G PAVAN KUMAR

21BCE9495

VIT-AP UNIVERSITY

```
import numpy as np
import pandas as pd
```

loading datset

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

```
df.head()
```

	species	island	culmen_length_mm	culmen_depth_mm	flipper_length_mm	body_mas
0	Adelie	Torgersen	39.1	18.7	181.0	375
1	Adelie	Torgersen	39.5	17.4	186.0	38(
2	Adelie	Torgersen	40.3	18.0	195.0	325
3	Adelie	Torgersen	NaN	NaN	NaN	N
4	Adelie	Torgersen	36.7	19.3	193.0	345

```
df.shape
```

```
    (344, 7)
```

```
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
```

Handling missing values

replace null values

```
df["culmen_length_mm"]=df["culmen_length_mm"].fillna(np.mean(df["culmen_length_mm"]))
```

```
df["culmen_length_mm"].isnull().sum()
```

```
    0
```

```
df["culmen_depth_mm"]=df["culmen_depth_mm"].fillna(np.mean(df["culmen_depth_mm"]))
df["culmen_depth_mm"].isnull().sum()
```

```
    0
```

```
df["flipper_length_mm"]=df["flipper_length_mm"].fillna(np.mean(df["flipper_length_mm"]))
df["flipper_length_mm"].isnull().sum()
```

```
    0
```

```
df["body_mass_g"]=df["body_mass_g"].fillna(np.mean(df["body_mass_g"]))
df["body_mass_g"].isnull().sum()
```

```
    0
```

```
df["sex"]=df["sex"].fillna(df["sex"].mode()[0])
df["sex"].isnull().sum()
```

```
    0
```

```
df.sex.value_counts()
```

```
MALE      178
FEMALE    165
.           1
Name: sex, dtype: int64
```

```
df['sex'] = df['sex'].str.replace('.', 'MALE')
```

```
<ipython-input-43-a5ac3d0c6a82>:1: FutureWarning: The default value of regex will change from True to False in a future version. Ir
  df['sex'] = df['sex'].str.replace('.', 'MALE')
```

```
df.sex.value_counts()
```

```
MALE      179
FEMALE    165
Name: sex, dtype: int64
```

```
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	39.10000	18.70000	181.000000	3750.000000	MALE
1	Adelie	Torgersen	39.50000	17.40000	186.000000	3800.000000	FEMALE
2	Adelie	Torgersen	40.30000	18.00000	195.000000	3250.000000	FEMALE
3	Adelie	Torgersen	43.92193	17.15117	200.915205	4201.754386	MALE
4	Adelie	Torgersen	36.70000	19.30000	193.000000	3450.000000	FEMALE

describing statistics

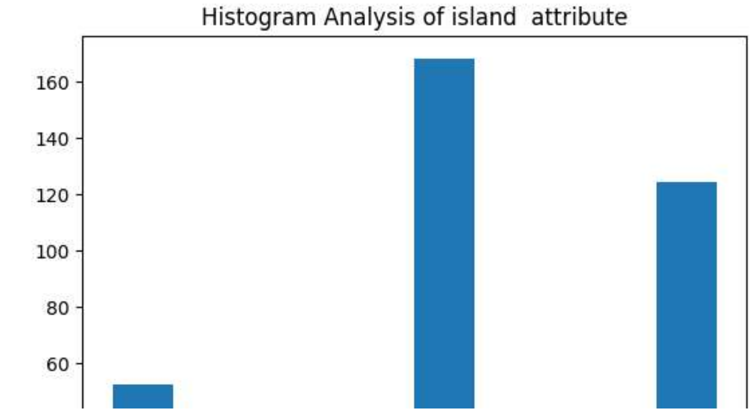
```
df.describe()
```

	culmen_length_mm	culmen_depth_mm	flipper_length_mm	body_mass_g
count	344.000000	344.000000	344.000000	344.000000
mean	43.921930	17.151170	200.915205	4201.754386
std	5.443643	1.969027	14.020657	799.613058
min	32.100000	13.100000	172.000000	2700.000000
25%	39.275000	15.600000	190.000000	3550.000000
50%	44.250000	17.300000	197.000000	4050.000000
75%	48.500000	18.700000	213.000000	4750.000000
max	59.600000	21.500000	231.000000	6300.000000

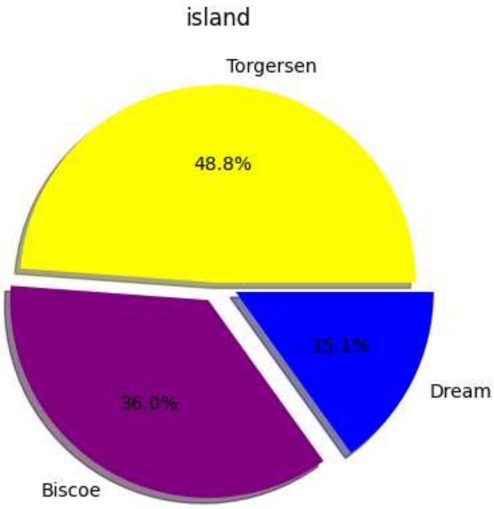
univariate analysis

