assignment3-21bds0001

September 14, 2023

Mukund Niranjan 21BDS0001

[ ]:
```
import pandas as pd
df=pd.read_csv('/content/penguins_size.csv')
df.head()
```

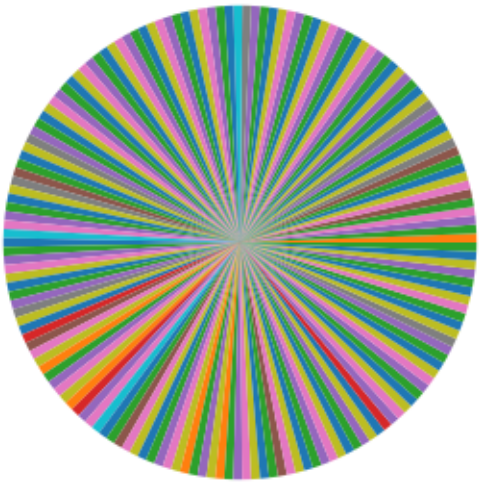
[ ]:

	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

Univariate Analysis

Pie Chart

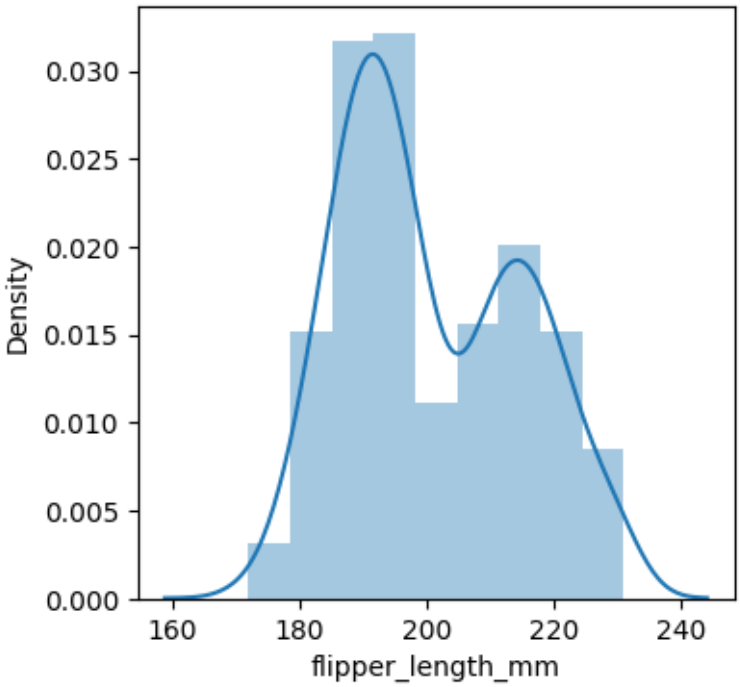
[ ]:
```
import matplotlib.pyplot as plt
import seaborn as sns
plt.figure(figsize=(4,4))
condition=df['sex']=='FEMALE'
plt.pie(condition)
plt.show()
```



Distribution Plot

[ ]:
```
plt.figure(figsize=(4,4))
sns.distplot(df['flipper_length_mm'])
plt.show()
```

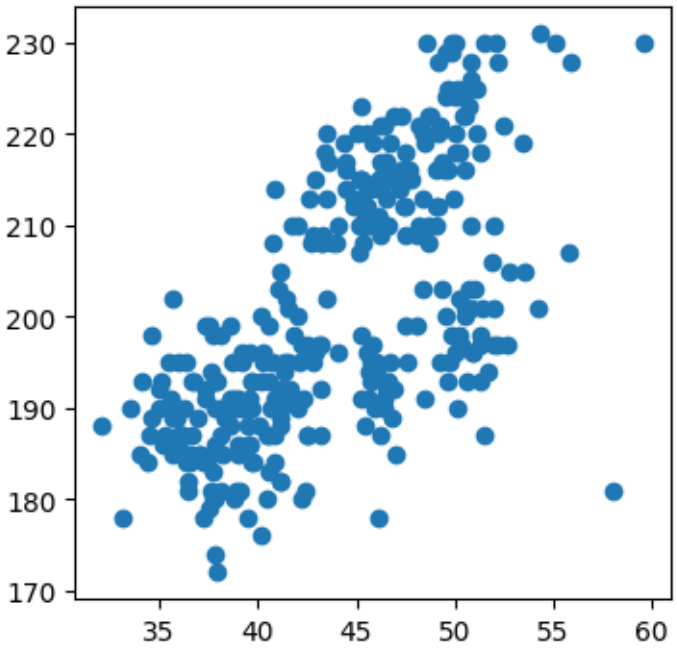
```
<ipython-input-4-b1fc4057de4c>:2: UserWarning:

`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'])
```



