1. DOWNLOAD DATASET
2. LOAD DATASET

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

```
df=pd.read_csv('penguins_size.csv')
```

```
df.head(15)
```

	species	island	culmen_length_mm	culmen_depth_mm	flipper_length_mm	body_mass_g	sex
0	0	Torgersen	39.10	18.7	181.0	3750.0	MALE
1	0	Torgersen	39.50	17.4	186.0	3800.0	FEMALE
2	0	Torgersen	40.30	18.0	195.0	3250.0	FEMALE
3	0	Torgersen	44.45	17.3	197.0	4050.0	MALE
4	0	Torgersen	36.70	19.3	193.0	3450.0	FEMALE
5	0	Torgersen	39.30	20.6	190.0	3650.0	MALE
6	0	Torgersen	38.90	17.8	181.0	3625.0	FEMALE
7	0	Torgersen	39.20	19.6	195.0	4675.0	MALE
8	0	Torgersen	34.10	18.1	193.0	3475.0	MALE
9	0	Torgersen	42.00	20.2	190.0	4250.0	MALE
10	0	Torgersen	37.80	17.1	186.0	3300.0	MALE
11	0	Torgersen	37.80	17.3	180.0	3700.0	MALE
12	0	Torgersen	41.10	17.6	182.0	3200.0	FEMALE
13	0	Torgersen	38.60	21.2	191.0	3800.0	MALE
14	0	Torgersen	34.60	21.1	198.0	4400.0	MALE

```
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
```

4. Perform descriptive statistics on the dataset.

```
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

5. Check for Missing values and deal with them

```
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.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
```

check for skewness using distplot

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

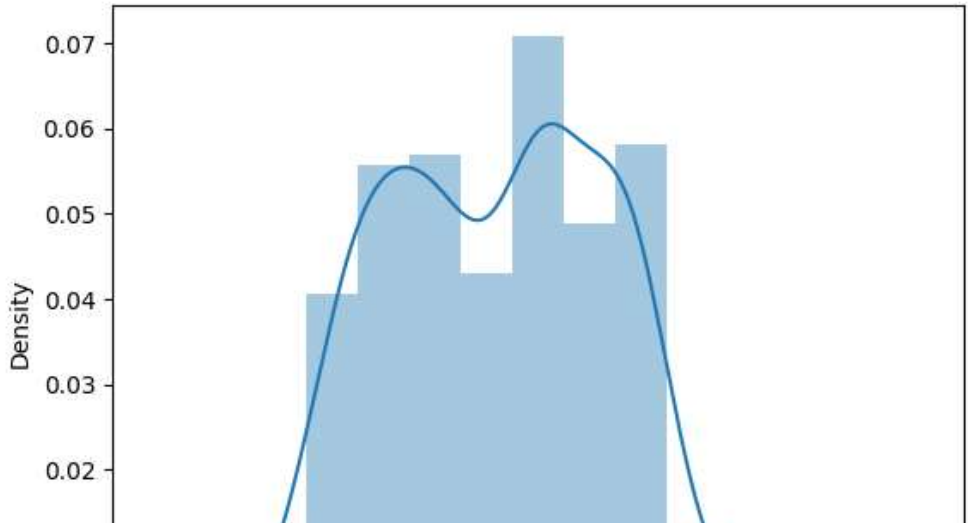
```
<ipython-input-13-87f900721a46>:1: UserWarning:

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

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df['culmen_length_mm'])
<Axes: xlabel='culmen_length_mm', ylabel='Density'>
```




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

```
<ipython-input-15-9161f519b2fb>:1: UserWarning:

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

Please adapt your code to use either `displot` (a figure-level function with
```

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

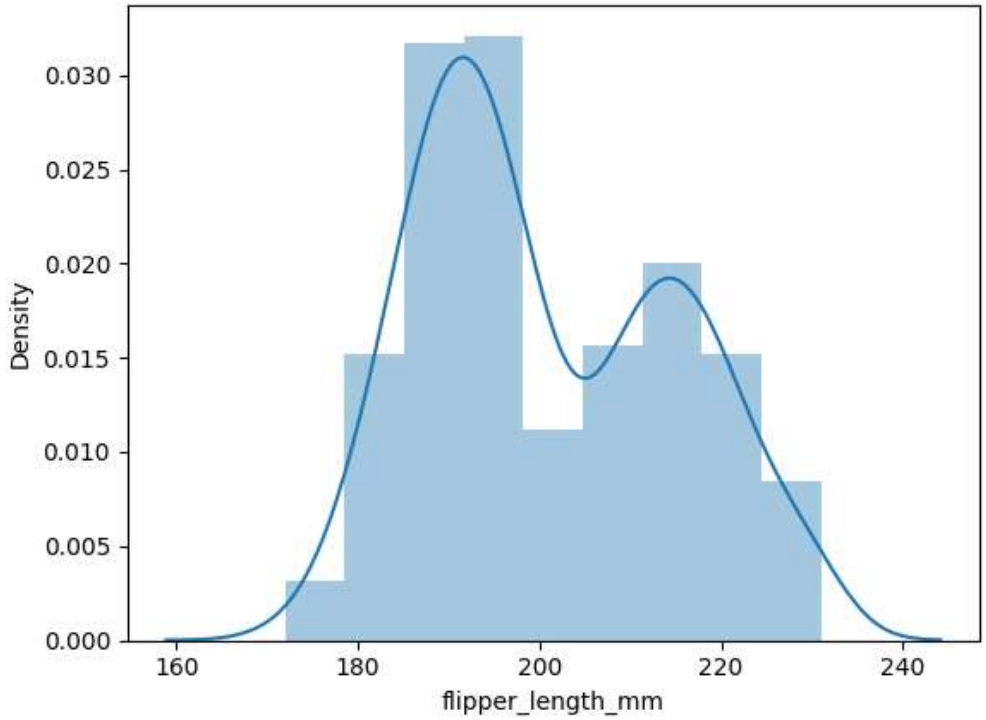
```
<ipython-input-18-25d29e01b18c>:1: UserWarning:

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

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df['flipper_length_mm'])
<Axes: xlabel='flipper_length_mm', ylabel='Density'>
```



Since its kind of right skewed and not proper bell shape we use median

```
df['culmen_length_mm']=df['culmen_length_mm'].fillna(df['culmen_length_mm'].median())
df['culmen_depth_mm']=df['culmen_depth_mm'].fillna(df['culmen_depth_mm'].median())
df['flipper_length_mm']=df['flipper_length_mm'].fillna(df['flipper_length_mm'].median())
df['body_mass_g']=df['body_mass_g'].fillna(df['body_mass_g'].median())
```

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

```
species              0
island               0
culmen_length_mm     0
culmen_depth_mm      0
flipper_length_mm    0
body_mass_g          0
```

```
sex                 10
dtype: int64
```

```
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   344 non-null    float64
 3   culmen_depth_mm    344 non-null    float64
 4   flipper_length_mm  344 non-null    float64
 5   body_mass_g        344 non-null    float64
 6   sex                334 non-null    object 
dtypes: float64(4), object(3)
memory usage: 18.9+ KB
```

```
df['sex']=df['sex'].fillna(df['sex'].mode()[0])
df['sex']
```

```
0       MALE
1     FEMALE
2     FEMALE
3       MALE
4     FEMALE
       ...
339     MALE
340   FEMALE
341     MALE
342   FEMALE
343     MALE
Name: sex, Length: 344, dtype: object
```

