NAME:A.VIJAY SANKAR REDDY

REG NO:21BPS1407

ASSINMENT 3

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
df=pd.read_csv('/content/penguins_size.csv');
```

```
df.head()
```

	species	island	culmen_length_mm	culmen_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

```
df.shape
```

```
(344, 7)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 344 entries, 0 to 343
Data columns (total 7 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   species            344 non-null    object 
 1   island             344 non-null    object 
 2   culmen_length_mm   342 non-null    float64
 3   culmen_depth_mm    342 non-null    float64
 4   flipper_length_mm  342 non-null    float64
 5   body_mass_g        342 non-null    float64
 6   sex                334 non-null    object 
```

```
dtypes: float64(4), object(3)
memory usage: 18.9+ KB
```

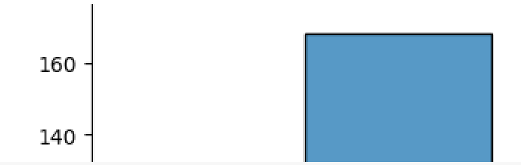
```
df.describe()
```

	culmen_length_mm	culmen_depth_mm	flipper_length_mm	body_mass_g
count	342.000000	342.000000	342.000000	342.000000
mean	43.921930	17.151170	200.915205	4201.754386
std	5.459584	1.974793	14.061714	801.954536
min	32.100000	13.100000	172.000000	2700.000000
25%	39.225000	15.600000	190.000000	3550.000000
50%	44.450000	17.300000	197.000000	4050.000000
75%	48.500000	18.700000	213.000000	4750.000000
max	59.600000	21.500000	231.000000	6300.000000

univariate analysis

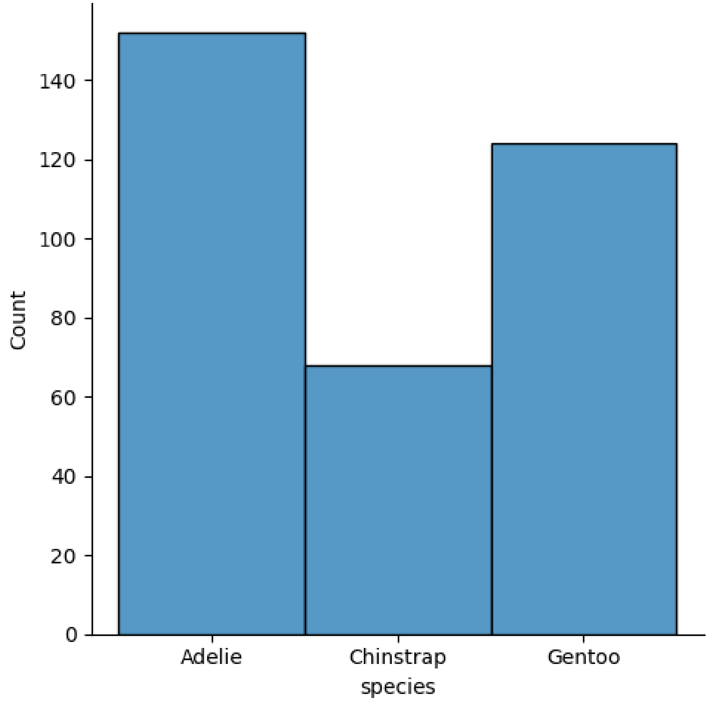
```
sns.displot(df.island)
```

```
<seaborn.axisgrid.FacetGrid at 0x78c3542c7610>
```



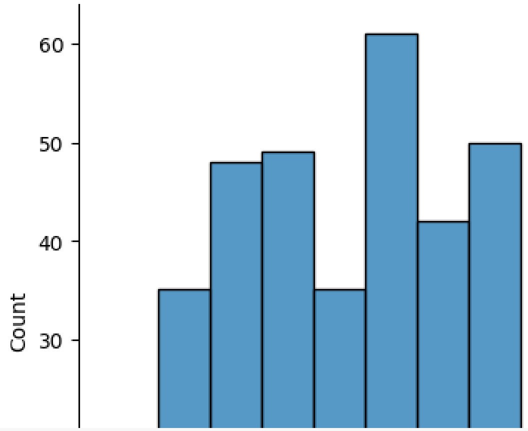
```
sns.displot(df.species)
```