```
import matplotlib.pyplot as plt
```

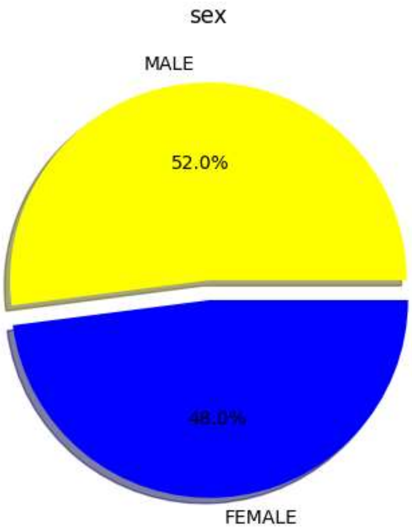
```
plt.title("Histogram Analysis of island  attribute")
plt.hist(df["island"])
plt.show()
```



```
plt.pie(df.island.value_counts(),[0,0.1,0.1],labels=['Torgersen','Biscoe','Dream'],autopct='%1.01f%%',shadow=True,colors=['yellow','purp
plt.title('island')
plt.show()
```



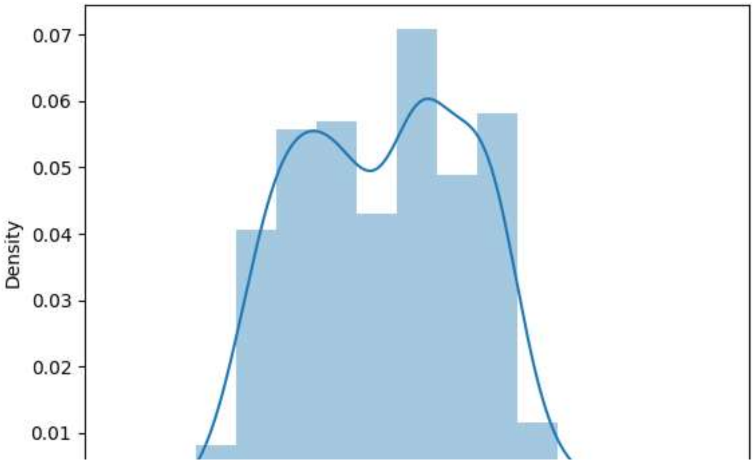
```
plt.pie(df.sex.value_counts(),[0,0.1],labels=['MALE','FEMALE'],autopct='%1.01f%%',shadow=True,colors=['yellow','blue'])
plt.title('sex')
plt.show()
```



```
import seaborn as sns
```

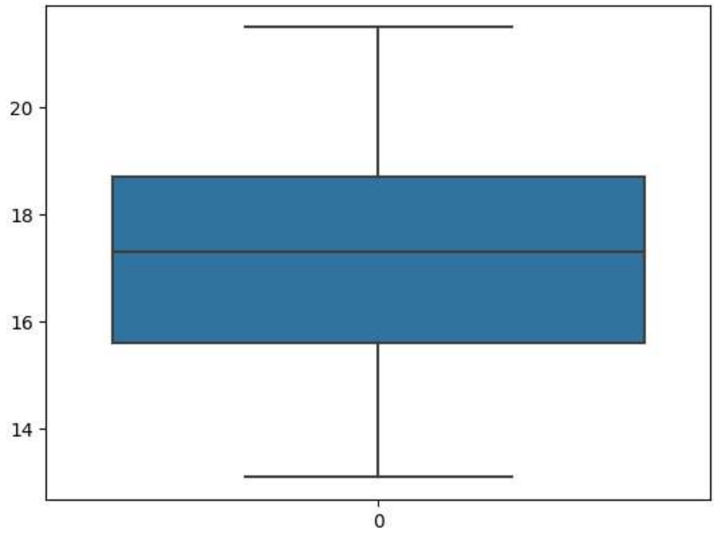
```
sns.distplot(df["culmen_length_mm"])
plt.show()
```