```
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
```

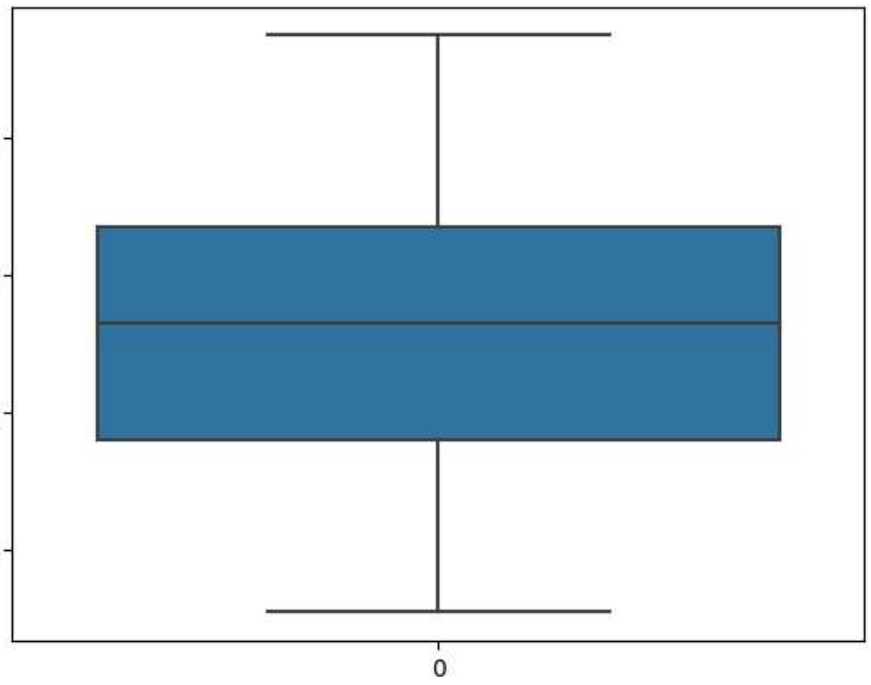
ALL NULL VALUES HAVE BEEN FIXED

3. Perform Below Visualizations: ● Univariate Analysis ● Bi- Variate Analysis ● Multi-Variate Analysis

Univariate Analysis

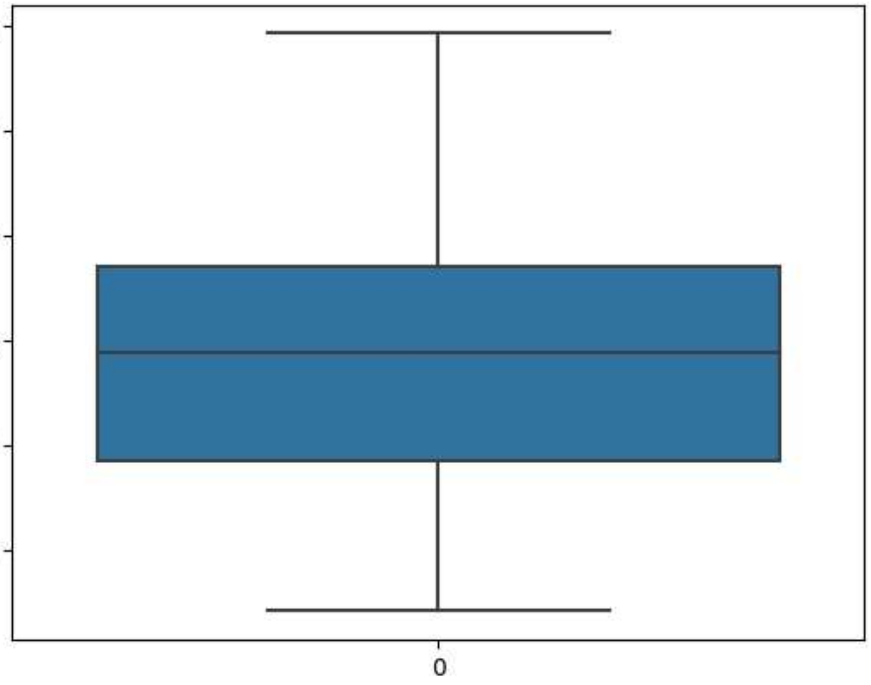
```
sns.boxplot(df['culmen_depth_mm'])
```