```
<seaborn.axisgrid.FacetGrid at 0x78c3546e6260>
```



```
sns.displot(df.culmen_length_mm)
```

```
<seaborn.axisgrid.FacetGrid at 0x78c3542c7820>
```



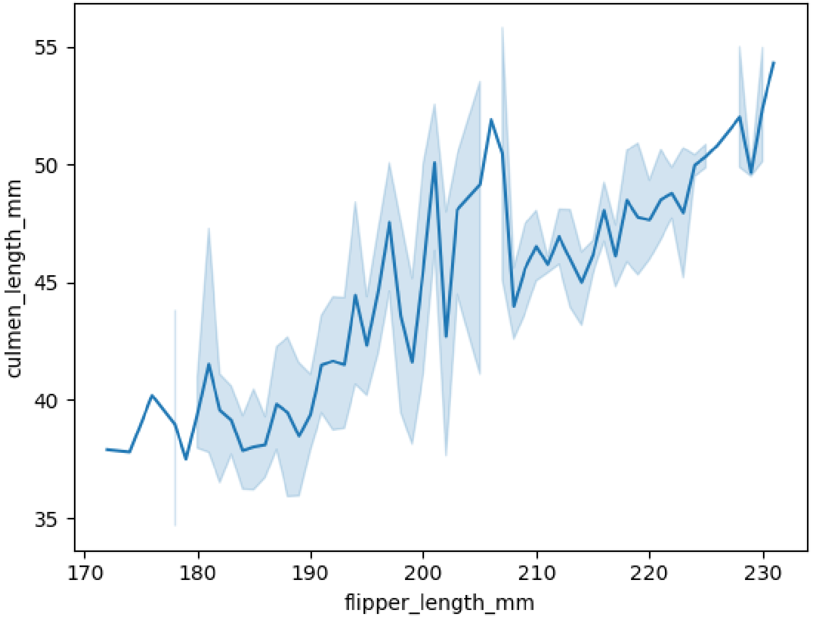
```
sns.distplot(df.flipper_length_mm   )
```

```
<ipython-input-13-ae65ebdd98e7>:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.


similar flexibility) or `histplot` (an axes-level function for histograms).
```

bivariant analysis

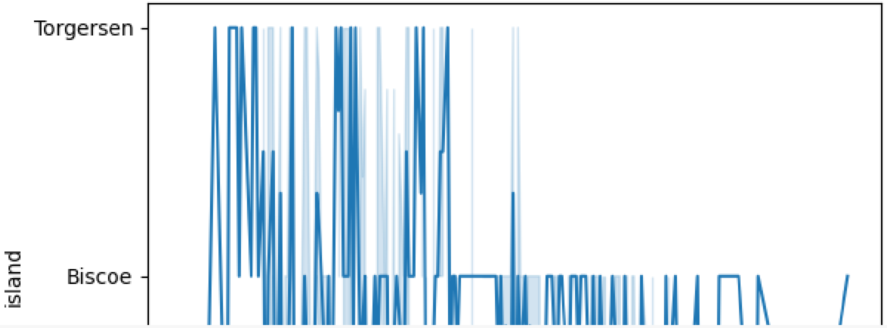
```
sns.lineplot(x=df.flipper_length_mm,y=df.culmen_length_mm)
```

