sptone-project-aboh-michael-27a-2

May 14, 2024

0.1 Capstone project for Basic Data Science

In this project, you would explore the Data preprocessing techniques learned. You will use, pandas, plots and the different data preprocessing techniques to explore and analyse the data. This project is on Exploratory Data Analysis

Step1: Import the necessary libraries

[7]:
```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import seaborn as sns
%matplotlib inline
```

[12]:
```
#List the set of available datasets in seaborn
print(len(sns.get_dataset_names()))
print(sns.get_dataset_names())
```

```
88
['anagrams', 'anscombe', 'attention', 'brain_networks', 'car_crashes',
'diamonds', 'dots', 'dowjones', 'exercise', 'flights', 'fmri', 'geyser', 'glue',
'healthexp', 'iris', 'mpg', 'penguins', 'planets', 'seaice', 'taxis', 'tips',
'titanic', 'anagrams', 'anagrams', 'anscombe', 'anscombe', 'attention',
'attention', 'brain_networks', 'brain_networks', 'car_crashes', 'car_crashes',
'diamonds', 'diamonds', 'dots', 'dots', 'dowjones', 'dowjones', 'exercise',
'exercise', 'flights', 'flights', 'fmri', 'fmri', 'geyser', 'geyser', 'glue',
'glue', 'healthexp', 'healthexp', 'iris', 'iris', 'mpg', 'mpg', 'penguins',
'penguins', 'planets', 'planets', 'seaice', 'seaice', 'taxis', 'taxis', 'tips',
'tips', 'titanic', 'titanic', 'anagrams', 'anscombe', 'attention',
'brain_networks', 'car_crashes', 'diamonds', 'dots', 'dowjones', 'exercise',
'flights', 'fmri', 'geyser', 'glue', 'healthexp', 'iris', 'mpg', 'penguins',
'planets', 'seaice', 'taxis', 'tips', 'titanic']
```

Dataset 1: exercise

[10]:
```
#1.Write the syntax for choosing the specific dataset
df=sns.load_dataset('penguins')
df
```

[10]:

```
  species     island  bill_length_mm  bill_depth_mm  flipper_length_mm  \
0    Adelie  Torgersen            39.1           18.7              181.0
1    Adelie  Torgersen            39.5           17.4              186.0
2    Adelie  Torgersen            40.3           18.0              195.0
3    Adelie  Torgersen             NaN            NaN                NaN
4    Adelie  Torgersen            36.7           19.3              193.0
..      …        …              …             …                 …
339  Gentoo     Biscoe             NaN            NaN                NaN
340  Gentoo     Biscoe            46.8           14.3              215.0
341  Gentoo     Biscoe            50.4           15.7              222.0
342  Gentoo     Biscoe            45.2           14.8              212.0
343  Gentoo     Biscoe            49.9           16.1              213.0

     body_mass_g     sex
0         3750.0    Male
1         3800.0  Female
2         3250.0  Female
3            NaN     NaN
4         3450.0  Female
..           …       …
339          NaN     NaN
340       4850.0  Female
341       5750.0    Male
342       5200.0  Female
343       5400.0    Male

[344 rows x 7 columns]
```

[13]:

```
#2.Display the top 5 records of penguin dataset
#3.Display the dimensionality of the dataset
print(df.head())
print(df.shape)
```

```
  species     island  bill_length_mm  bill_depth_mm  flipper_length_mm  \
0  Adelie  Torgersen            39.1           18.7              181.0
1  Adelie  Torgersen            39.5           17.4              186.0
2  Adelie  Torgersen            40.3           18.0              195.0
3  Adelie  Torgersen             NaN            NaN                NaN
4  Adelie  Torgersen            36.7           19.3              193.0

   body_mass_g     sex
0       3750.0    Male
1       3800.0  Female
2       3250.0  Female
3          NaN     NaN
4       3450.0  Female
(344, 7)
```