```
<Axes: >
```



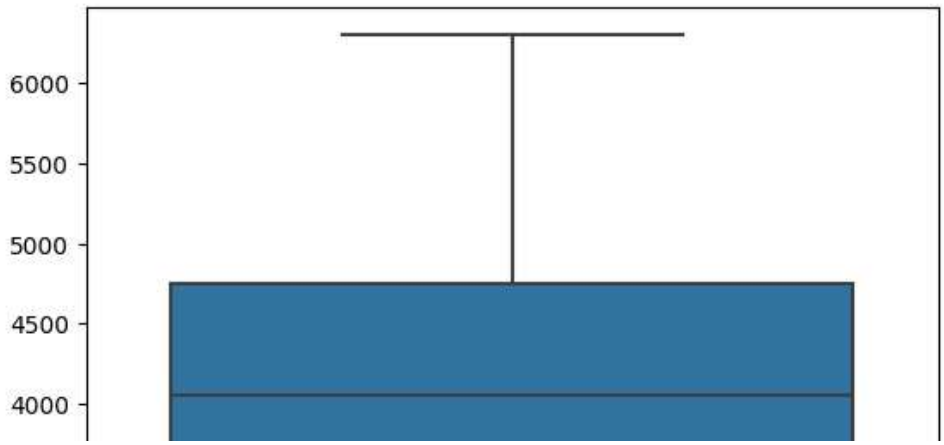

```
sns.boxplot(df['culmen_length_mm'])
```

```
<Axes: >
```



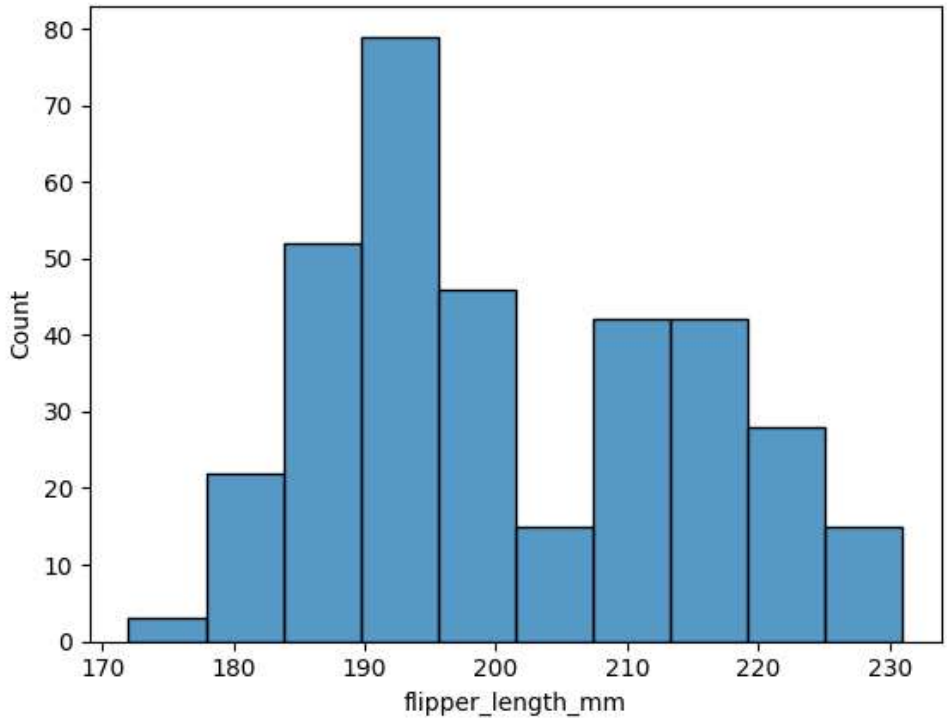

```
sns.boxplot(df['body_mass_g'])
```

```
<Axes: >
```




```
sns.histplot(data=df['flipper_length_mm'])
```

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

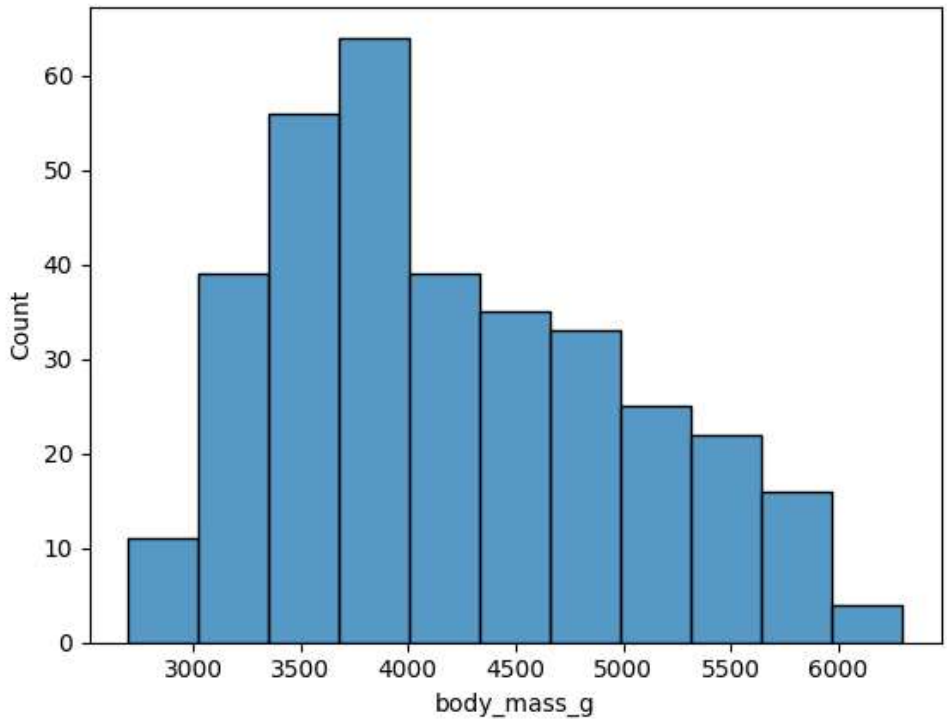



```
df.info()
```

```
<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   344 non-null    float64
 3   culmen_depth_mm    344 non-null    float64
 4   flipper_length_mm  344 non-null    float64
 5   body_mass_g        344 non-null    float64
 6   sex                344 non-null    object 
dtypes: float64(4), object(3)
memory usage: 18.9+ KB
```

```
sns.histplot(data=df['body_mass_g'])
```

```
<Axes: xlabel='body_mass_g', ylabel='Count'>
```



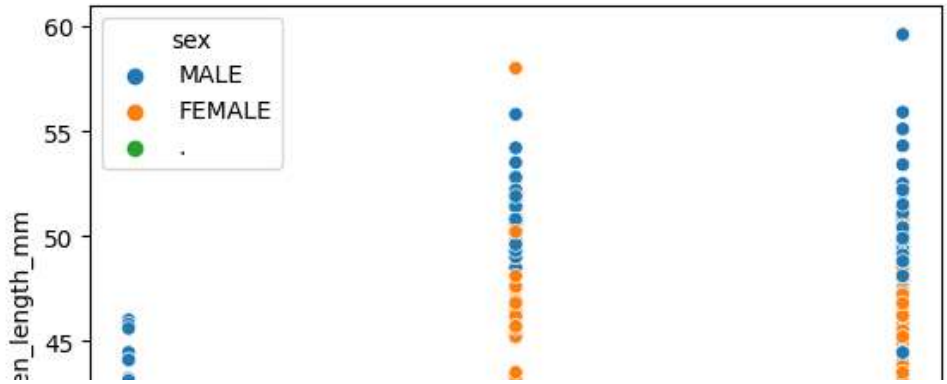
Bivariate Analysis

```
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   344 non-null    float64
 3   culmen_depth_mm    344 non-null    float64
 4   flipper_length_mm  344 non-null    float64
 5   body_mass_g        344 non-null    float64
 6   sex                344 non-null    object 
dtypes: float64(4), object(3)
memory usage: 18.9+ KB
```

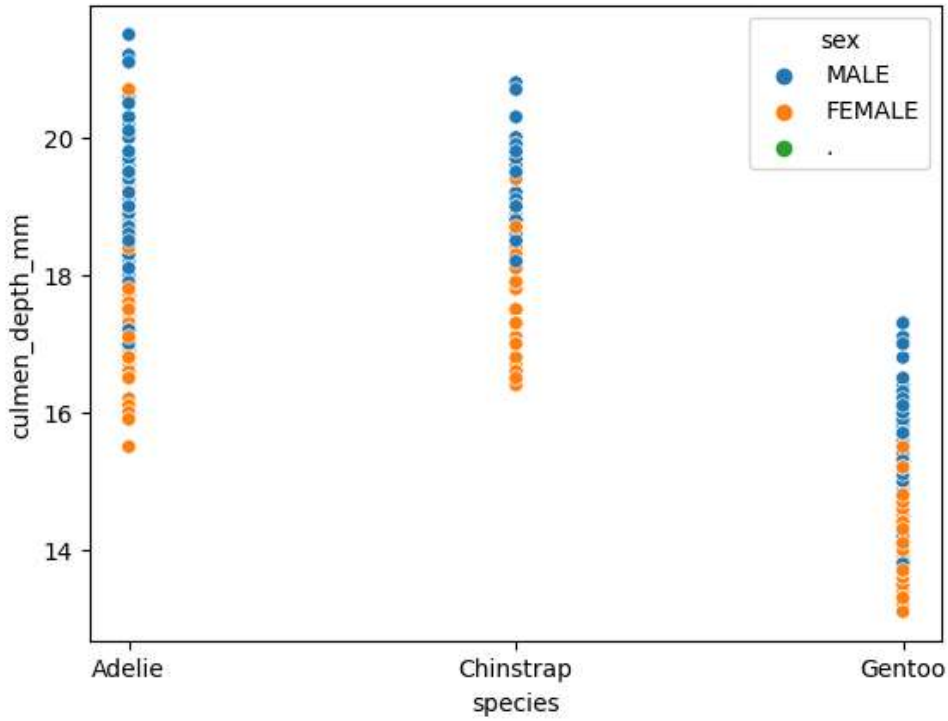
```
sns.scatterplot(x='species',y='culmen_length_mm',hue='sex',data=df)
```

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



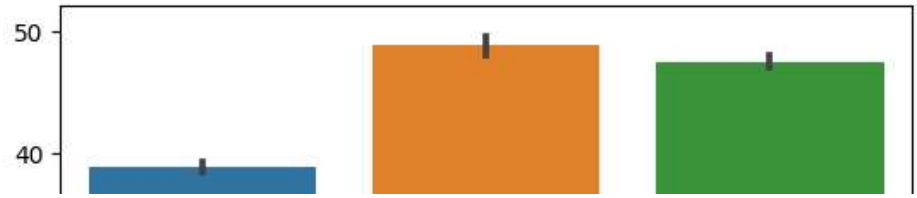

