inflation rate', 'Buying Power(PKR)', 'Saving'],
      dtype='object')
```

In [ ]:
```
corr_df = df.corr('pearson')
corr_df
```

```
<ipython-input-42-b57e8b96ad48>:1: FutureWarning: The default value of numeric_only i
n DataFrame.corr is deprecated. In a future version, it will default to False. Select
only valid columns or specify the value of numeric_only to silence this warning.
  corr_df = df.corr('pearson')
```

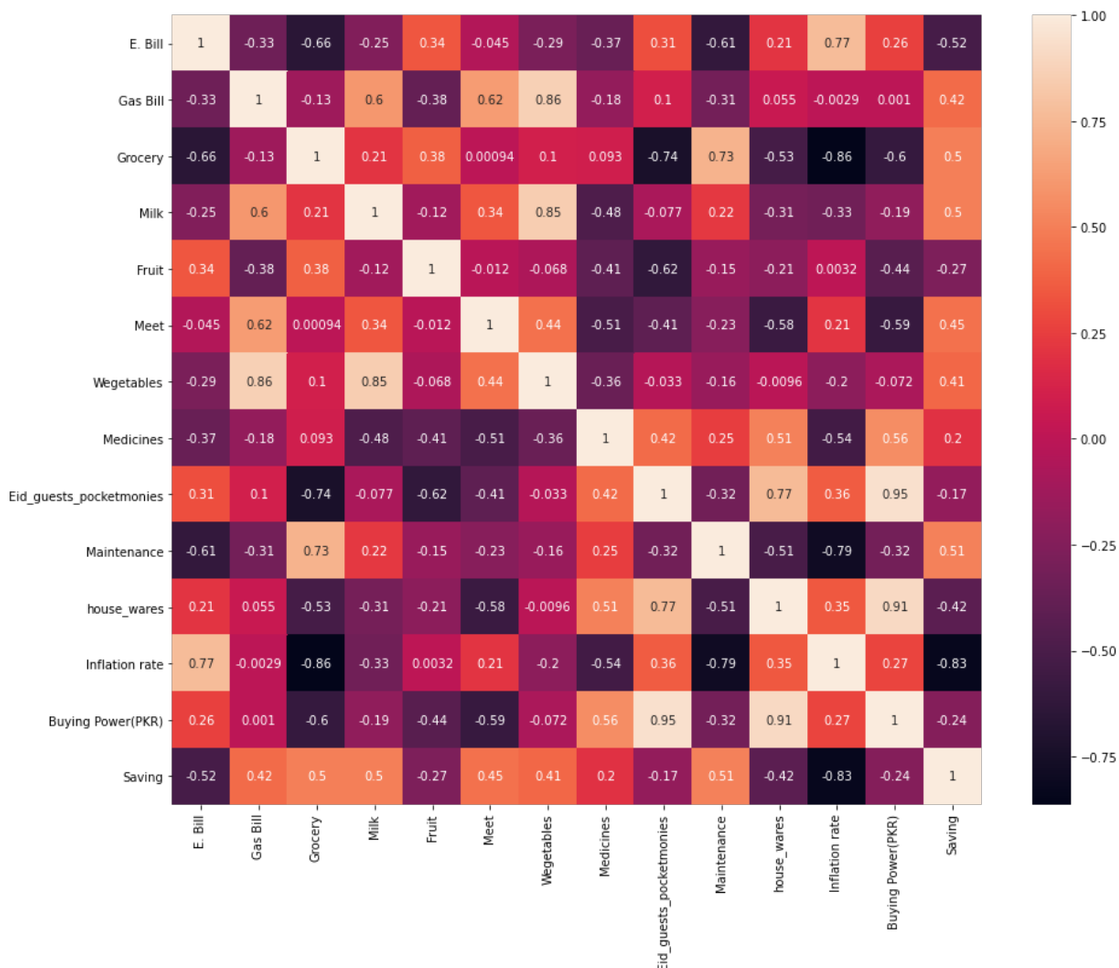
Out[ ]:

	E. Bill	Gas Bill	Grocery	Milk	Fruit	Meet	Wegetables

	E. Bill	Gas Bill	Grocery	Milk	Fruit	Meet	Wegetables
E. Bill	1.000000	-0.333596	-0.658929	-0.252684	0.338906	-0.044632	-0.287951
Gas Bill	-0.333596	1.000000	-0.134358	0.600796	-0.381480	0.623686	0.864663
Grocery	-0.658929	-0.134358	1.000000	0.210455	0.376687	0.000939	0.100573
Milk	-0.252684	0.600796	0.210455	1.000000	-0.117434	0.341335	0.847616