```
<ipython-input-53-9c96b21500a0>: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"])
```

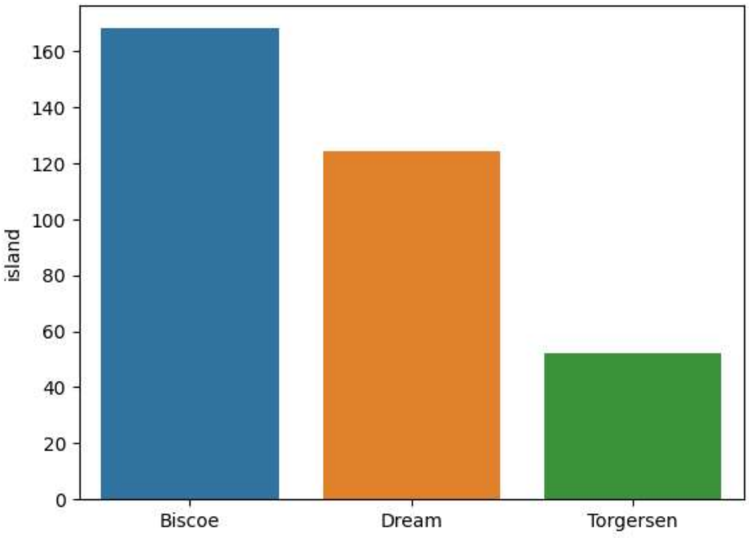


```
sns.boxplot(df.culmen_depth_mm)
```

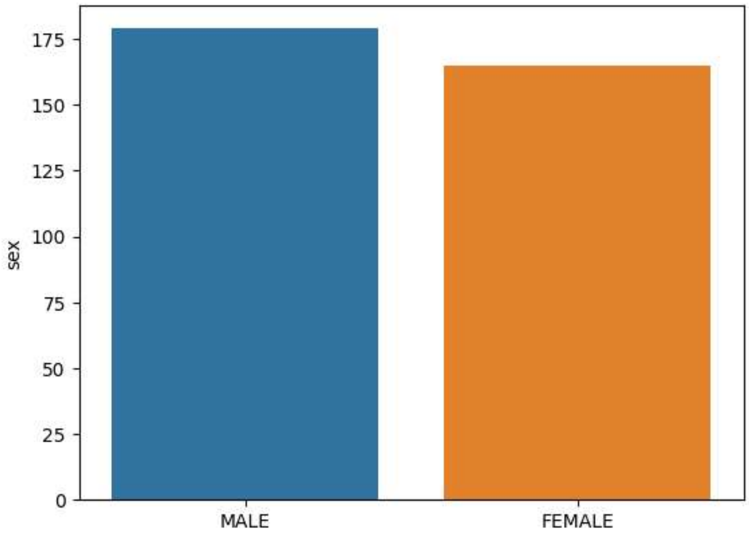
```
<Axes: >
```