```
sns.scatterplot(x='species',y='culmen_depth_mm',hue='sex',data=df)
```

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



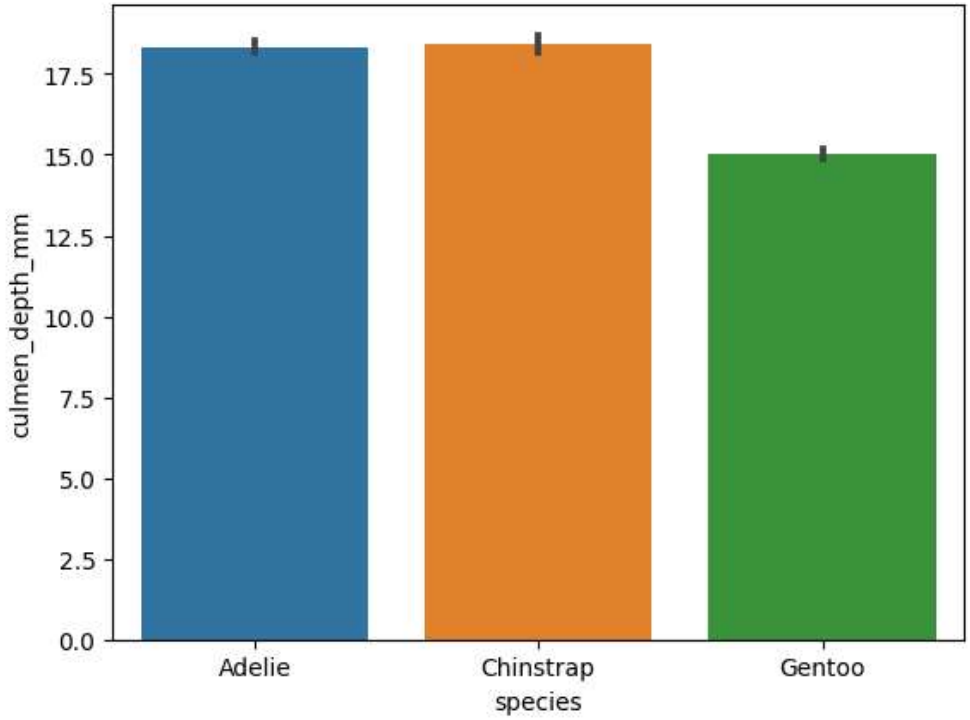

```
sns.barplot(x=df['species'],y=df['culmen_length_mm'])
```

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




```
sns.barplot(x=df['species'],y=df['culmen_depth_mm'])
```

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

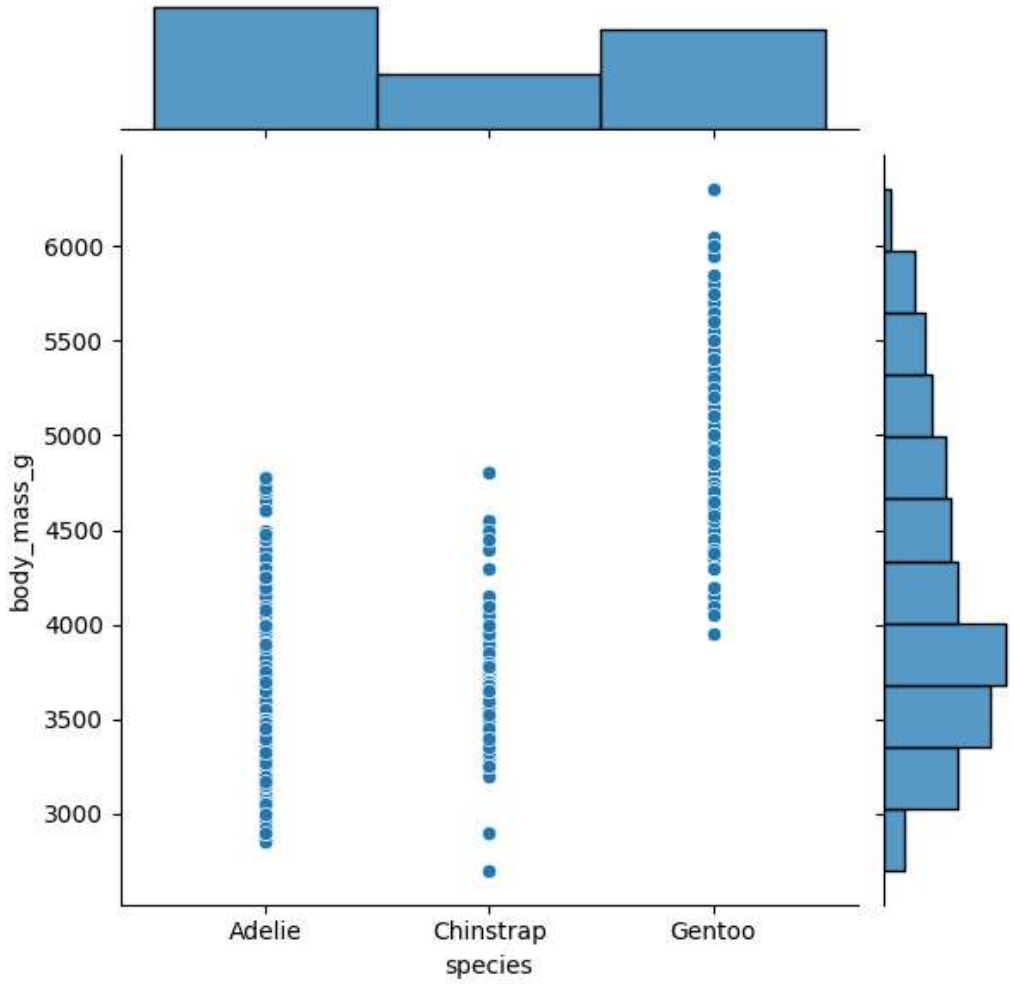



```
sns.barplot(x=df['species'],y=df['flipper_length_mm'])
```

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