Bivariate Analysis

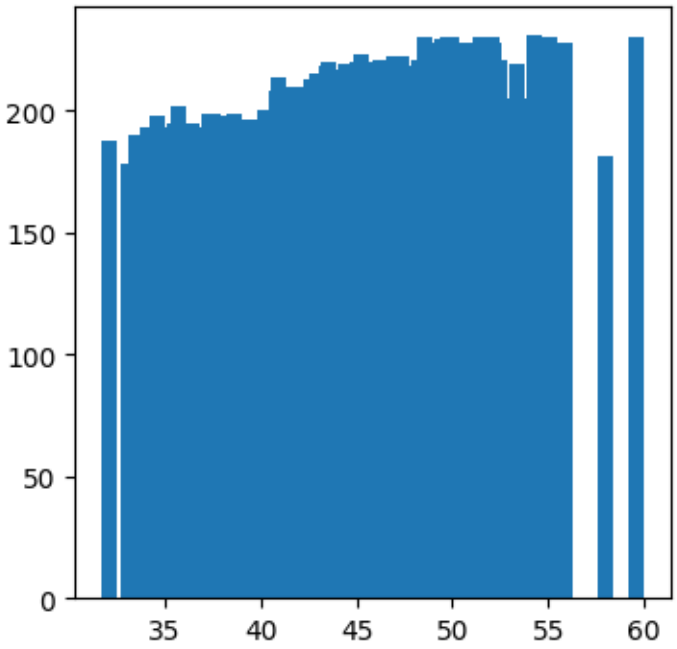
Scatter Graph

[ ]:
```
plt.figure(figsize=(4,4))
plt.scatter(df['culmen_length_mm'], df['flipper_length_mm'])
plt.show()
```