```
sns.barplot(x=df.island.value_counts().index,y=df.island.value_counts())
plt.show()
```



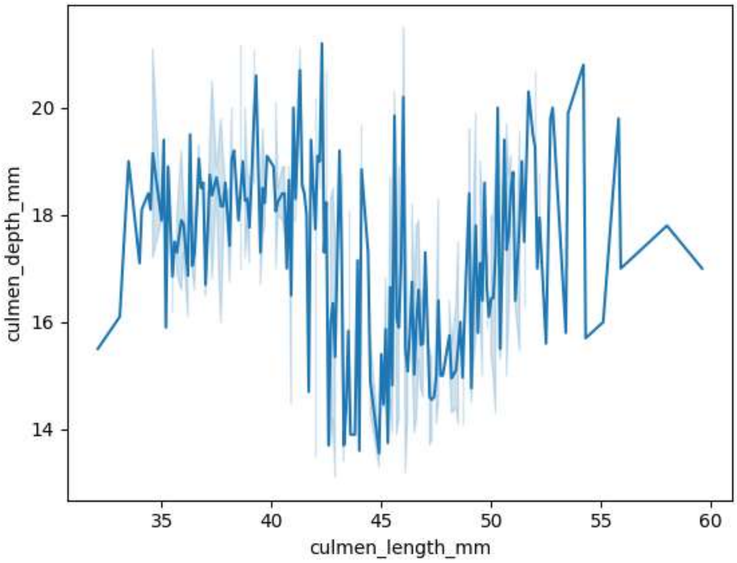
```
sns.barplot(x=df.sex.value_counts().index,y=df.sex.value_counts())
plt.show()
```



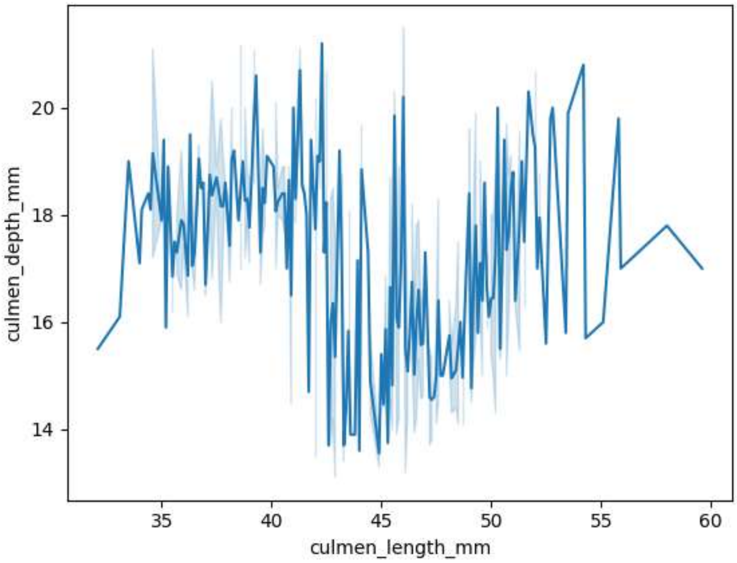
Bi variate analysis

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

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

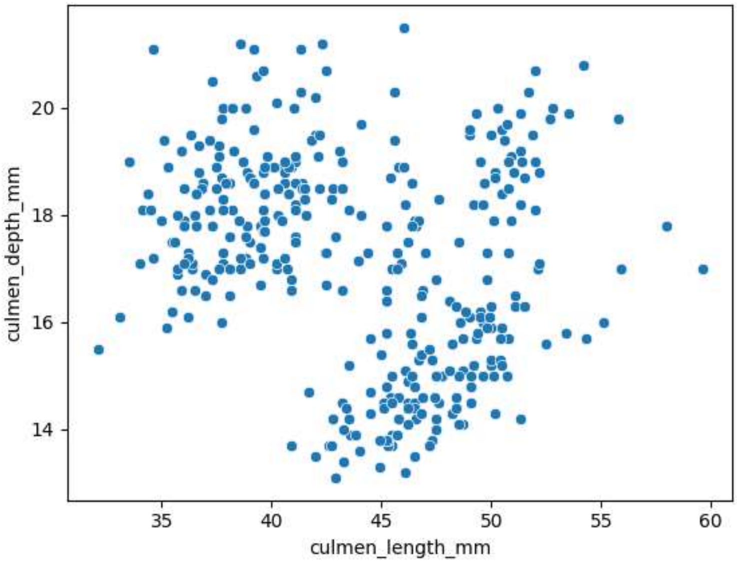


```
<Axes: xlabel='culmen_length_mm', ylabel='culmen_depth_mm'>
```

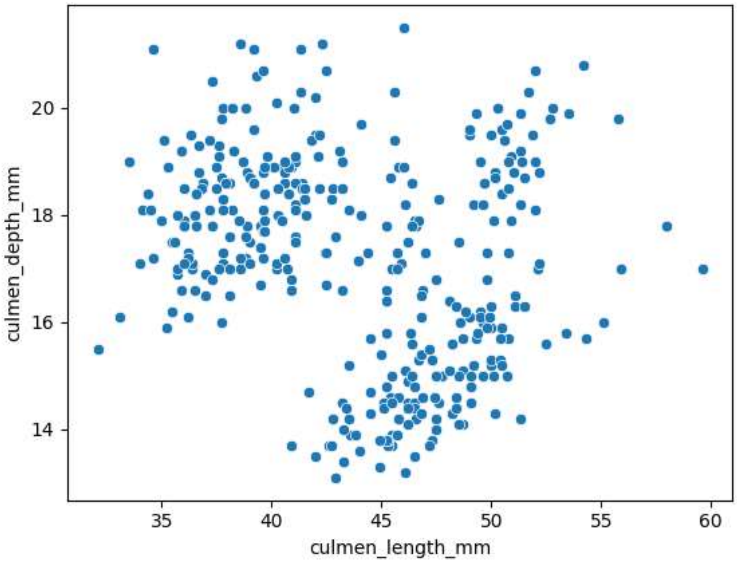