[14]:

```
#4.Display the datatypes of the attributes
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   bill_length_mm     342 non-null    float64
 3   bill_depth_mm      342 non-null    float64
 4   flipper_length_mm  342 non-null    float64
 5   body_mass_g        342 non-null    float64
 6   sex                333 non-null    object
dtypes: float64(4), object(3)
memory usage: 18.9+ KB
```

[15]:

```
#5.Describe the dataset extensively
df.describe()
```

[15]:

```
       bill_length_mm  bill_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
```

[16]:

```
#6.Display the number of null values in the dataset and the total count of it
df.isnull().sum()
```

[16]:

```
species               0
island                0
bill_length_mm        2
bill_depth_mm         2
flipper_length_mm     2
body_mass_g           2
sex                  11
dtype: int64
```

[18]:

```
#7.Identify the columns having null values and remove them
penguins_cleaned=df.dropna()
```

[19]:

```
# 8.check the size of dataset after cleaning and compare with the size before␣
 ↪cleaning
print('Before cleaning:', df.shape)
print('After cleaning:',penguins_cleaned.shape)
```

```
Before cleaning: (344, 7)
After cleaning: (333, 7)
```

[20]:

```
#9.Find out the count of male and female penguins
penguins_cleaned['sex'].value_counts()
```

[20]:

```
sex
Male      168
Female    165
Name: count, dtype: int64
```

[21]:

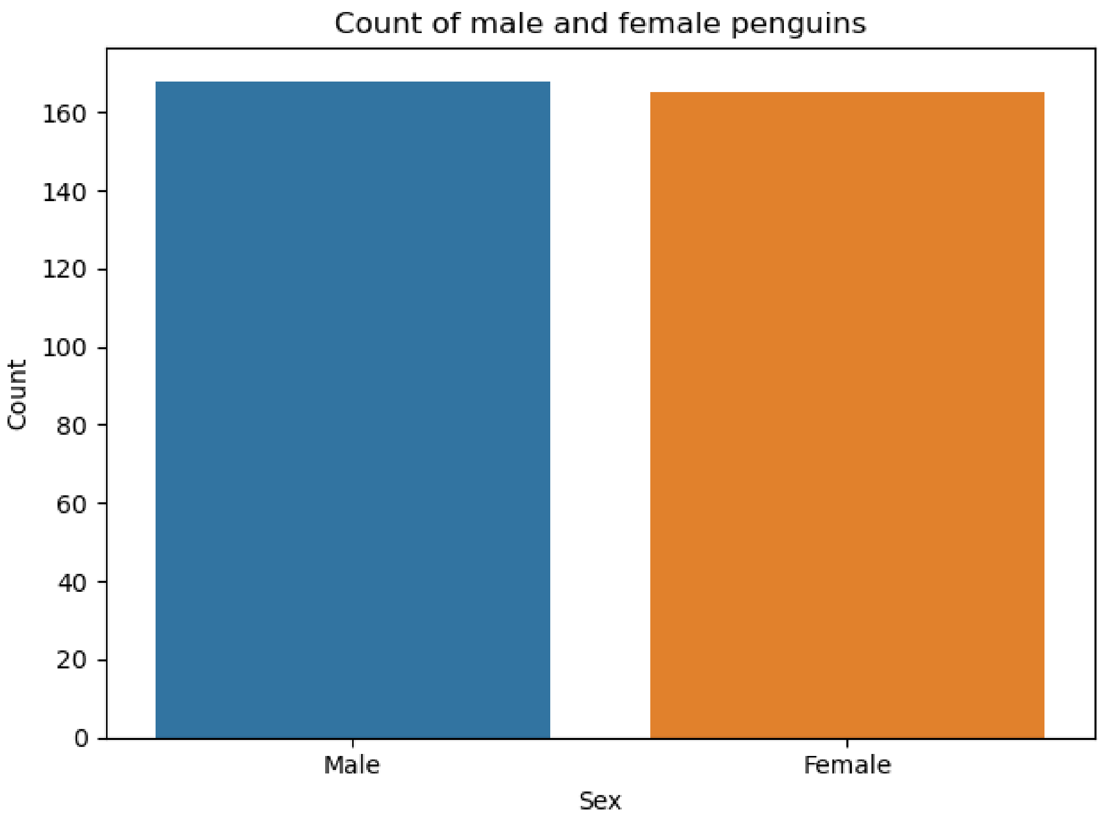
```
#10.Find out the count of species
penguins_cleaned['species'].value_counts()
```