Fruit	0.338906	-0.381480	0.376687	-0.117434	1.000000	-0.012416	-0.067974
Meet	-0.044632	0.623686	0.000939	0.341335	-0.012416	1.000000	0.443741
Wegetables	-0.287951	0.864663	0.100573	0.847616	-0.067974	0.443741	1.000000
Medicines	-0.367530	-0.175674	0.093385	-0.482648	-0.408359	-0.507341	-0.360418
Eid_guests_pocketmonies	0.310057	0.104797	-0.738716	-0.077111	-0.616861	-0.409191	-0.033228
Maintenance	-0.611900	-0.310782	0.728834	0.224218	-0.150142	-0.228264	-0.156502
house_wares	0.208273	0.055266	-0.525829	-0.306631	-0.205594	-0.575040	-0.009608
Inflation rate	0.767361	-0.002879	-0.863095	-0.330068	0.003213	0.211376	-0.201887

In [ ]:
```
plt.figure(figsize= (15, 12))
sns.heatmap(corr_df, annot=True)
```

Out[ ]: <AxesSubplot:>

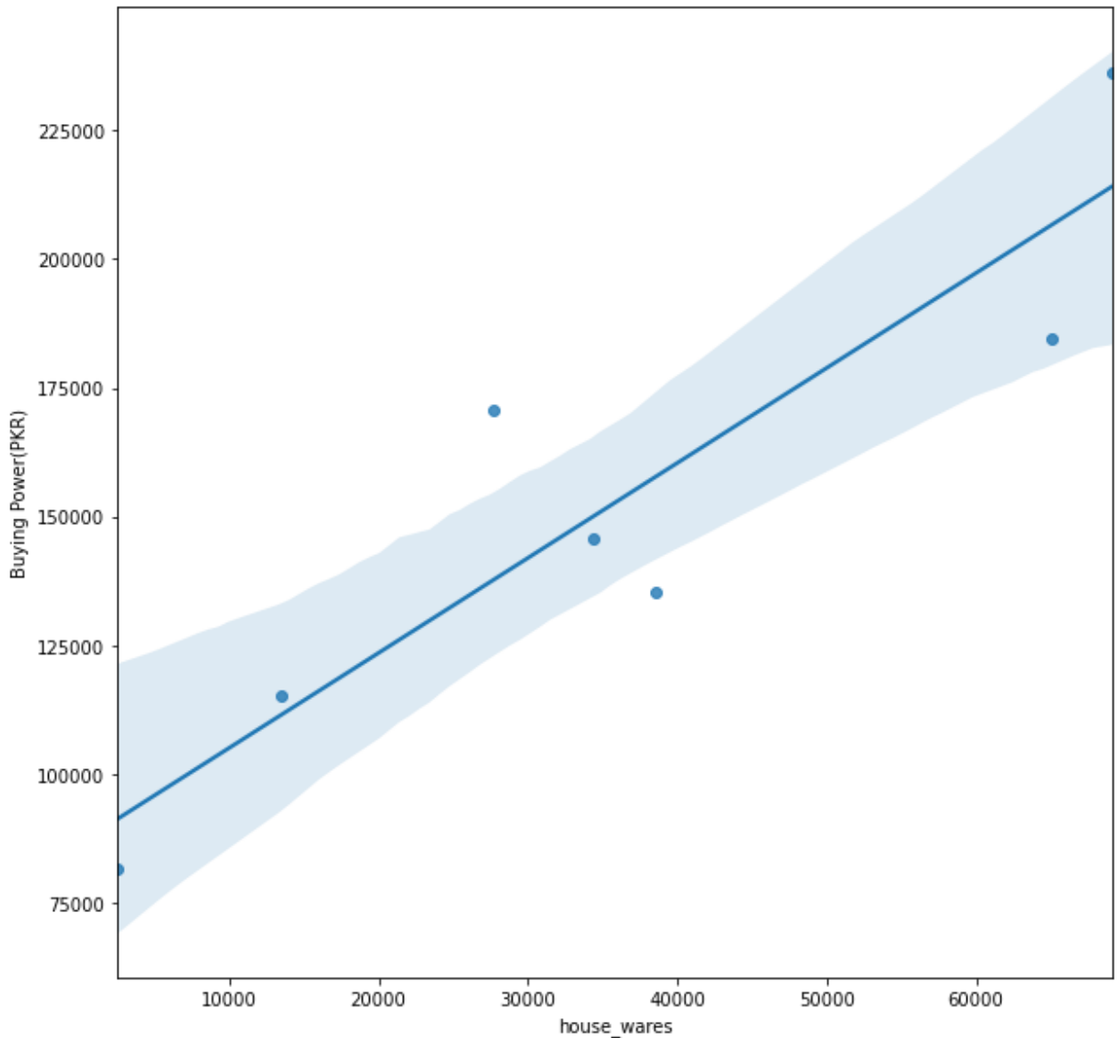


1. +ve correlation

In [ ]:
```
plt.figure(figsize=(10,10))
sns.regplot(df['house_wares'], df['Buying Power(PKR)'])
```