```
sns.scatterplot(x=df.culmen_length_mm,y=df.culmen_depth_mm)
```

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



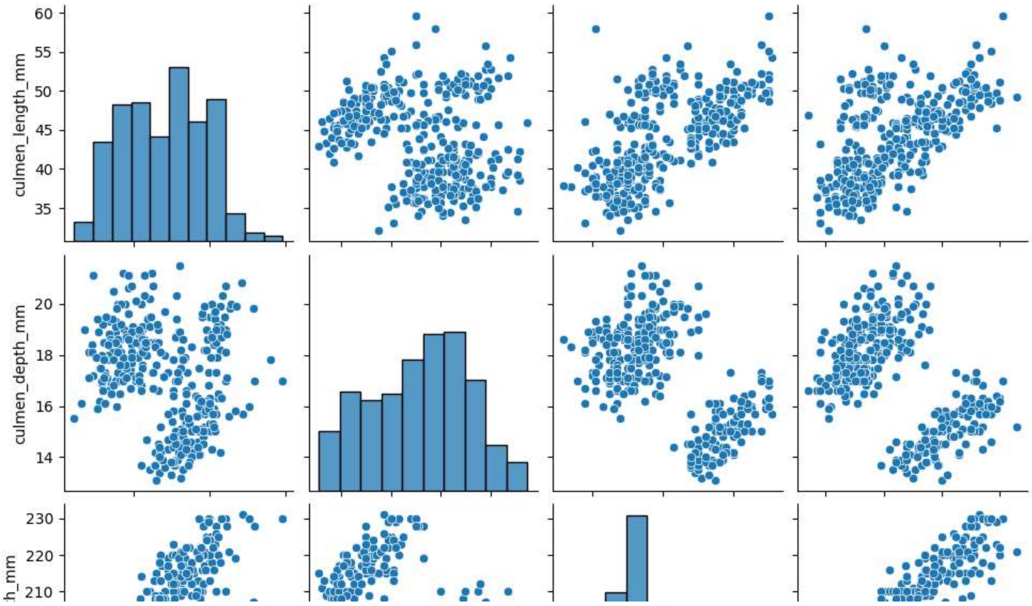
```
<Axes: xlabel='culmen_length_mm', ylabel='culmen_depth_mm'>
```



Multivariate analysis

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

```
<seaborn.axisgrid.PairGrid at 0x792f3167d8a0>
```



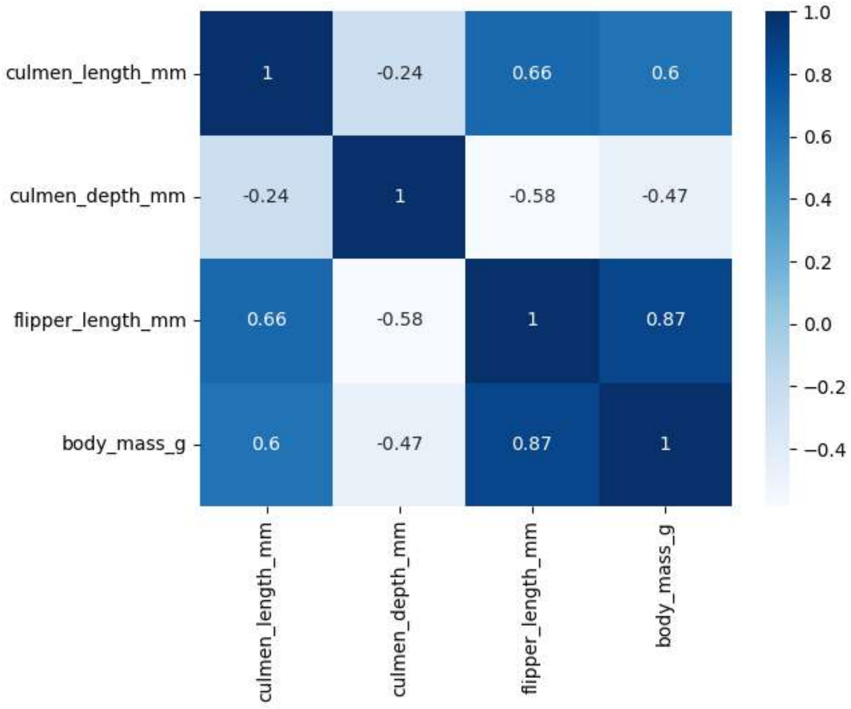
```
df.corr()
```

