Bandlamudi sharan 21bai1312

1. Download the dataset penguin_size.csv

2. Load the dataset into the tool

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

```
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

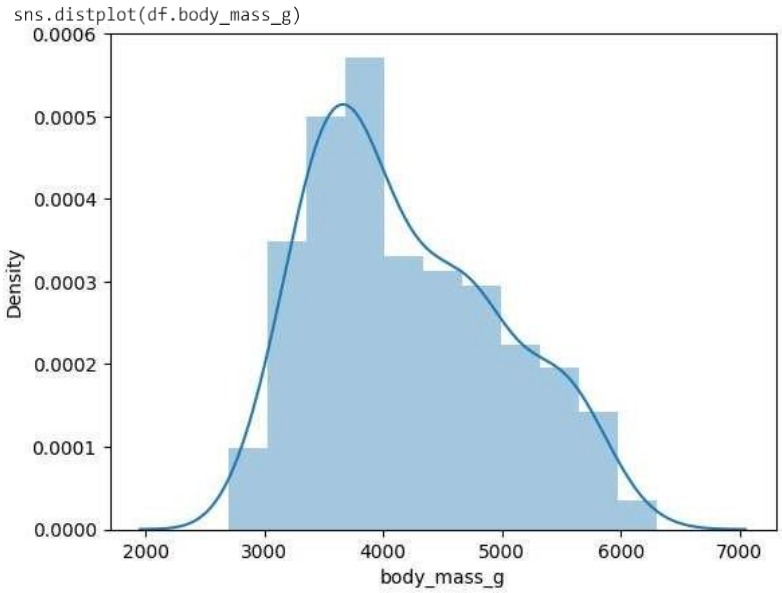
3. Visualizations

3.1 Univariate analysis

```
import seaborn as sns
import matplotlib.pyplot as plt
```

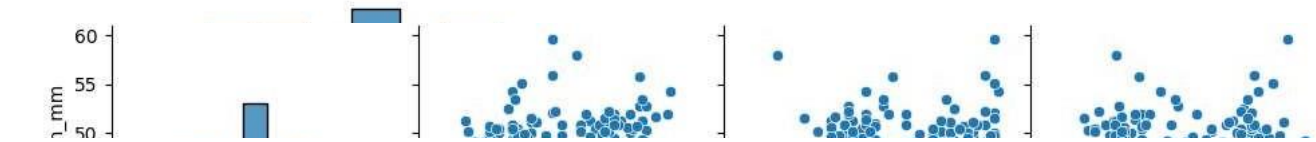
```
sns.distplot(df.body_mass_g)
plt.show()
```