```
sns.jointplot(x='species',y='body_mass_g',data=df)
```

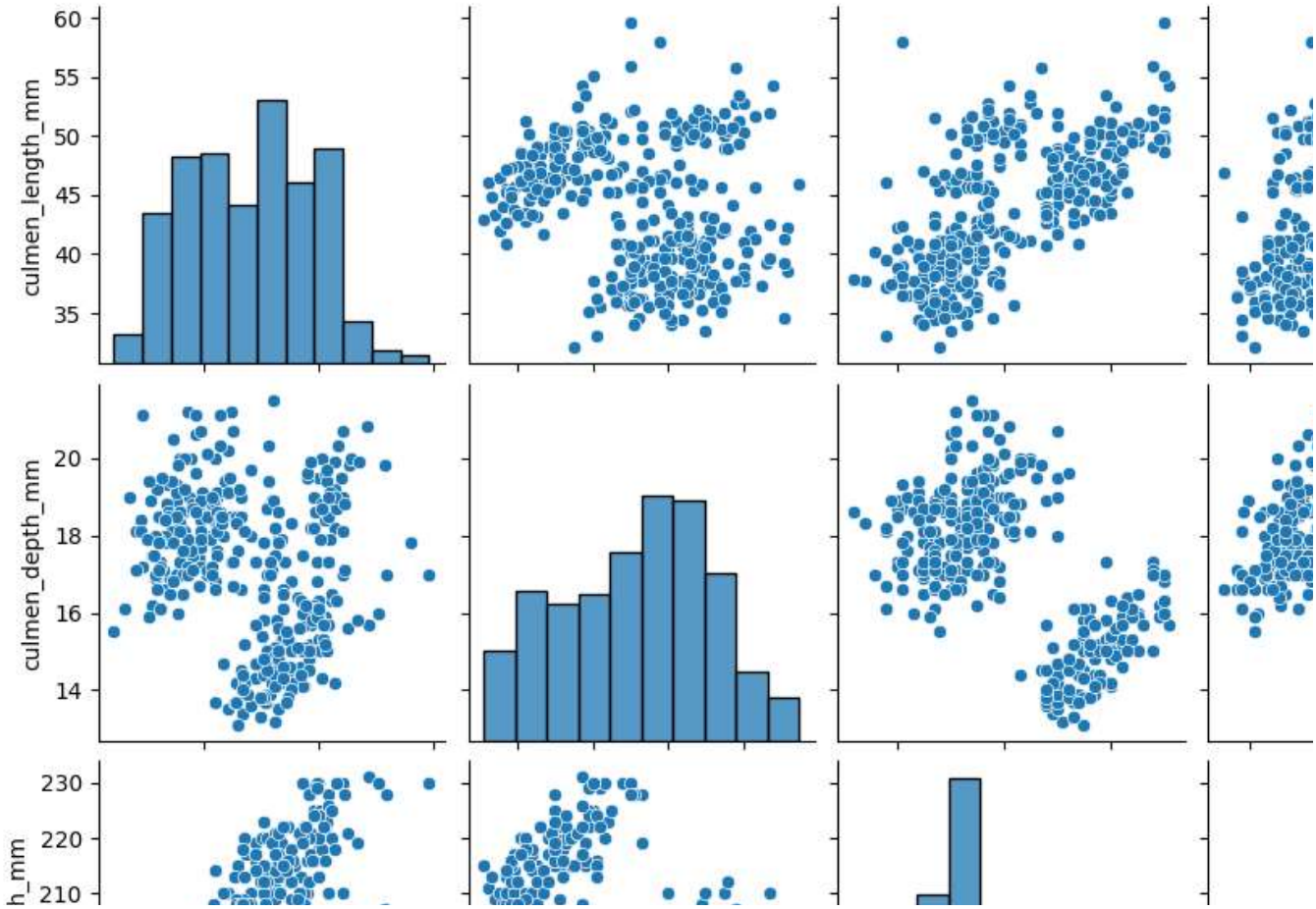
```
<seaborn.axisgrid.JointGrid at 0x79076d2cd3f0>
```



Multivariate Anlysis

```
sns.pairplot(data=df)
```

```
<seaborn.axisgrid.PairGrid at 0x79077018f910>
```



7.Check the correlation of independent variables with the target

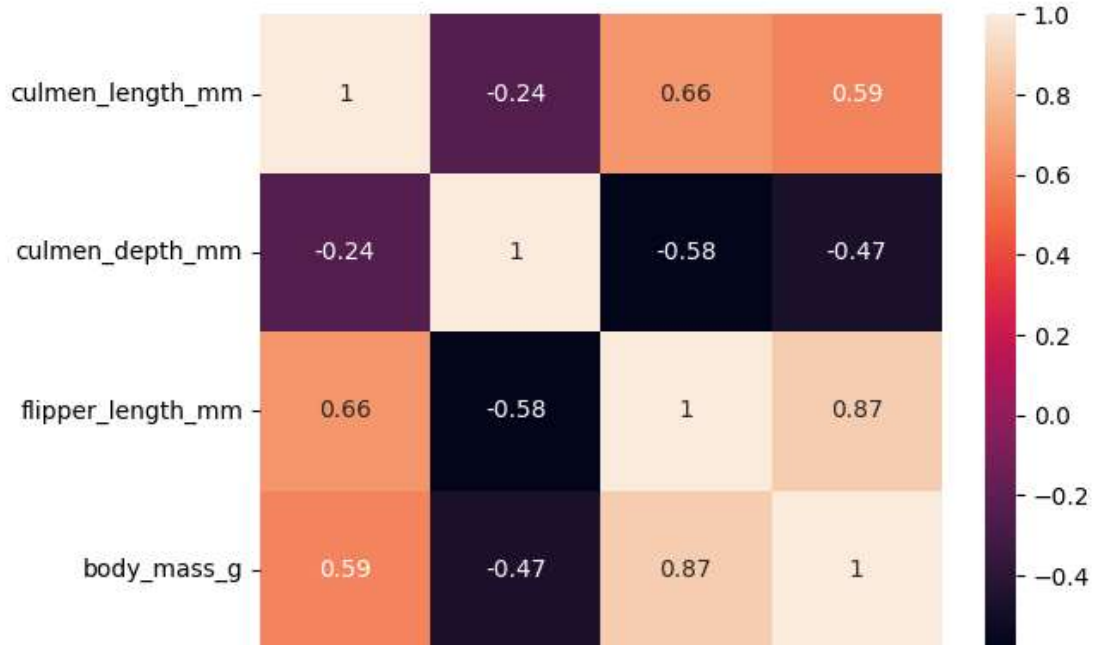
```
df.corr()
```

```
<ipython-input-39-2f6f6606aa2c>:1: FutureWarning: The default value of numeric_only in DataFrame.corr i
  df.corr()
```

	culmen_length_mm	culmen_depth_mm	flipper_length_mm	body_mass_g
culmen_length_mm	1.000000	-0.235000	0.655858	0.594925
culmen_depth_mm	-0.235000	1.000000	-0.583832	-0.471942
flipper_length_mm	0.655858	-0.583832	1.000000	0.871221
body_mass_g	0.594925	-0.471942	0.871221	1.000000

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

```
<ipython-input-41-aa355c3d392f>:1: FutureWarning: The default value of numeric_only in DataFrame.corr i
  sns.heatmap(data=df.corr(),annot=True)
<Axes: >
```



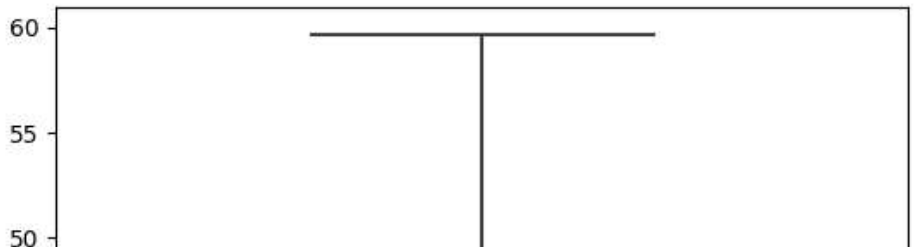
6. Find the outliers and replace them outliers