```
c:\Users\kalee\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 k
eyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[ ]: <AxesSubplot:xlabel='house_wares', ylabel='Buying Power(PKR)'>



In [ ]:
```
plt.figure(figsize=(10,10))
sns.regplot(df['Eid_guests_pocketmonies'], df['Buying Power(PKR)'])
```

```
c:\Users\kalee\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 k
eyword will result in an error or misinterpretation.
  warnings.warn(
```

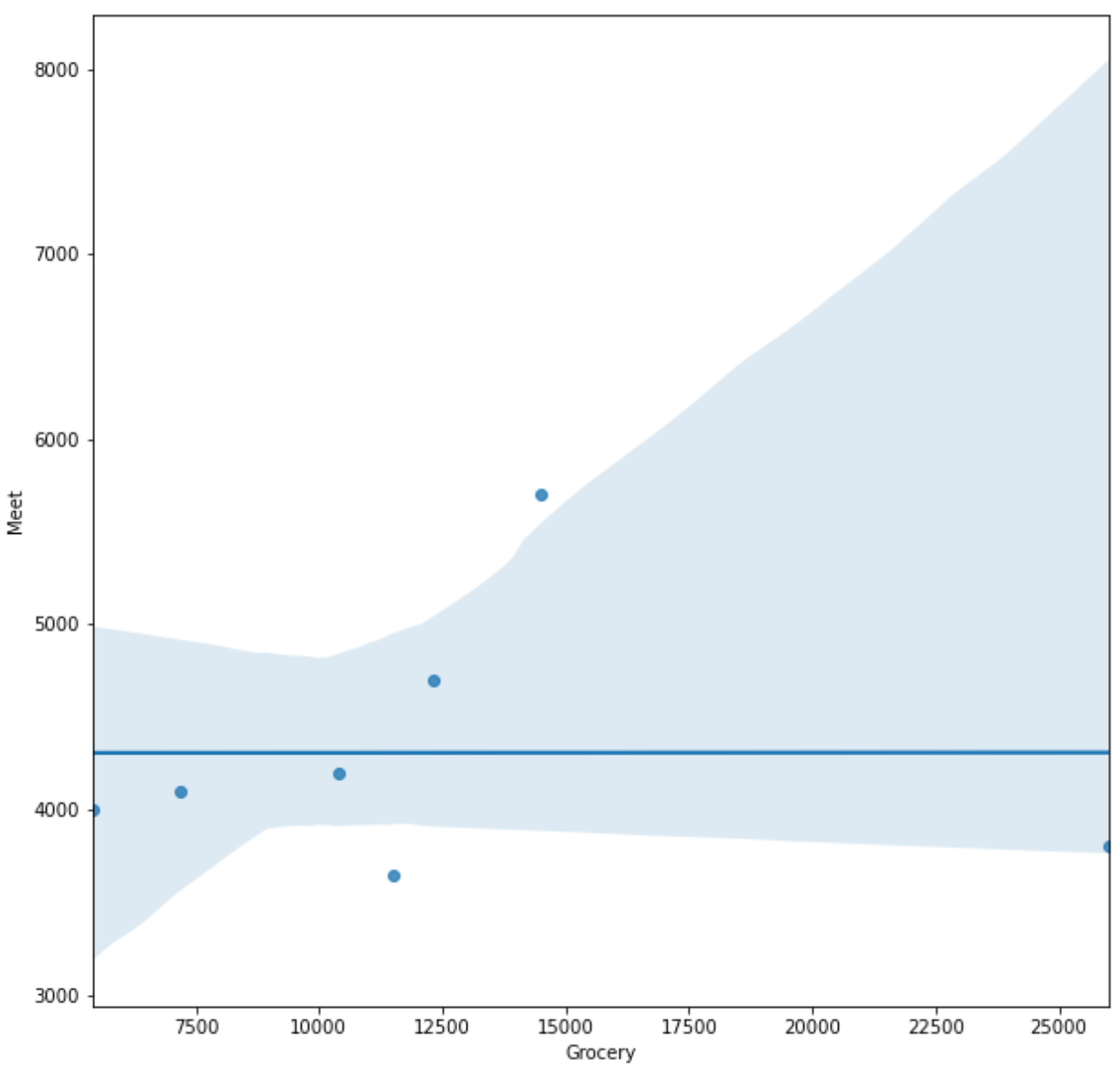
Out[ ]: <AxesSubplot:xlabel='Eid_guests_pocketmonies', ylabel='Buying Power(PKR)'>