```
<ipython-input-12-6c2911e6788a>: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).
```



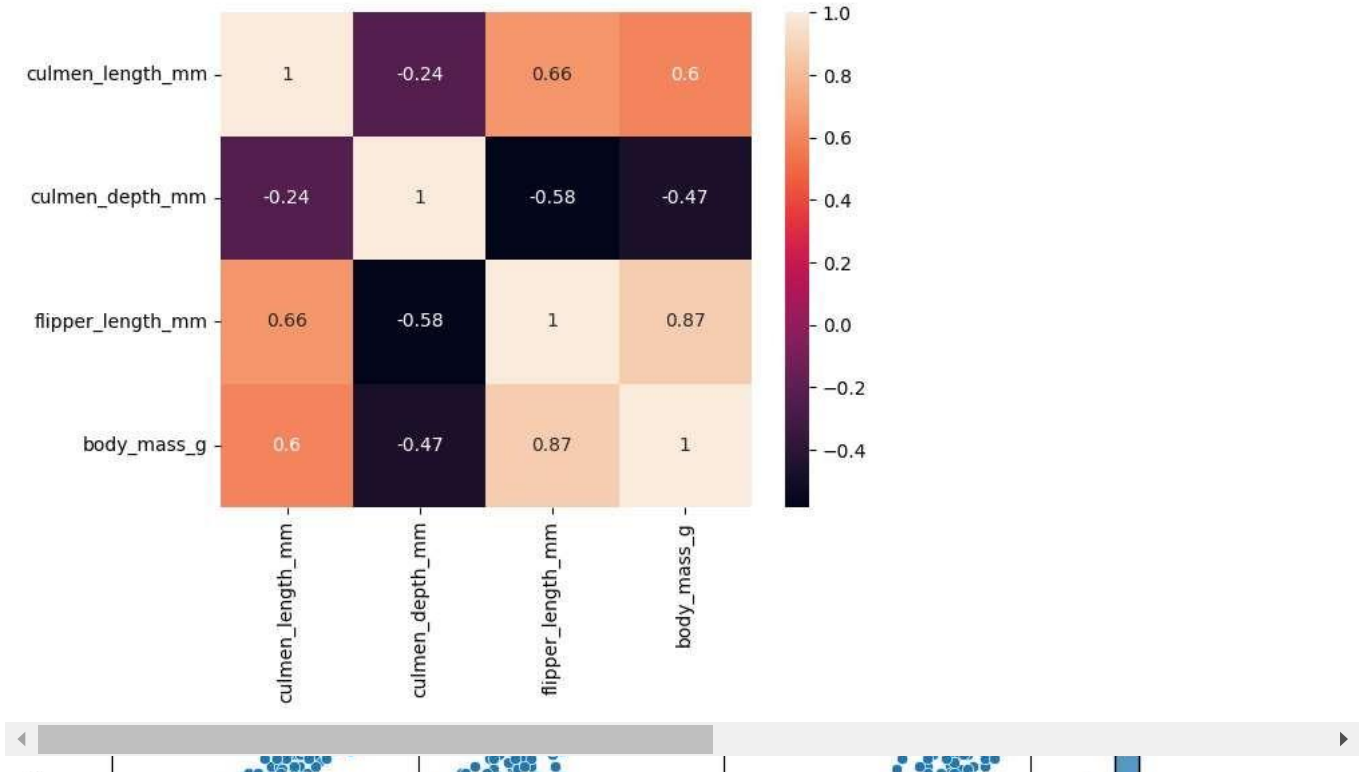
3.2 Bivariate Analysis

```
sns.jointplot(x='culmen_length_mm',y='culmen_depth_mm',data=df)
plt.show()
```



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

```
<ipython-input-16-2d3646dd07cf>:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future ve
  sns.heatmap(df.corr(),annot = True)
<Axes: >
```



4. Descriptive statistics on 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().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()
```

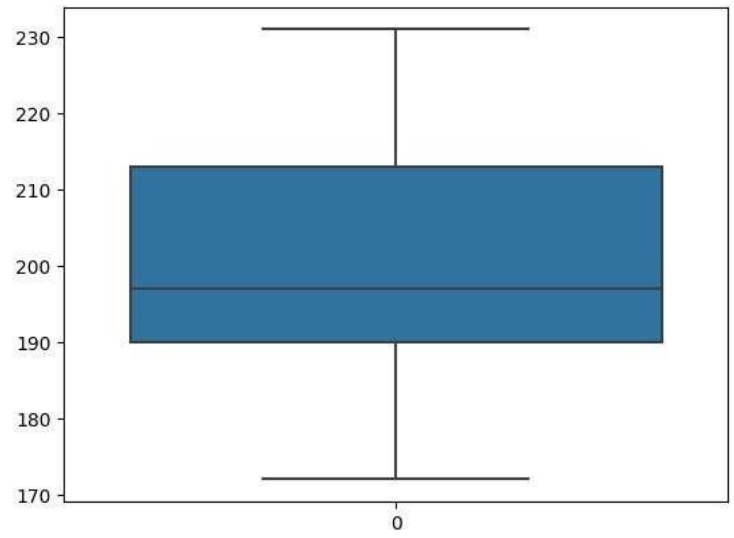
```
species             0
island              0
```