```
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   344 non-null    float64
 3   culmen_depth_mm    344 non-null    float64
 4   flipper_length_mm  344 non-null    float64
 5   body_mass_g        344 non-null    float64
 6   sex                344 non-null    object 
dtypes: float64(4), object(3)
memory usage: 18.9+ KB
```

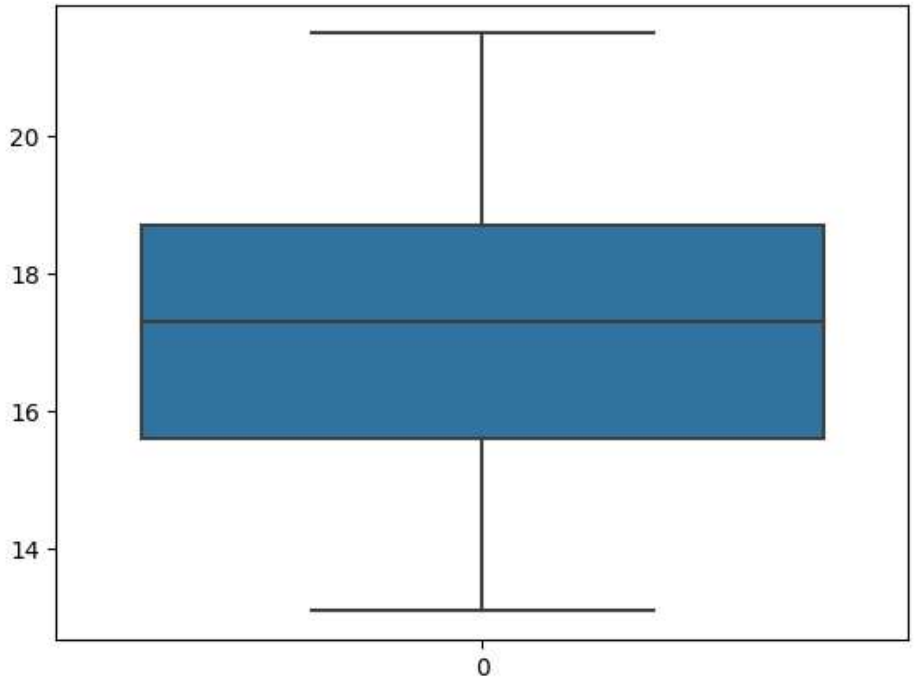
```
sns.boxplot(data=df['culmen_length_mm'])
```

```
<Axes: >
```




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

```
<Axes: >
```

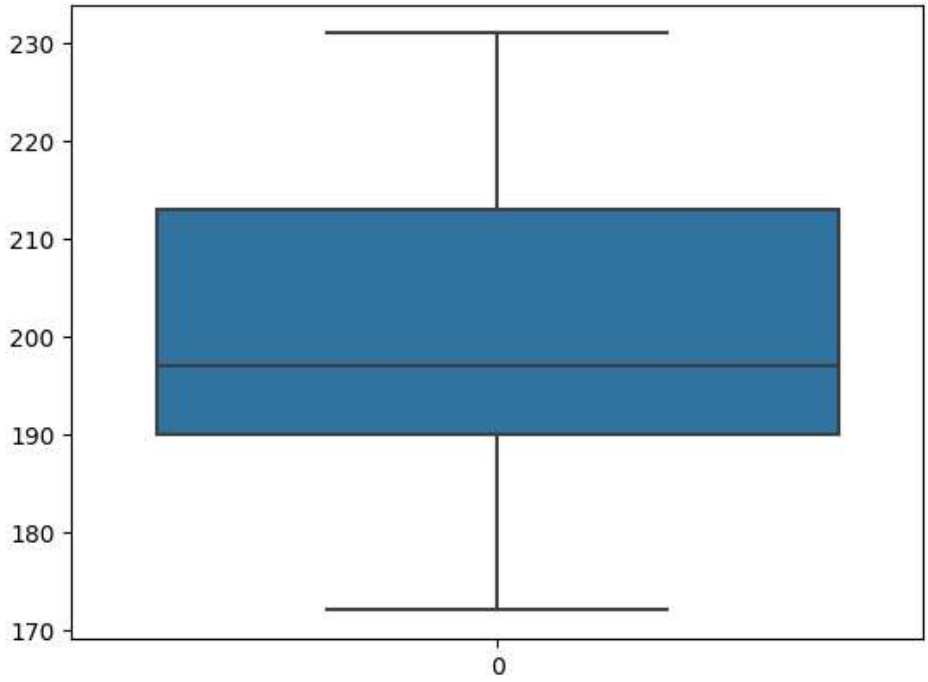



```
sns.boxplot(data=df['body_mass_g'])
```

```
<Axes: >
```

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

```
<Axes: >
```



NO OUTLIERS PRESENT

8. Check for Categorical columns and perform encoding.

```
# Import label encoder
from sklearn import preprocessing

# label_encoder object knows
# how to understand word labels.
label_encoder = preprocessing.LabelEncoder()

# Encode labels in column 'species'.
df['species']= label_encoder.fit_transform(df['species'])
df.tail(10)
```

	species	island	culmen_length_mm	culmen_depth_mm	flipper_length_mm	body_mass_g	sex
334	2	Biscoe	46.20	14.1	217.0	4375.0	FEMALE
335	2	Biscoe	55.10	16.0	230.0	5850.0	MALE

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

df['island']= label_encoder.fit_transform(df['island'])
df['island'].unique()
```