Bar Graph

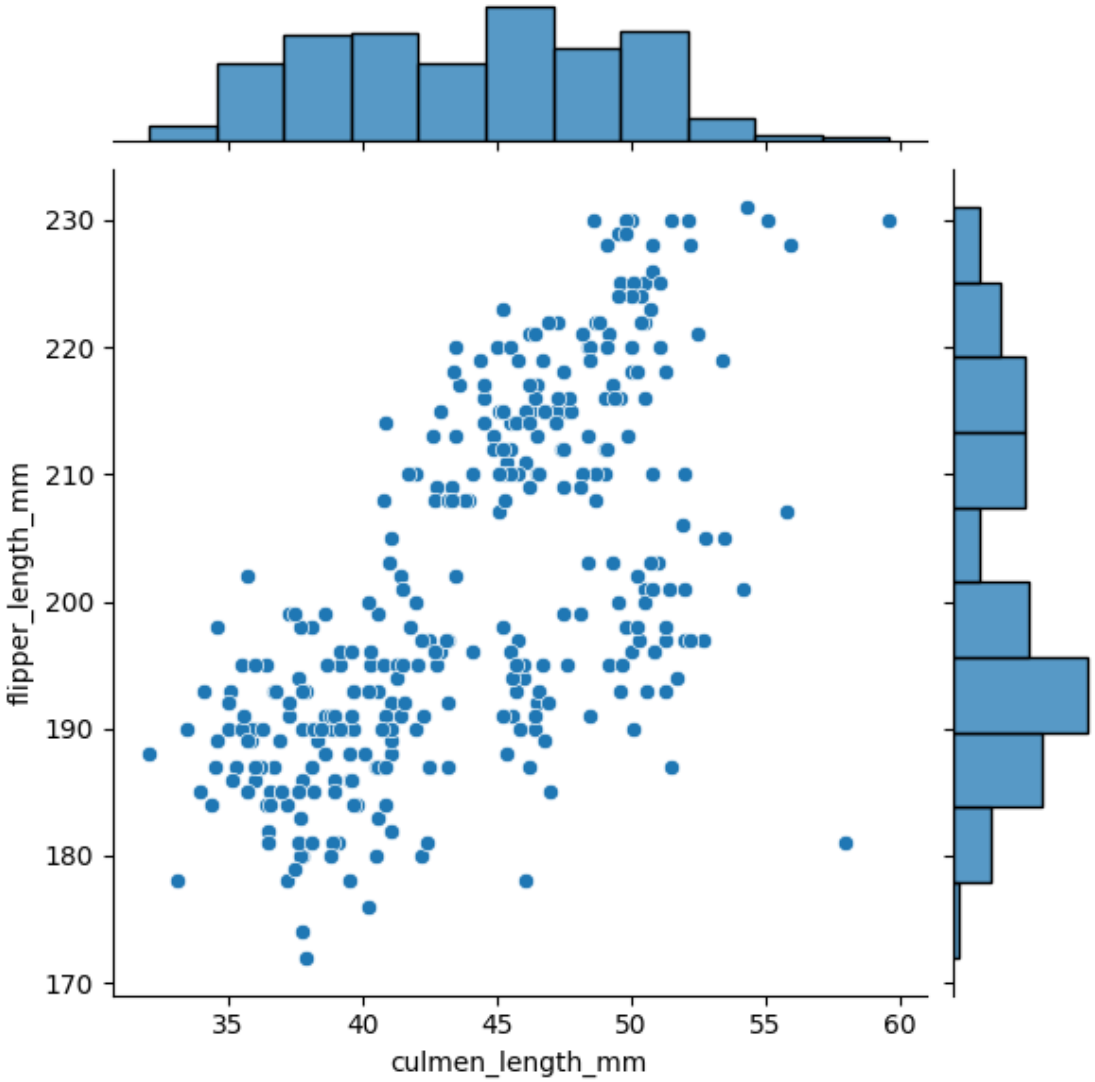
[ ]:
```
plt.figure(figsize=(4,4))
plt.bar(df['culmen_length_mm'], df['flipper_length_mm'])
plt.show()
```