```
<Axes: xlabel='flipper_length_mm', ylabel='culmen_length_mm'>
```



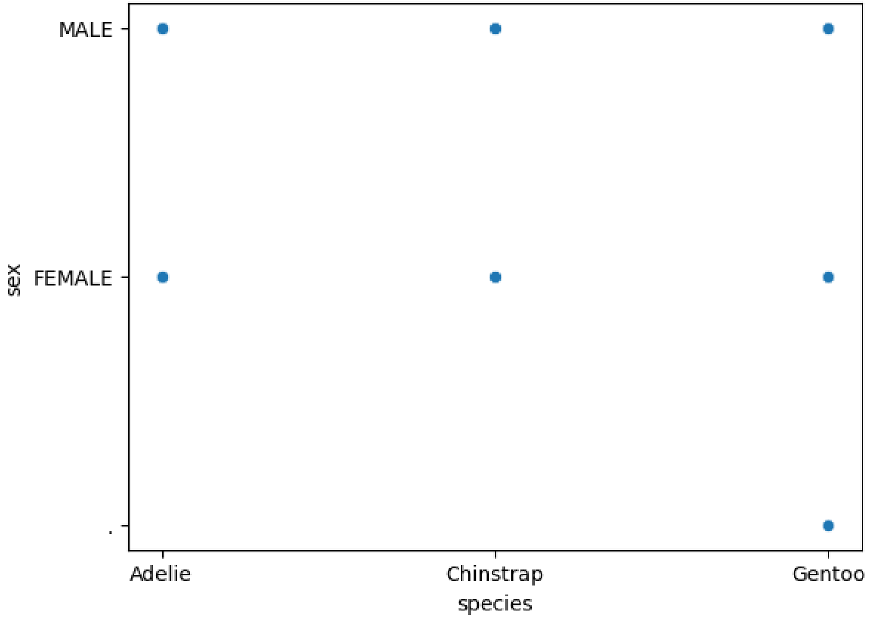
```
sns.lineplot(x=df.culmen_length_mm,y=df.island)
```