2. 0 correlation

In [ ]:
```
plt.figure(figsize=(10,10))
sns.regplot(df['Grocery'], df['Meet'])
```

```
c:\Users\kalee\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 k
eyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[ ]: <AxesSubplot:xlabel='Grocery', ylabel='Meet'>

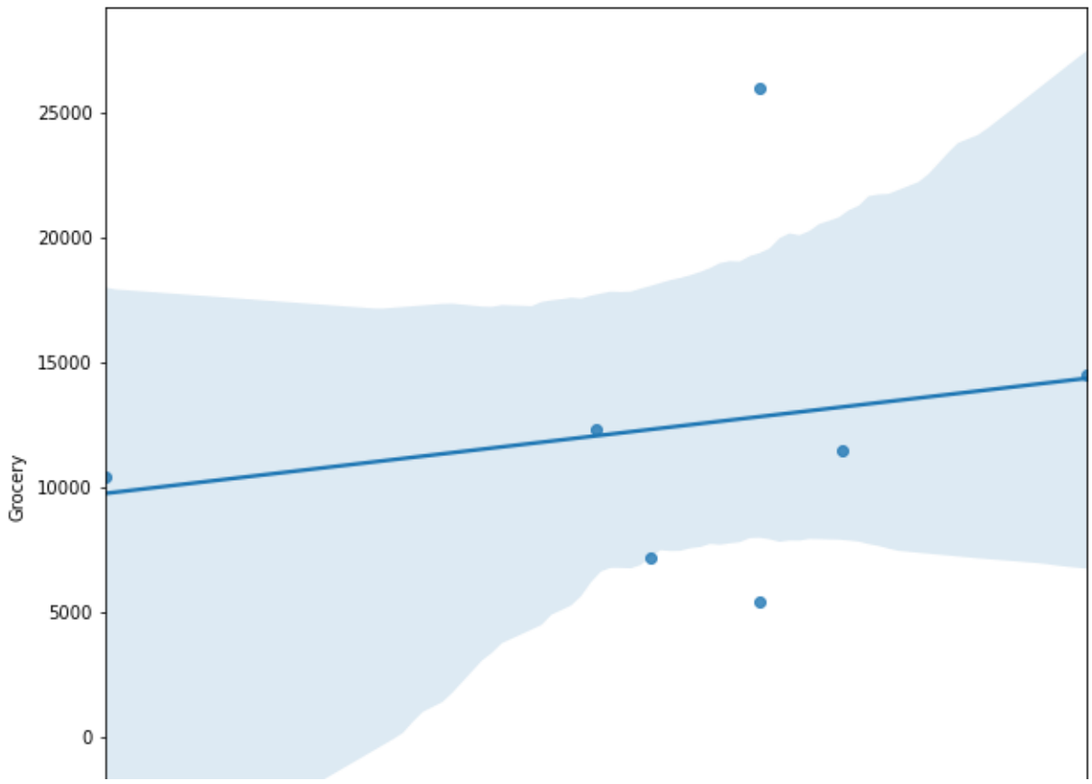


3. Slightly +ve Correlation

In [ ]:
```
plt.figure(figsize=(10,10))
sns.regplot(df['Milk'], df['Grocery'])
```

```
c:\Users\kalee\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 k
eyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[ ]: <AxesSubplot:xlabel='Milk', ylabel='Grocery'>