Joint Plot

[ ]:
```
sns.jointplot(x='culmen_length_mm', y='flipper_length_mm',data=df)
```

[ ]:
```
<seaborn.axisgrid.JointGrid at 0x7e9f4e987370>
```



Multivariate Analysis

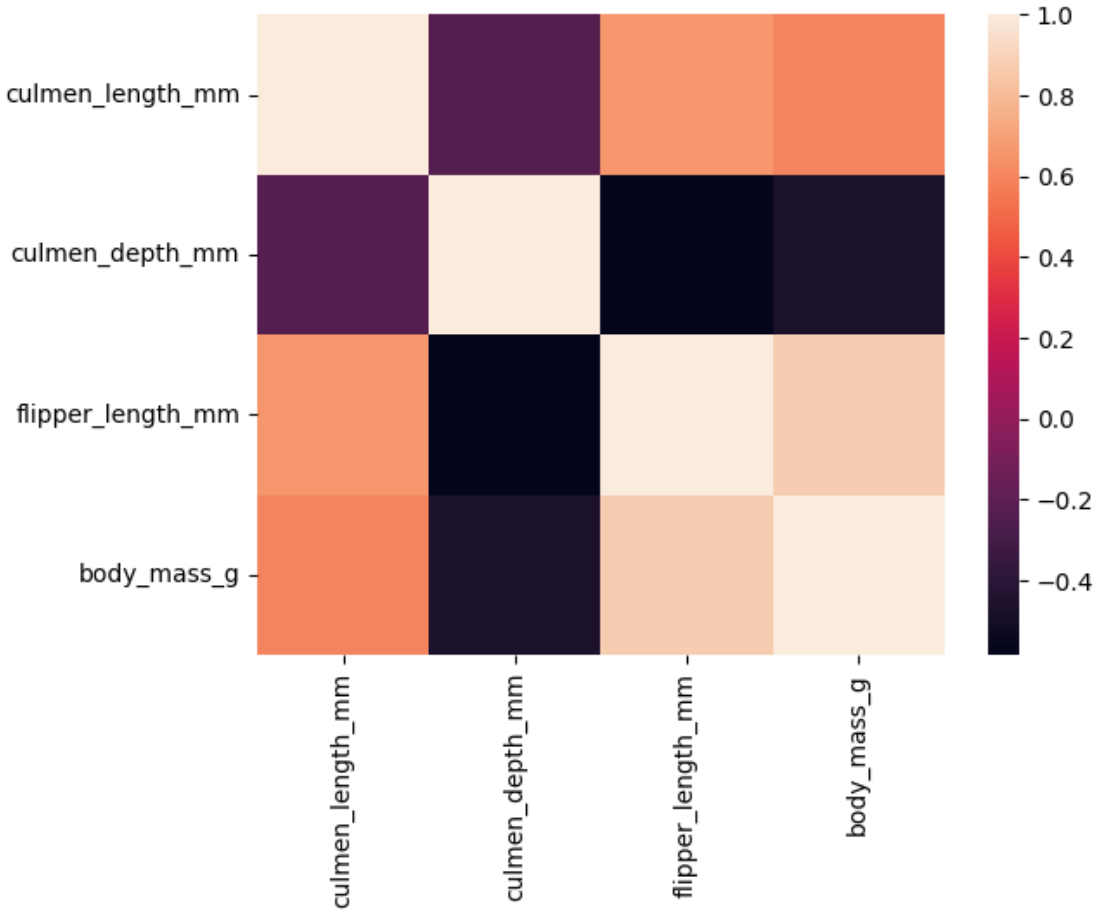
Heatmap

[ ]:
```
sns.heatmap(df.corr())
```

```
<ipython-input-10-aa4f4450a243>:1: FutureWarning: The default value of
```

```
numeric_only in 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.
  sns.heatmap(df.corr())
```