```
<Axes: xlabel='culmen_length_mm', ylabel='island'>
```



```
sns.scatterplot(x=df.species,y=df.sex)
```

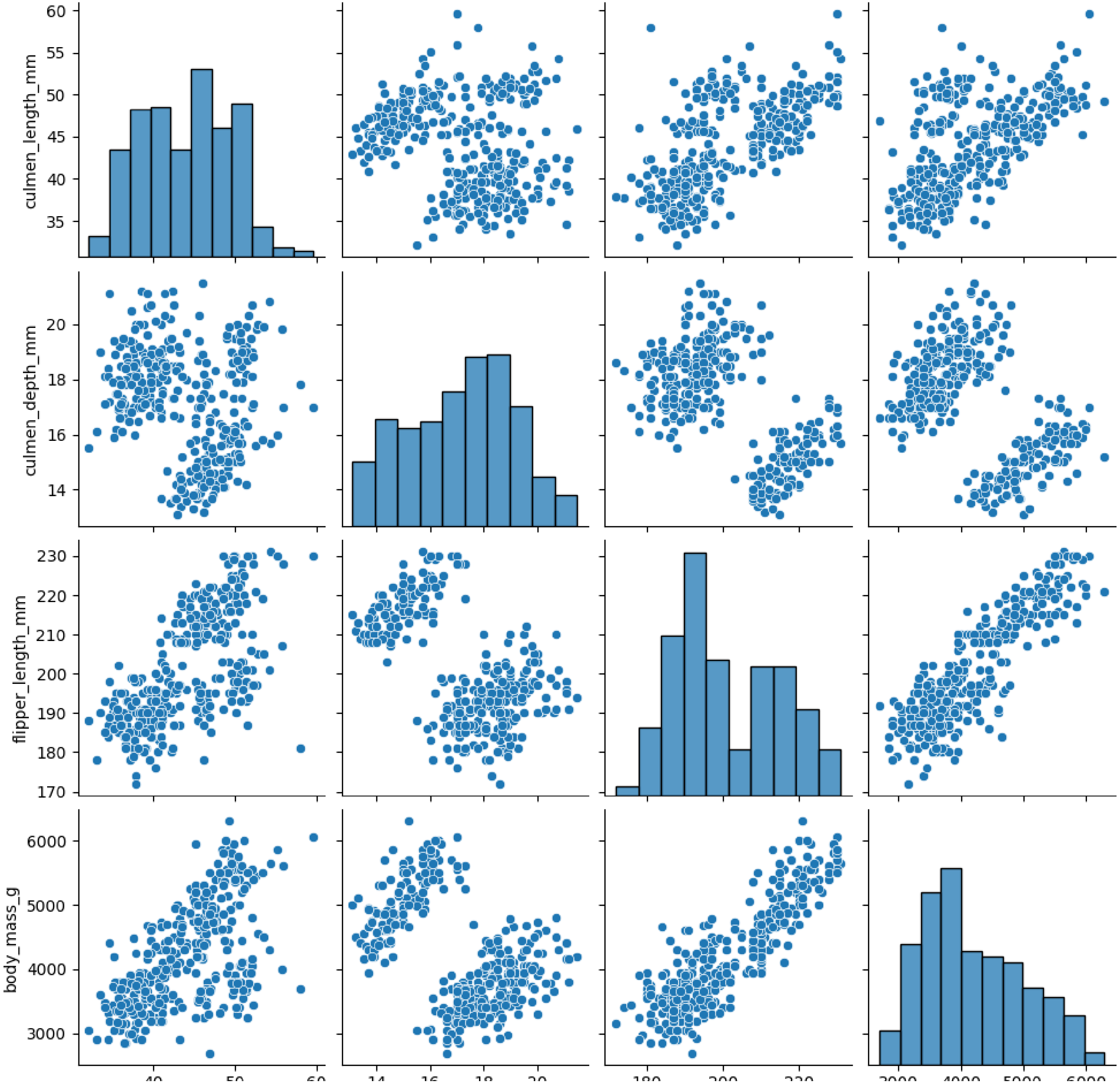
```
<Axes: xlabel='species', ylabel='sex'>
```



Multivariant analysis

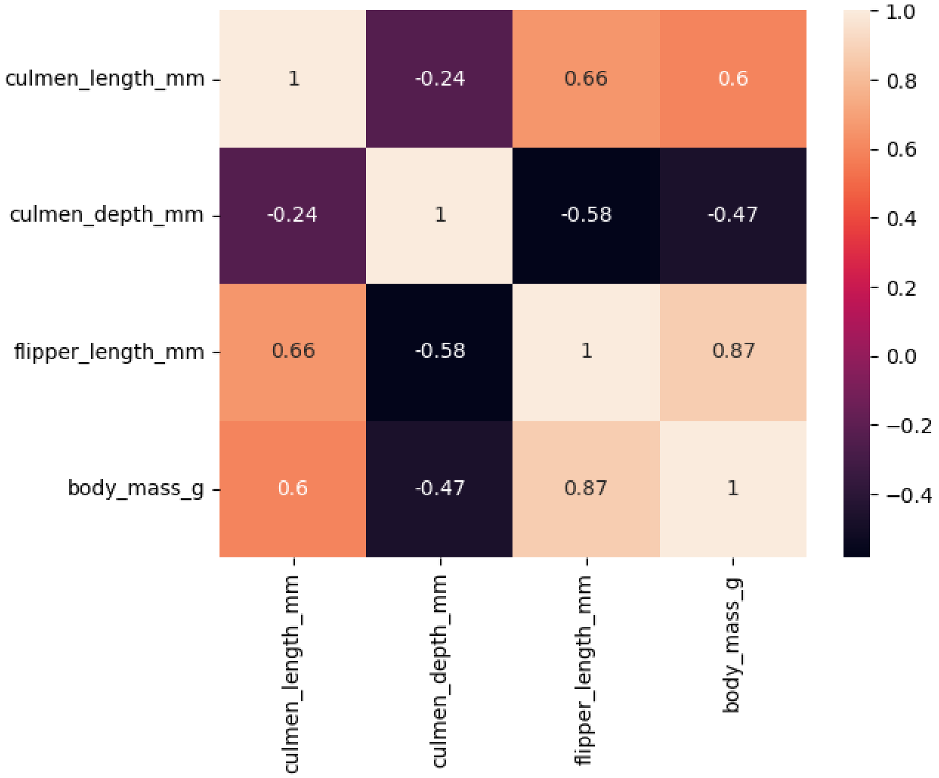
```
sns.pairplot(df)
```