```
<ipython-input-59-2f6f6606aa2c>: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	-0.235053	0.656181	0.595110
culmen_depth_mm	-0.235053	1.000000	-0.583851	-0.471916
flipper_length_mm	0.656181	-0.583851	1.000000	0.871202
body_mass_g	0.595110	-0.471916	0.871202	1.000000

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

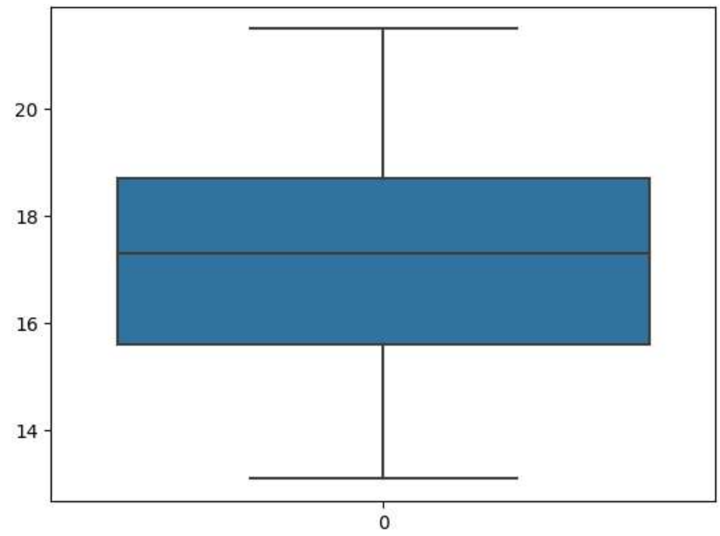
```
<ipython-input-60-86807cfe395e>:1: FutureWarning: The default value of numeric_only in DataFrame.corr i
  sns.heatmap(df.corr(),annot=True,cmap="Blues")
<Axes: >
```



outliers checking

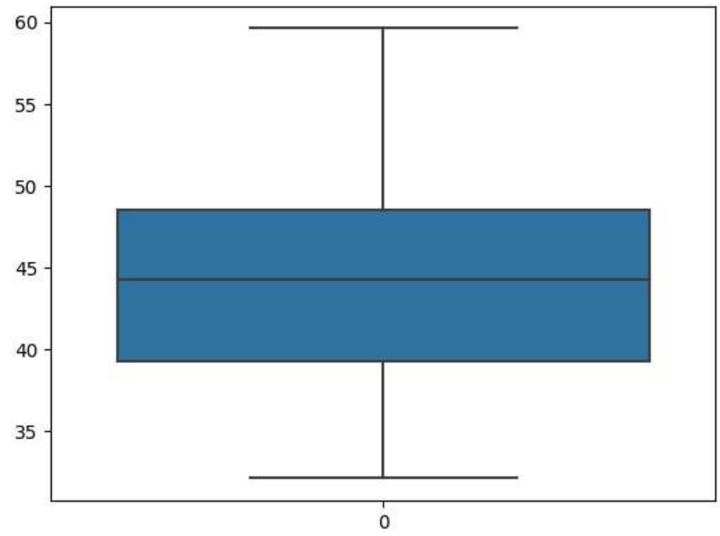
```
sns.boxplot(df.culmen_depth_mm)
```

```
<Axes: >
```