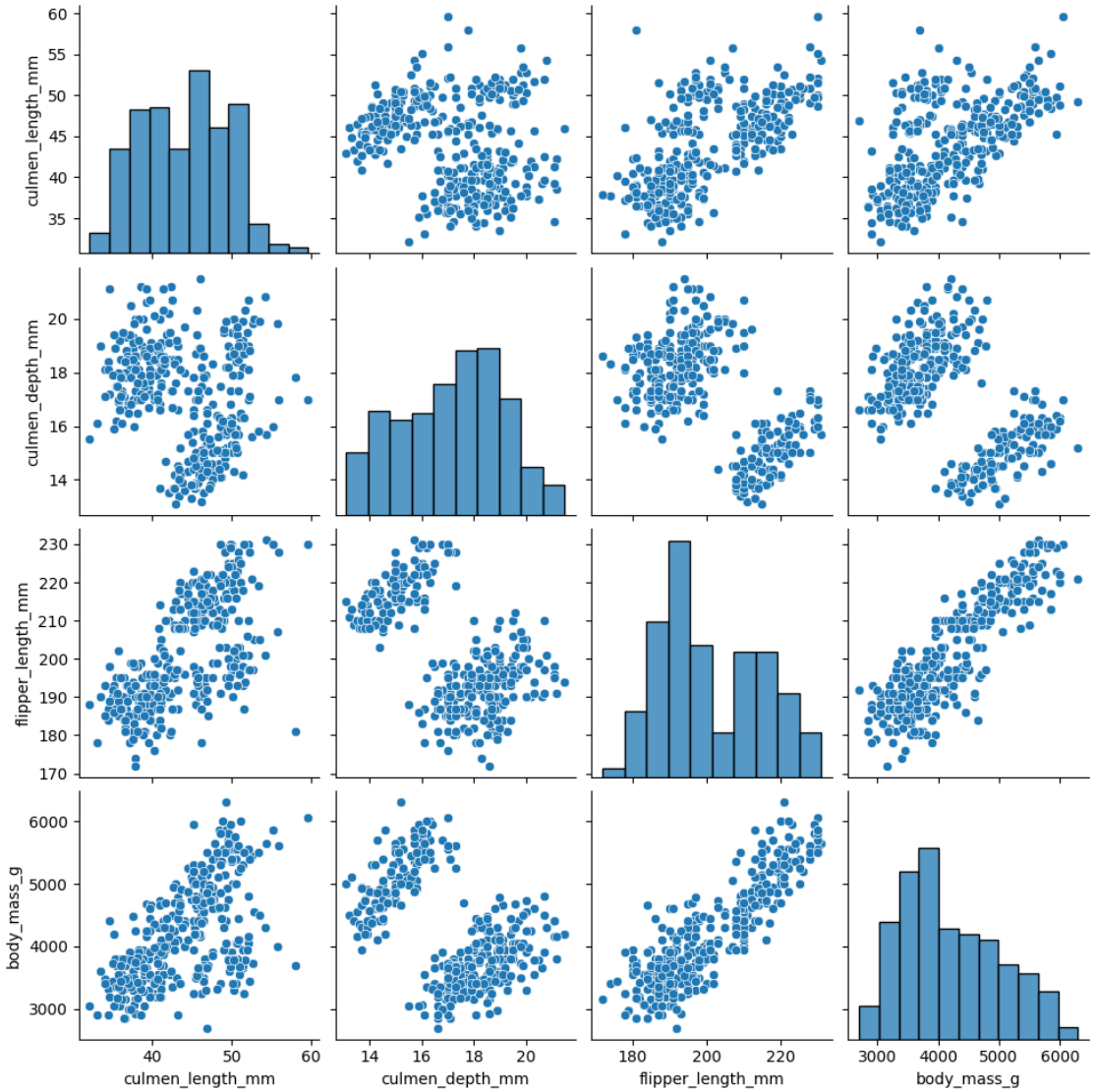
[ ]: <Axes: >



Pairplot

[ ]:
```
sns.pairplot(df)
```

[ ]: <seaborn.axisgrid.PairGrid at 0x7e9f525c3790>



Describe

[ ]:
```
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

Check for Missing values and deal with them

[13]:
```
df.isnull().any()
```

[13]:
```
species              False
island               False
culmen_length_mm      True
culmen_depth_mm       True
flipper_length_mm     True
body_mass_g           True
sex                   True
dtype: bool
```

[14]:
```
df.sex.value_counts ()
```

[14]:
```
MALE      168
FEMALE    165
.           1
Name: sex, dtype: int64
```

[15]:
```
df['sex']=df['sex'].replace(".", "MALE")
df.sex. value_counts ()
```

[15]:
```
MALE      169
FEMALE    165
Name: sex, dtype: int64
```

[16]:
```
df['sex']=df['sex'].fillna ("MALE")
```

[17]:
```
df.median()
```

```
<ipython-input-17-6d467abf240d>:1: FutureWarning: The default value of
numeric_only in DataFrame.median is deprecated. In a future version, it will
default to False. In addition, specifying 'numeric_only=None' is deprecated.
Select only valid columns or specify the value of numeric_only to silence this
warning.
  df.median()
```

[17]:
```
culmen_length_mm       44.45
culmen_depth_mm        17.30
flipper_length_mm     197.00
body_mass_g          4050.00
dtype: float64
```

[18]:
```
df=df.fillna(df.median ( ))
df.isnull ().sum()
```

```
<ipython-input-18-fea379c4db1f>:1: FutureWarning: The default value of
```

```
numeric_only in DataFrame.median is deprecated. In a future version, it will
default to False. In addition, specifying 'numeric_only=None' is deprecated.
Select only valid columns or specify the value of numeric_only to silence this
warning.
  df=df.fillna(df.median ( ))
```

[18]:
```
species              0
island               0
culmen_length_mm     0
culmen_depth_mm      0
flipper_length_mm    0
body_mass_g          0
sex                  0
dtype: int64
```

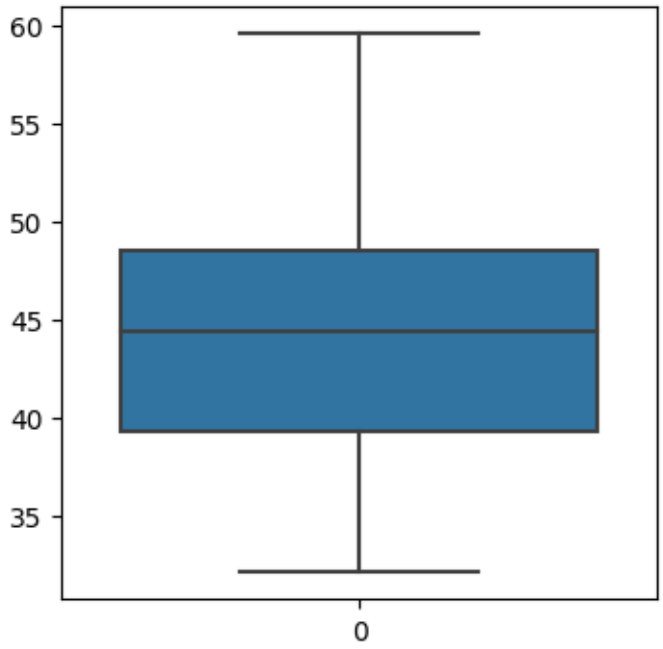
[19]:
```
df.info()
```