```
<seaborn.axisgrid.PairGrid at 0x78c34f890370>
```


culmen_length_mm
60
55
50
45
40
35
culmen_depth_mm
20
18
16
14
flipper_length_mm
230
220
210
200
190
180
170
body_mass_g
6000
5000
4000
3000
40
50
60
14
16
18
20
180
200
220
3000
4000
5000
6000

```
sns.heatmap(df.corr(),annot=True)
```

```
<ipython-input-19-8df7bcac526d>:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select only
  sns.heatmap(df.corr(),annot=True)
<Axes: >
```



Double-click (or enter) to edit

descriptive statistics

```
df.describe()
```

	culmen_length_mm	culmen_depth_mm	flipper_length_mm	body_mass_g
count	342.000000	342.000000	342.000000	342.000000
mean	43.921930	17.151170	200.915205	4201.754386
std	5.459584	1.974793	14.061714	801.954536
min	32.100000	13.100000	172.000000	2700.000000
25%	39.225000	15.600000	190.000000	3550.000000
50%	44.450000	17.300000	197.000000	4050.000000

```
df.head()
```

	species	island	culmen_length_mm	culmen_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

```
df.isnull().any()
```

```
species              False
island               False
culmen_length_mm      True
culmen_depth_mm       True
flipper_length_mm     True
body_mass_g           True
sex                   True
dtype: bool
```

```
df.isnull().sum()
```

```
species               0
island                0
culmen_length_mm      2
culmen_depth_mm       2
flipper_length_mm     2
body_mass_g           2
sex                  10
dtype: int64
```

```
df['culmen_length_mm'].fillna(df['culmen_length_mm'].median(),inplace=True)
df['culmen_depth_mm'].fillna(df['culmen_depth_mm'].median(),inplace=True)
df['flipper_length_mm'].fillna(df['flipper_length_mm'].median(),inplace=True)
```

```
df['body_mass_g'].fillna(df['body_mass_g'].median(),inplace=True)
df['sex'].fillna(df['sex'].mode()[0],inplace=True)
```

```
df.isnull().sum()
```

```
species              0
island               0
culmen_length_mm     0
culmen_depth_mm      0
flipper_length_mm    0
body_mass_g          0
sex                  0
dtype: int64
```

```
df.head()
```

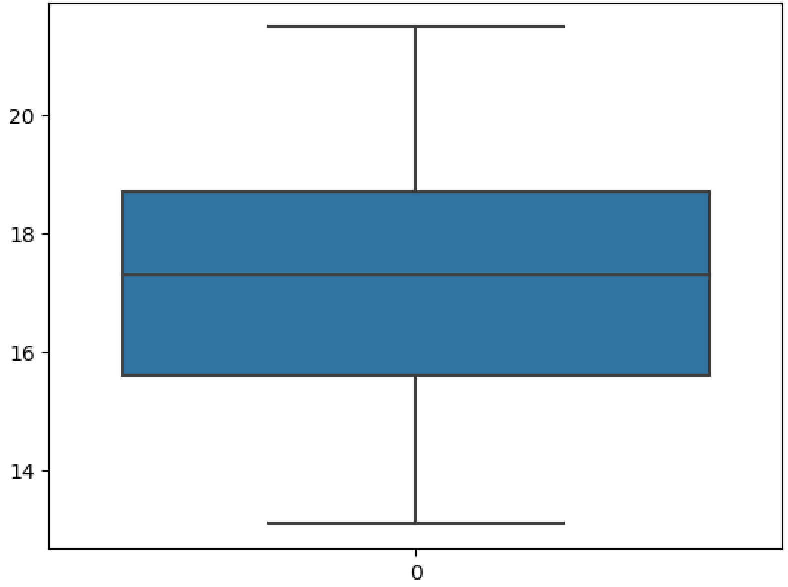
	species	island	culmen_length_mm	culmen_depth_mm	flipper_length_mm	body_mass_g	sex
0	Adelie	Torgersen	39.10	18.7	181.0	3750.0	MALE
1	Adelie	Torgersen	39.50	17.4	186.0	3800.0	FEMALE
2	Adelie	Torgersen	40.30	18.0	195.0	3250.0	FEMALE
3	Adelie	Torgersen	44.45	17.3	197.0	4050.0	MALE
4	Adelie	Torgersen	36.70	19.3	193.0	3450.0	FEMALE