culmen_length_mm 2

<Axes: >

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

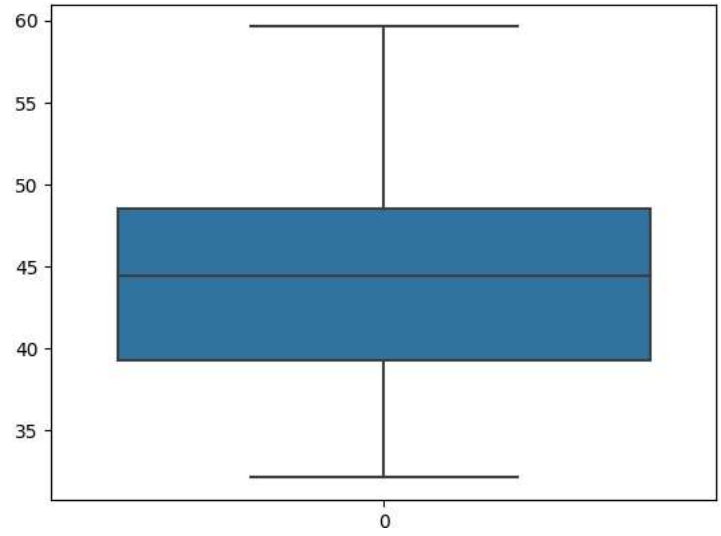
<Axes: >



6. Find the outliers and replace them

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

<Axes: >




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

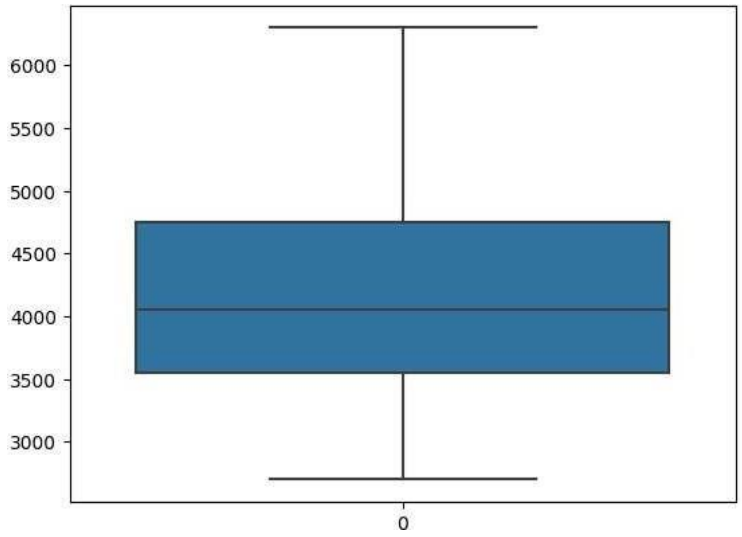
In a f

	culmen_length_mm	culmen_depth_mm	flipper_length_mm	body_mass_g
culmen_length_mm	1.000000	-0.239640	0.659542	0.594217

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

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

```
<Axes: >
```



there are no outliers

7. Check the correlation

```
df.corr()
```

```
<ipython-input-33-2f6f6606aa2c>:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated.
  df.corr()
```

culmen_depth_mm	-0.239640	1.000000	-0.583832	-0.471942
flipper_length_mm	0.659542	-0.583832	1.000000	0.871221
body_mass_g	0.594217	-0.471942	0.871221	1.000000

8. Check the categorical columns and perform encoding

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

```
df['species'] = le.fit_transform(df['species'])
df['island'] = le.fit_transform(df['island'])
df.head()
```

	species	island	culmen_length_mm	culmen_depth_mm	flipper_length_mm	body_mass_g	sex
0	0	2	42	18.7	181.0	3750.0	2
1	0	2	45	17.4	186.0	3800.0	1
2	0	2	51	18.0	195.0	3250.0	1
3	0	2	85	17.3	197.0	4050.0	2
4	0	2	22	19.3	193.0	3450.0	1

9. Split the data into independent and dependent 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	42	18.7	181.0	3750.0	2
1	2	45	17.4	186.0	3800.0	1
2	2	51	18.0	195.0	3250.0	1
3	2	85	17.3	197.0	4050.0	2
4	2	22	19.3	193.0	3450.0	1

```
y = df['species']
y.head()
```

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

10. Scale the 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	0.256098	0.666667	0.152542	0.291667	1.0
1	1.0	0.274390	0.511905	0.237288	0.305556	0.5
2	1.0	0.310976	0.583333	0.389831	0.152778	0.5
3	1.0	0.518293	0.500000	0.423729	0.375000	1.0
4	1.0	0.134146	0.738095	0.355932	0.208333	0.5

11. 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 = 0.2, random_state= 0)
```

12. 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,)
```