```
<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
```

Find the outliers and replace them outliers

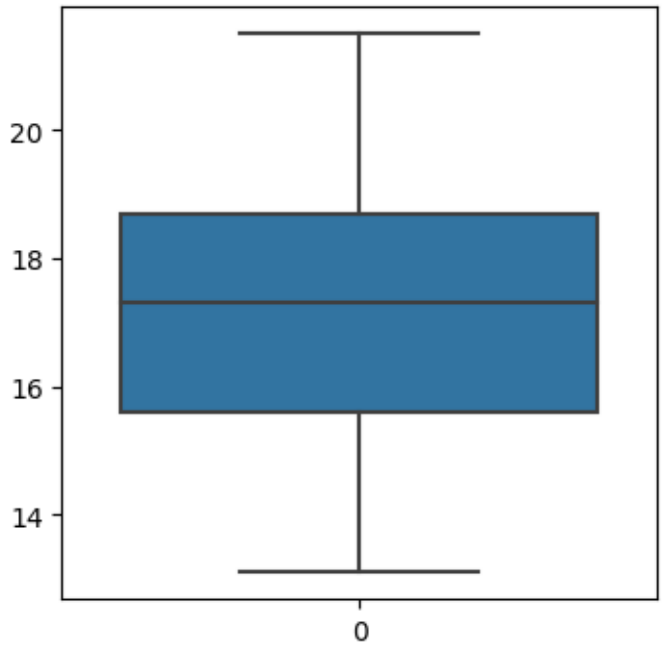
[20]:
```
plt.figure(figsize=(4,4))
sns.boxplot(df.culmen_length_mm)
```

[20]:
```
<Axes: >
```



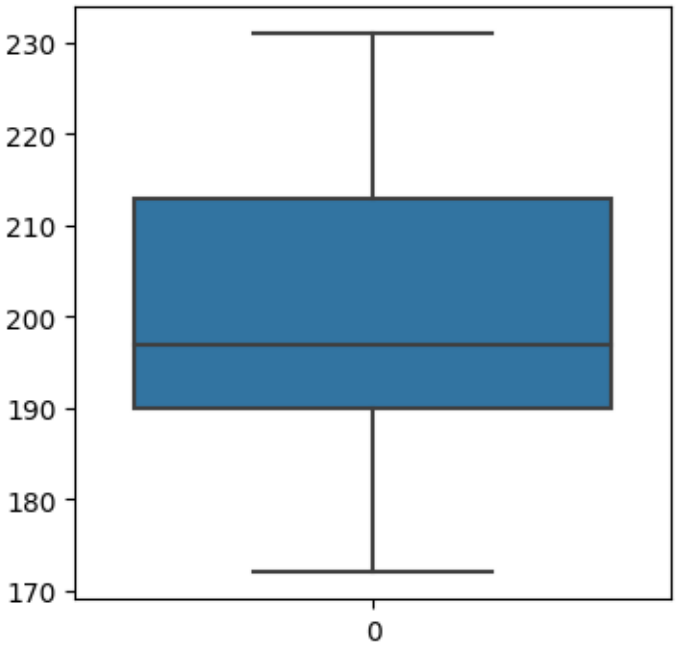
[21]:
```
plt.figure(figsize=(4,4))
sns.boxplot(df.culmen_depth_mm)
```

[21]: <Axes: >



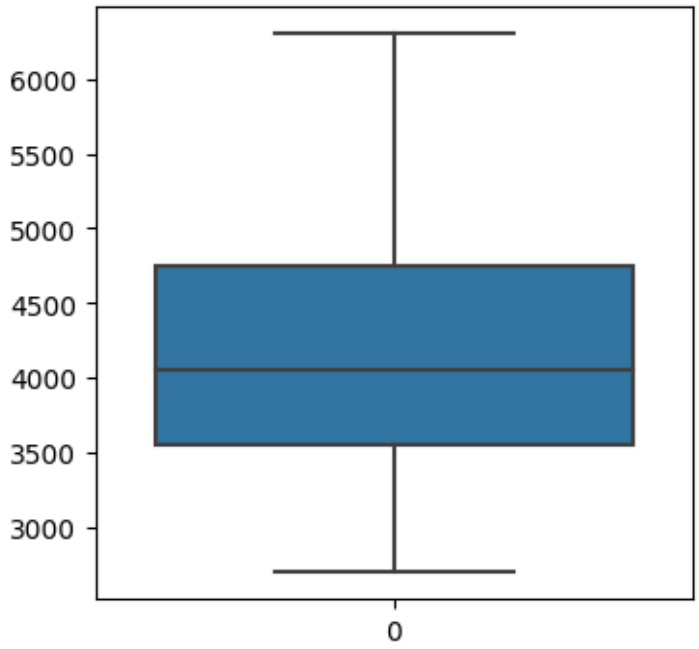
[22]:
```
plt.figure(figsize=(4,4))
sns.boxplot(df.flipper_length_mm)
```

[22]: <Axes: >



[23]:
```
plt.figure(figsize=(4,4))
sns.boxplot(df.body_mass_g)
```

[23]: <Axes: >



Check for Categorical columns and perform encoding.

[24]:
```
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
```

[25]:
```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
df['sex'] = le.fit_transform(df['sex'])
df['species'] = le.fit_transform(df['species'])
df['island'] = le.fit_transform(df['island'])
df.head()
```

[25]:
```
   species  island  culmen_length_mm  culmen_depth_mm  flipper_length_mm  \
0        0       2              39.10             18.7              181.0
1        0       2              39.50             17.4              186.0
2        0       2              40.30             18.0              195.0
3        0       2              44.45             17.3              197.0
4        0       2              36.70             19.3              193.0