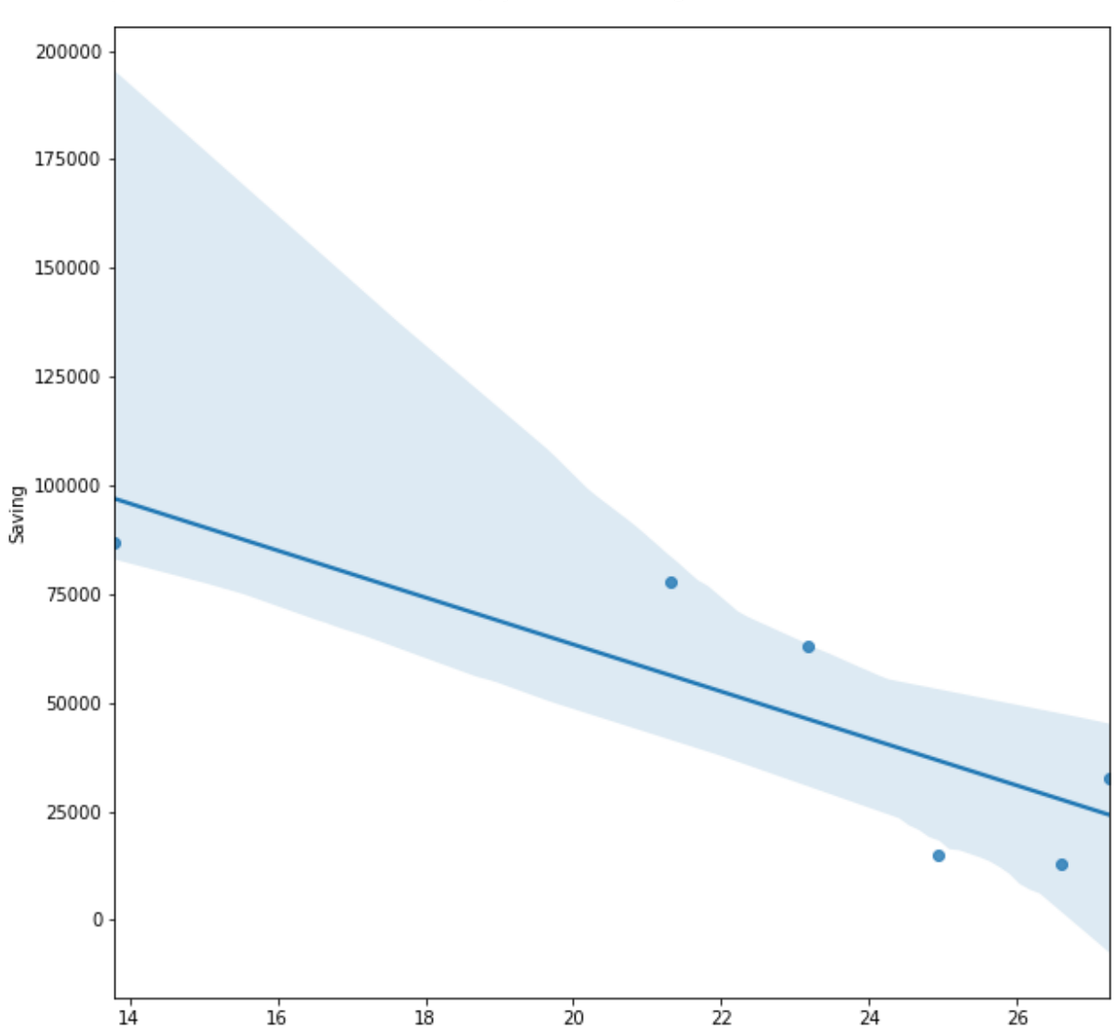


4. -ve Correlation

In [ ]:
```
plt.figure(figsize=(10,10))
sns.regplot(df['Inflation rate'], df['Saving'])
```

```
c:\Users\kalee\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 k
eyword will result in an error or misinterpretation.
  warnings.warn(
```