[21]:

```
species
Adelie       146
Gentoo       119
Chinstrap     68
Name: count, dtype: int64
```

[22]:

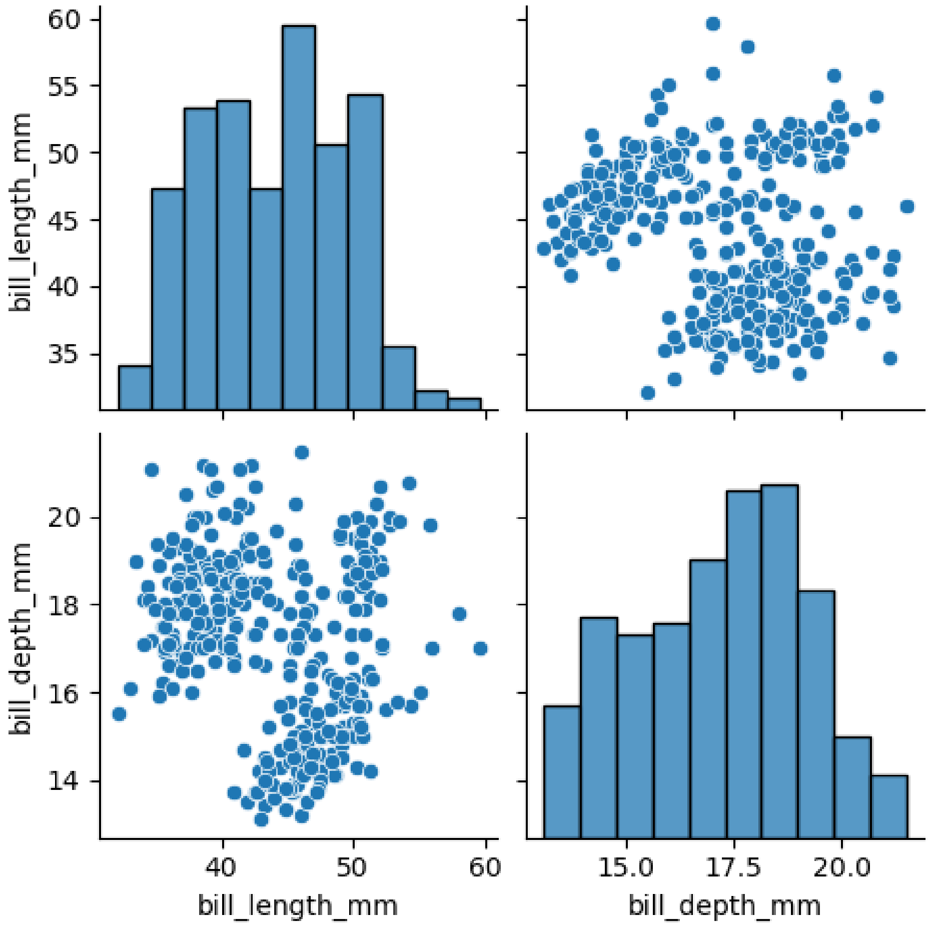
```
#11. Use a countplot graph to display the number of male and female penguins
#Give the plot a title "Count of male and female penguins"
#Give a xlabel and ylabel
sns.countplot( x="sex",data=penguins_cleaned)
plt.title("Count of male and female penguins")
plt.xlabel("Sex")
plt.ylabel("Count")
plt.tight_layout()
plt.show()
```