```
sns.boxplot(df['culmen_length_mm'])
```

```
<Axes: >
```

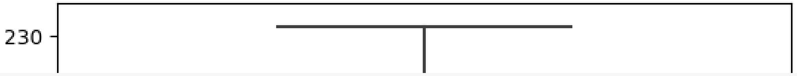
```
sns.boxplot(df['culmen_depth_mm'])
```

```
<Axes: >
```



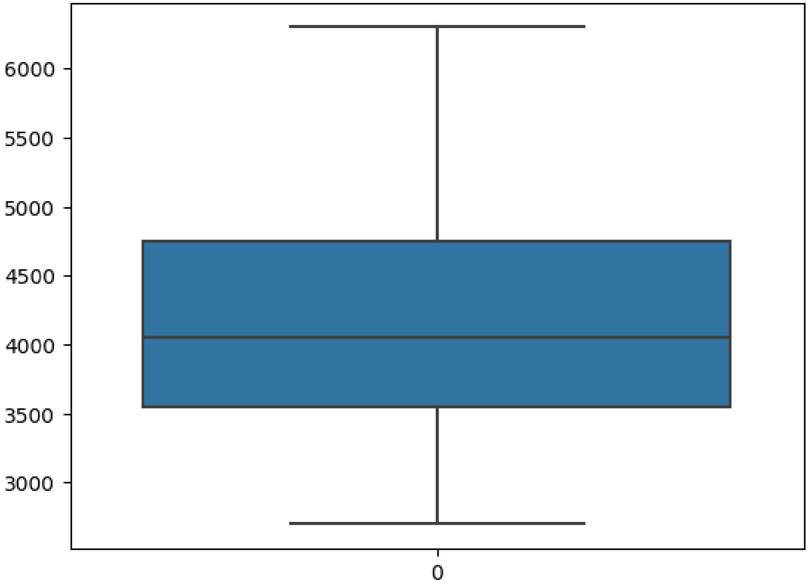
```
sns.boxplot(df['flipper_length_mm'])
```

```
<Axes: >
```



```
sns.boxplot(df['body_mass_g'])
```

```
<Axes: >
```



7 checking correlation

```
df.corr()
```

```
<ipython-input-44-2f6f6606aa2c>:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select only
```

8 check for categrical coloumns and perform encoding

```
flipper_length_mm          0.655858          0.583832          1.000000          0.871221
```

```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
```

```
df.sex=le.fit_transform(df.sex)
df.island=le.fit_transform(df.island)
df.species=le.fit_transform(df.species)
```

```
df.head()
```

	species	island	culmen_length_mm	culmen_depth_mm	flipper_length_mm	body_mass_g	sex
0	0	2	39.10	18.7	181.0	3750.0	2
1	0	2	39.50	17.4	186.0	3800.0	1
2	0	2	40.30	18.0	195.0	3250.0	1
3	0	2	44.45	17.3	197.0	4050.0	2
4	0	2	36.70	19.3	193.0	3450.0	1

finding correlation between target column and all other columns after encoding the target column into numerical column

```
df.corr().species.sort_values(ascending=False)
```

```
species              1.000000
flipper_length_mm    0.850819
body_mass_g          0.747547
culmen_length_mm     0.728706
sex                 -0.003823
island              -0.635659
culmen_depth_mm     -0.741282
Name: species, dtype: float64
```

split the data into dependent and independent variable