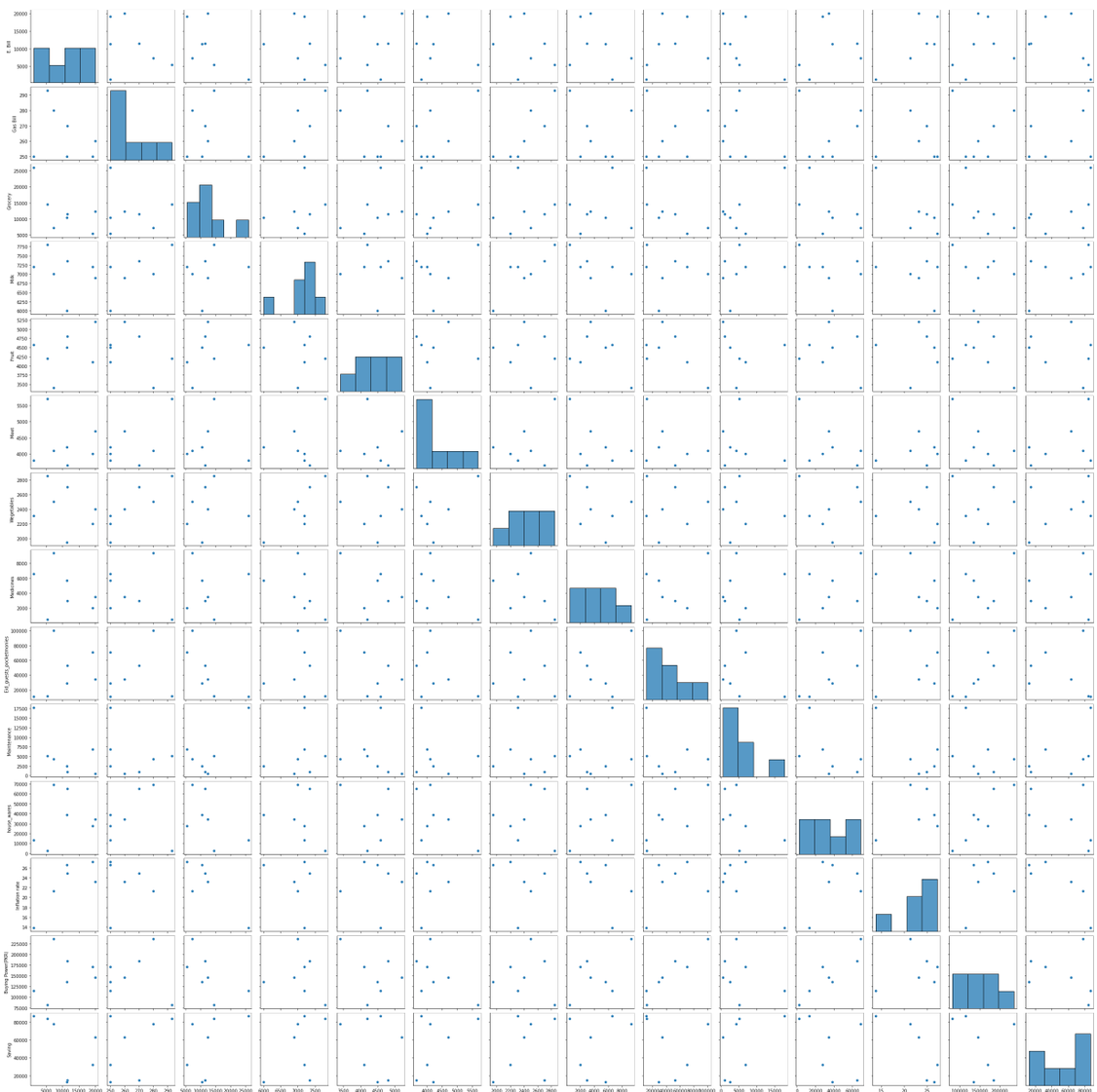
Out[ ]: <AxesSubplot:xlabel='Inflation rate', ylabel='Saving'>

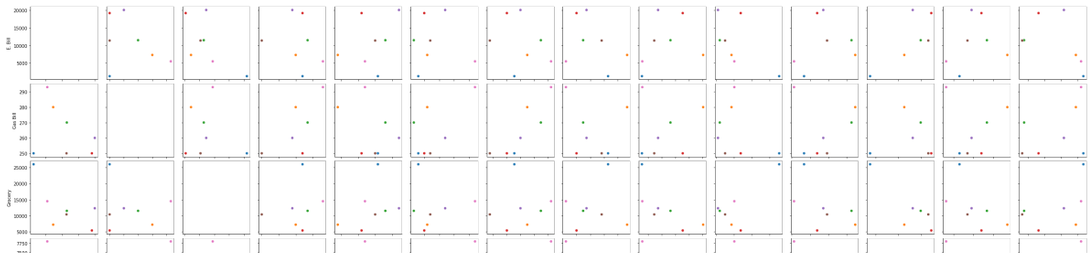


In [ ]:
```
import warnings
warnings.filterwarnings("ignore")
```

In [ ]:
```
sns.pairplot(df)
sns.pairplot(df, hue = 'Month')
```

Out[ ]: <seaborn.axisgrid.PairGrid at 0x201218978b0>





In [ ]:
```
sns.pairplot(df, hue = 'Month', diag_kind='hist')
```

Out[ ]: <seaborn.axisgrid.PairGrid at 0x2012899c040>