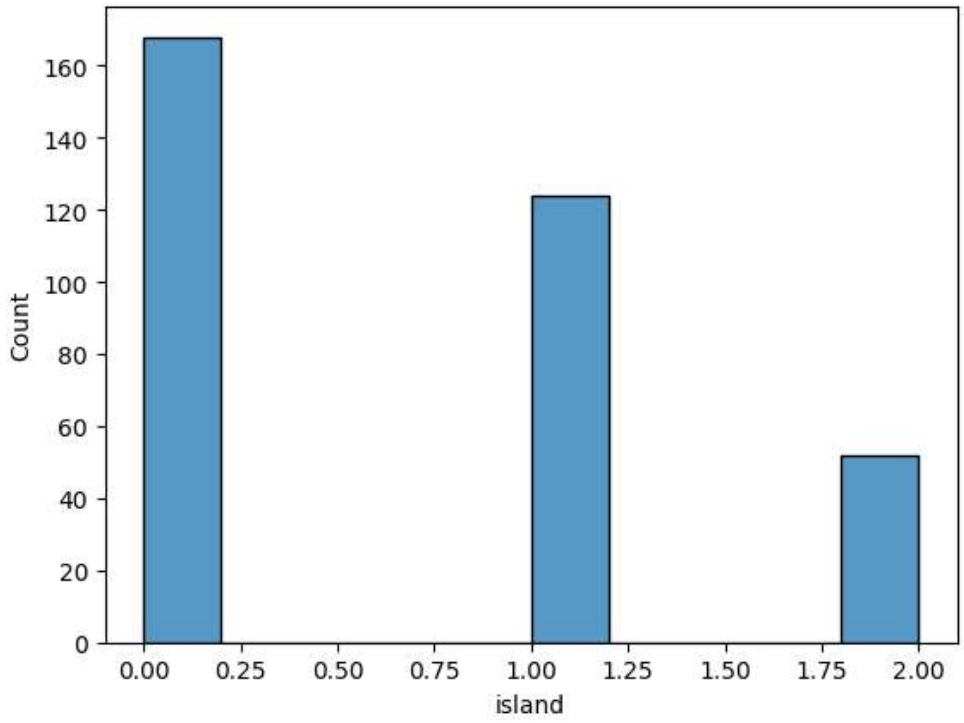
```
array([2, 0, 1])
```

```
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	MALE
1	0	2	39.50	17.4	186.0	3800.0	FEMALE
2	0	2	40.30	18.0	195.0	3250.0	FEMALE
3	0	2	44.45	17.3	197.0	4050.0	MALE
4	0	2	36.70	19.3	193.0	3450.0	FEMALE

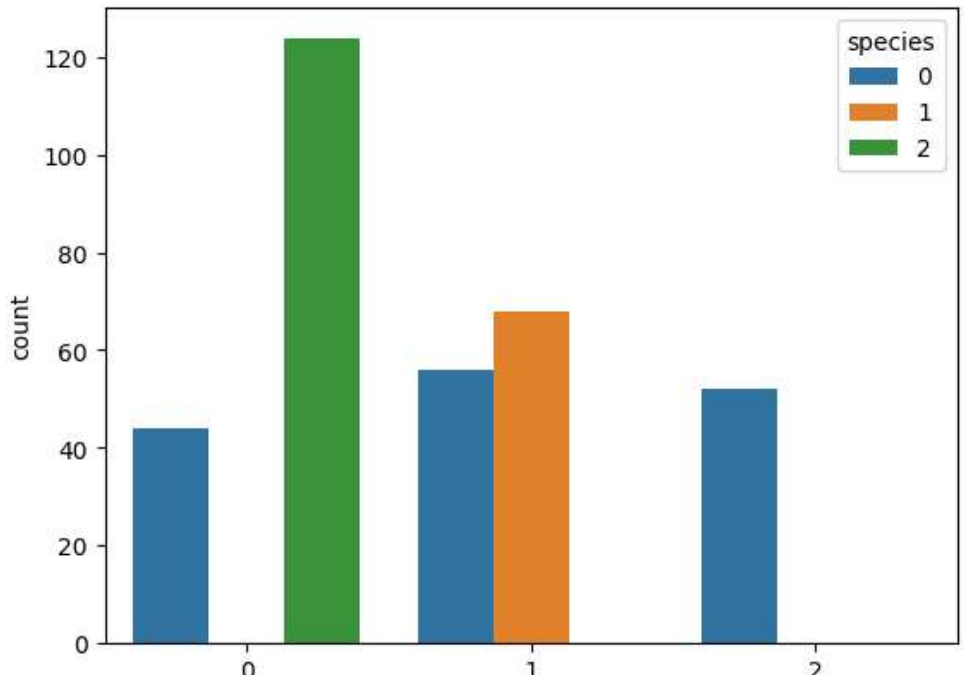
```
sns.histplot(data=df['island'])
```

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



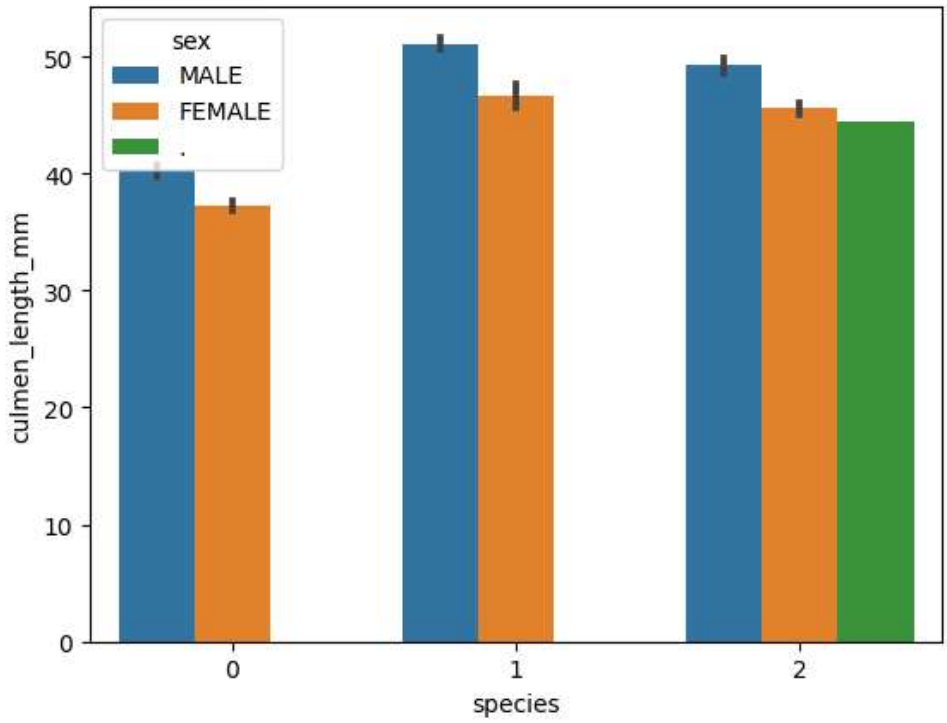

```
sns.countplot(x='island',hue='species',data=df)
```

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



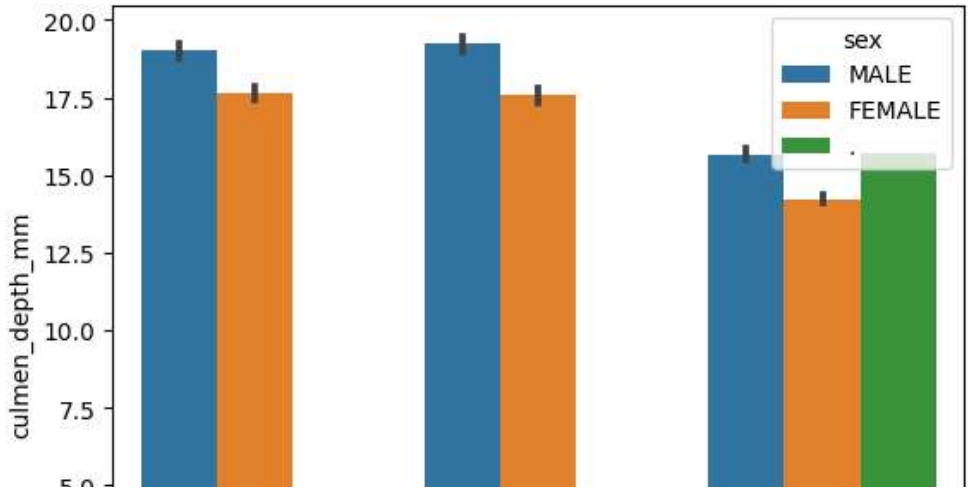