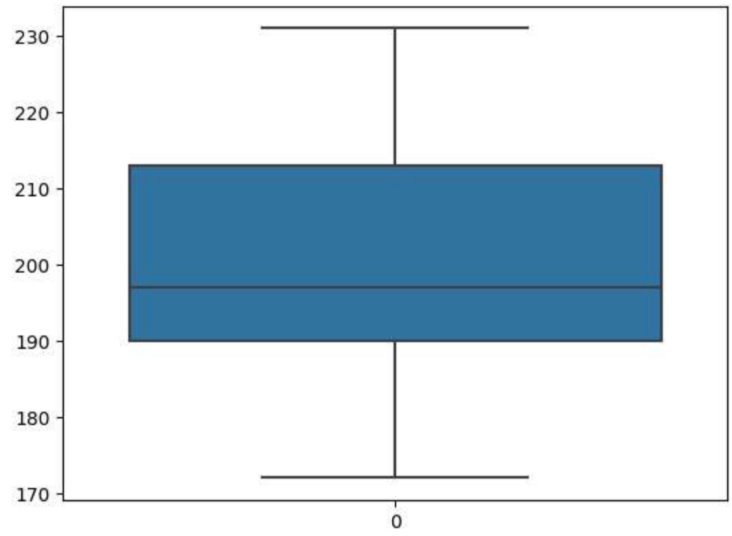
```
sns.boxplot(df.culmen_length_mm)
```

```
<Axes: >
```



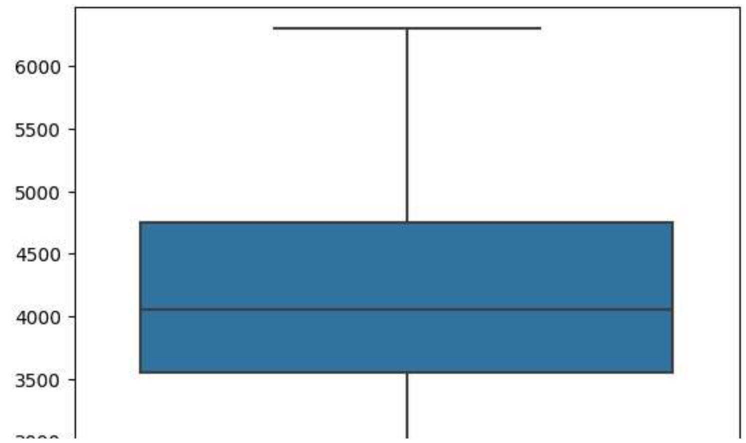
```
sns.boxplot(df. flipper_length_mm   )
```

```
<Axes: >
```



```
sns.boxplot(df.body_mass_g)
```

```
<Axes: >
```



Labelencoding

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

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

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

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

```
df.head()
```

	species	island	culmen_length_mm	culmen_depth_mm	flipper_length_mm	body_mass_g	sex
0	0	2	39.10000	18.70000	181.000000	3750.000000	1
1	0	2	39.50000	17.40000	186.000000	3800.000000	0
2	0	2	40.30000	18.00000	195.000000	3250.000000	0
3	0	2	43.92193	17.15117	200.915205	4201.754386	1
4	0	2	36.70000	19.30000	193.000000	3450.000000	0

correlation of independent variables

```
df.corr()
```

	species	island	culmen_length_mm	culmen_depth_mm	flipper_length_mm	body_mass
species	1.000000	-0.635659	0.728674	-0.741335	0.851160	0.7477
island	-0.635659	1.000000	-0.351461	0.567506	-0.562328	-0.5580
culmen_length_mm	0.728674	-0.351461	1.000000	-0.235053	0.656181	0.5951
culmen_depth_mm	-0.741335	0.567506	-0.235053	1.000000	-0.583851	-0.4719
flipper_length_mm	0.851160	-0.562328	0.656181	-0.583851	1.000000	0.8712
body_mass_g	0.747726	-0.558045	0.595110	-0.471916	0.871202	1.0000
sex	0.010240	0.002893	0.322338	0.354374	0.243556	0.4082

Split the data into dependent and independent variables

```
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	39.10000	18.70000	181.000000	3750.000000	1

```
y=df.species
y.head()
```

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

normalizing (scaling 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	0.0
2	1.0	0.298182	0.583333	0.389831	0.152778	0.0
3	1.0	0.429888	0.482282	0.490088	0.417154	1.0
4	1.0	0.167273	0.738095	0.355932	0.208333	0.0

train test splitting

```
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=0)
```

check the training and testing data shape

```
x_train.shape
```

```
(275, 6)
```

```
x_test.shape
```

```
(69, 6)
```

```
y_train.shape
```

```
(275,)
```

```
y_test.shape
```

```
(69,)
```

✓ 0s completed at 20:26 ● ×