   body_mass_g  sex
0       3750.0    1
1       3800.0    0
2       3250.0    0
3       4050.0    1
4       3450.0    0
```

[26]:
```
df.corr().species.sort_values(ascending=False)
```

[26]:
```
species              1.000000
flipper_length_mm    0.850819
body_mass_g          0.747547
culmen_length_mm     0.728706
sex                  0.010240
island              -0.635659
culmen_depth_mm     -0.741282
Name: species, dtype: float64
```

[27]:
```
x=df.drop(columns=['species'], axis=1)
y=df.species
x.head()
```

[27]:
```
   island  culmen_length_mm  culmen_depth_mm  flipper_length_mm  body_mass_g  \
0       2              39.10             18.7              181.0       3750.0
1       2              39.50             17.4              186.0       3800.0
2       2              40.30             18.0              195.0       3250.0
3       2              44.45             17.3              197.0       4050.0
4       2              36.70             19.3              193.0       3450.0

   sex
0    1
1    0
2    0
3    1
4    0
```

[28]:
```
y.head()
```

[28]:
```
0    0
1    0
```

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

[29]:
```
from sklearn.preprocessing import MinMaxScaler
scale=MinMaxScaler()
x_s=pd.DataFrame(scale.fit_transform(x),columns=x.columns)
x_s.head()
```

[29]:

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

	sex
0	1.0
1	0.0
2	0.0
3	1.0
4	0.0

[30]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x_s,y,test_size=0.
 ↪2,random_state=0)
```

[31]:
```
x_train.shape
```

[31]: (275, 6)

[32]:
```
x_test.shape
```

[32]: (69, 6)

[33]:
```
y_train.shape
```

[33]: (275,)

[34]:
```
y_test.shape
```

[34]: (69,)