```
x=df.drop(columns=['species'],axis=1)
x.head()
```

	island	culmen_length_mm	culmen_depth_mm	flipper_length_mm	body_mass_g	sex
0	2	39.10	18.7	181.0	3750.0	2
1	2	39.50	17.4	186.0	3800.0	1
2	2	40.30	18.0	195.0	3250.0	1
3	2	44.45	17.3	197.0	4050.0	2
4	2	36.70	19.3	193.0	3450.0	1

```
y=df['species']
y
```

```
0      0
1      0
2      0
3      0
4      0
      ..
339    2
340    2
341    2
342    2
343    2
Name: species, Length: 344, dtype: int64
```

scaling a data

```
from sklearn.preprocessing import MinMaxScaler
scale=MinMaxScaler()
```

```
x_scaled= pd.DataFrame(scale.fit_transform(x),columns =x.columns)
x_scaled.head()
```

	island	culmen_length_mm	culmen_depth_mm	flipper_length_mm	body_mass_g	sex
0	1.0	0.254545	0.666667	0.152542	0.291667	1.0
1	1.0	0.269091	0.511905	0.237288	0.305556	0.5
2	1.0	0.298182	0.583333	0.389831	0.152778	0.5
3	1.0	0.449091	0.500000	0.423729	0.375000	1.0
4	1.0	0.167273	0.738095	0.355932	0.208333	0.5

split the data into training and testing

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x_scaled,y,test_size=0.3,random_state=0)
```

```
x_train
```

	island	culmen_length_mm	culmen_depth_mm	flipper_length_mm	body_mass_g	sex
219	0.5	0.658182	0.666667	0.440678	0.298611	0.5
271	0.0	0.596364	0.119048	0.813559	0.722222	1.0
266	0.0	0.487273	0.095238	0.644068	0.416667	0.5
335	0.0	0.836364	0.345238	0.983051	0.875000	1.0
217	0.5	0.636364	0.607143	0.355932	0.298611	1.0
...	...	...	...	...	...	...
323	0.0	0.618182	0.226190	0.949153	0.777778	1.0
192	0.5	0.614545	0.761905	0.644068	0.347222	1.0
117	1.0	0.189091	0.880952	0.457627	0.298611	1.0
47	0.5	0.196364	0.690476	0.118644	0.076389	1.0
172	0.5	0.374545	0.500000	0.152542	0.250000	0.5

240 rows × 6 columns

```
y_train
```

```
219    1
271    2
266    2
335    2
```

```
217    1
      ..
323    2
192    1
117    0
47     0
172    1
Name: species, Length: 240, dtype: int64
```

```
x_test
```

	island	culmen_length_mm	culmen_depth_mm	flipper_length_mm	body_mass_g	sex
141	0.5	0.309091	0.488095	0.254237	0.215278	1.0
6	1.0	0.247273	0.559524	0.152542	0.256944	0.5
60	0.0	0.130909	0.452381	0.220339	0.125000	0.5
249	0.0	0.650909	0.261905	0.813559	0.791667	1.0
54	0.0	0.087273	0.595238	0.254237	0.055556	0.5
...	...	...	...	...	...	...
81	1.0	0.392727	0.535714	0.406780	0.555556	1.0
1	1.0	0.269091	0.511905	0.237288	0.305556	0.5
120	1.0	0.149091	0.488095	0.254237	0.125000	0.5
8	1.0	0.072727	0.595238	0.355932	0.215278	1.0
313	0.0	0.632727	0.357143	0.881356	0.819444	1.0

104 rows × 6 columns

```
y_test
```

```
141    0
6      0
60     0
249    2
54     0
      ..
81     0
1      0
120    0
8      0
313    2
Name: species, Length: 104, dtype: int64
```

```
x_train.shape
```

```
    (240, 6)
```

```
x_test.shape
```

```
    (104, 6)
```

```
y_train.shape
```

```
    (240,)
```

```
y_test.shape
```

```
    (104,)
```

✓ 0s completed at 5:21 PM ● ✕