```
sns.barplot(x='species',y='culmen_length_mm',hue='sex',data=df)
```

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



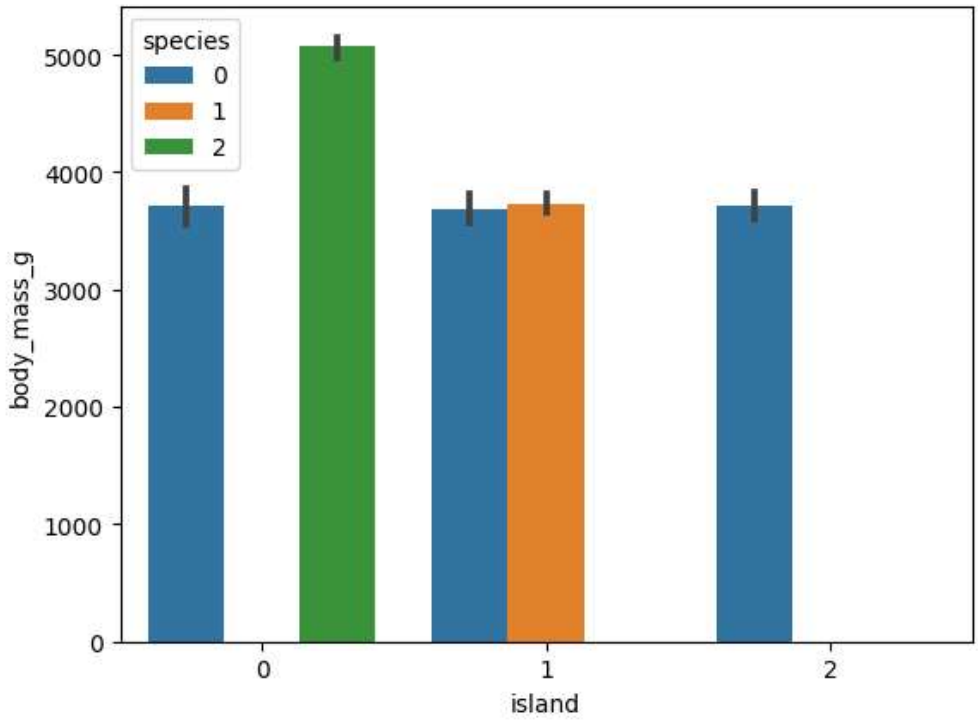

```
sns.barplot(x='species',y='culmen_depth_mm',hue='sex',data=df)
```

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



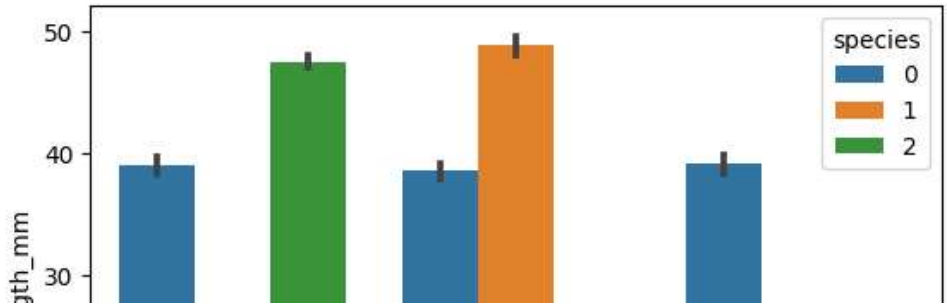

```
sns.barplot(x='island',y='body_mass_g',hue='species',data=df)
```

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



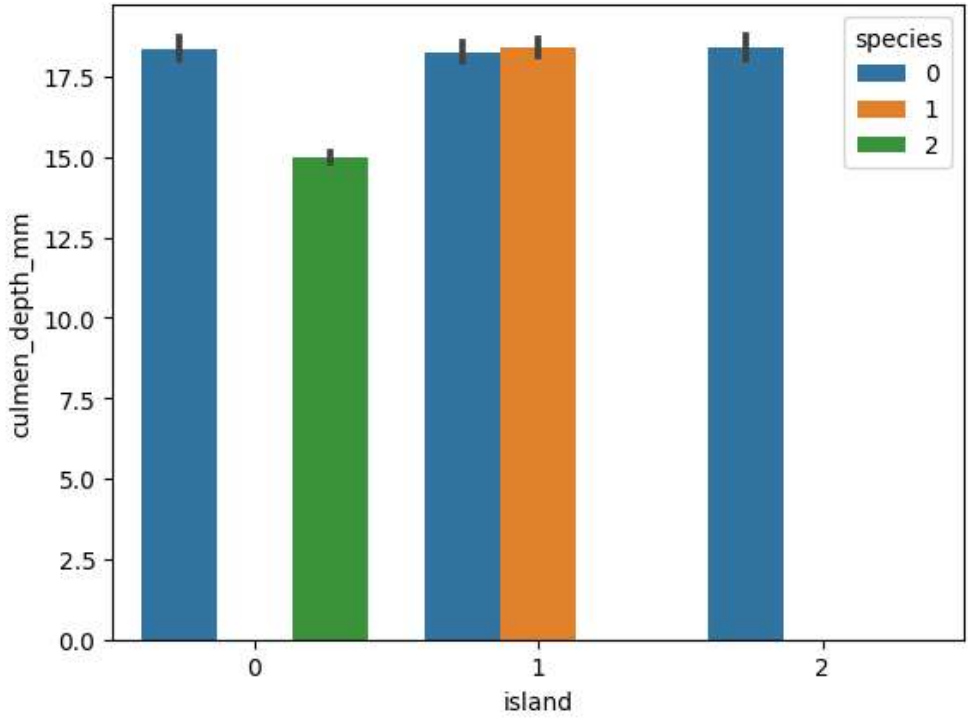

```
sns.barplot(x='island',y='culmen_length_mm',hue='species',data=df)
```

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




```
sns.barplot(x='island',y='culmen_depth_mm',hue='species',data=df)
```

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



9. Split the data into dependent and independent variables.

```
X=df.drop('species',axis=1)
y=df['species']
```

```
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.head()
```

```
0    0
1    0
2    0
3    0
4    0
Name: species, dtype: int64
```

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

df['sex']= label_encoder.fit_transform(df['sex'])
df['sex']
```

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

10. Scaling the data
11. Split the data into training and testing
12. check the training and testing data shape

```
from sklearn.preprocessing import StandardScaler
sc=StandardScaler()
X_scaled= pd.DataFrame(sc.fit_transform(X),columns =X.columns)
X_scaled.head()
```

	island	culmen_length_mm	culmen_depth_mm	flipper_length_mm	body_mass_g	sex
0	1.844076	-0.887622	0.787289	-1.420541	-0.564625	0.960230
1	1.844076	-0.814037	0.126114	-1.063485	-0.502010	-1.017729
2	1.844076	-0.666866	0.431272	-0.420786	-1.190773	-1.017729
3	1.844076	0.096581	0.075255	-0.277964	-0.188936	0.960230
4	1.844076	-1.329133	1.092447	-0.563608	-0.940314	-1.017729

```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(X_scaled,y,test_size=0.2,random_state=42)
X_train.shape
```

```
(275, 6)
```

```
y_train.shape
```

```
(275,)
```

```
X_test.shape
```

```
(69, 6)
```

```
y_test.shape
```

```
(69,)
```

✓ 0s completed at 10:06 PM