[23]:
```
#12. For all the penguins display the bill_length_mm and bill_depth_mm using␣
↪pairplot
sns.pairplot(df[['bill_length_mm','bill_depth_mm']])
```

```
C:\Users\PC\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning:
use_inf_as_na option is deprecated and will be removed in a future version.
Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\PC\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning:
use_inf_as_na option is deprecated and will be removed in a future version.
Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

[23]:
```
<seaborn.axisgrid.PairGrid at 0x226b68b16d0>
```



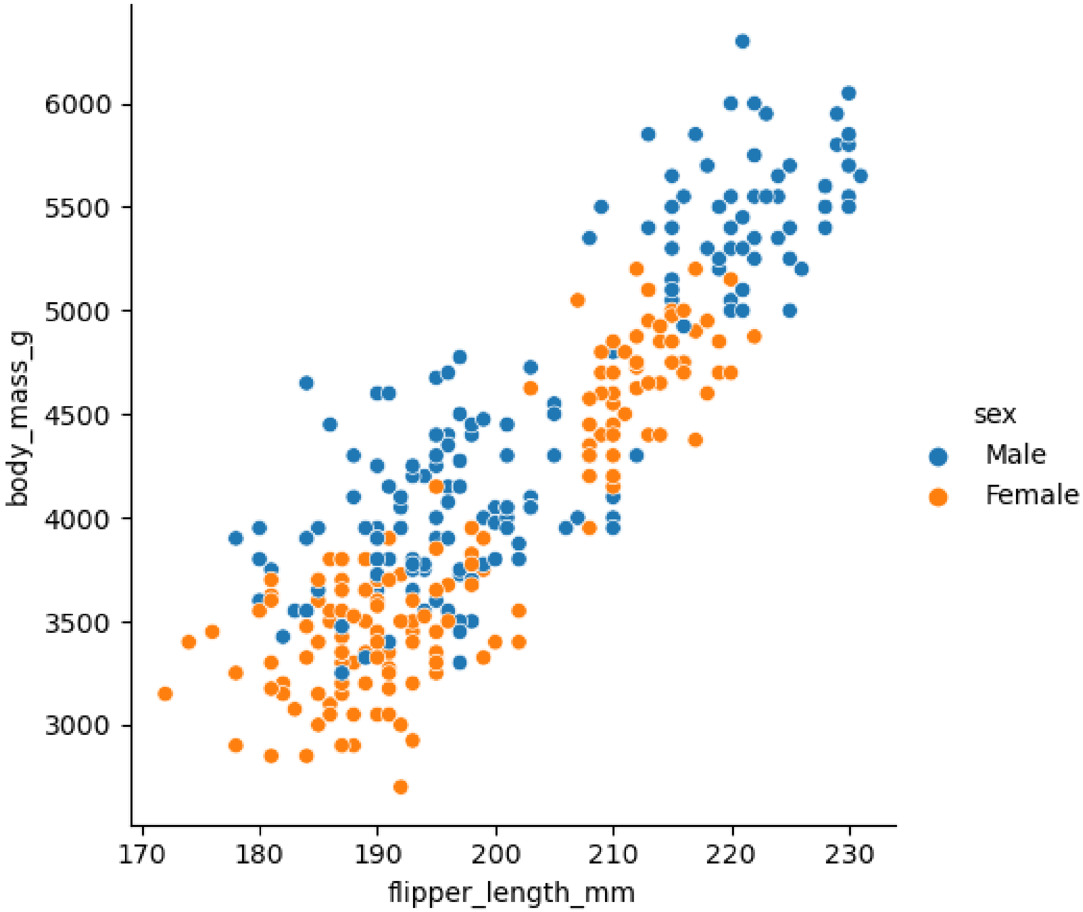
```
[24]: #13.Create a visualization to display the flipper_length_mm and body_mass_g for␣
      ↪the penguin dataset.
      sns.relplot (

          data=penguins_cleaned,

          x="flipper_length_mm", y="body_mass_g", hue = "sex"

      )
```

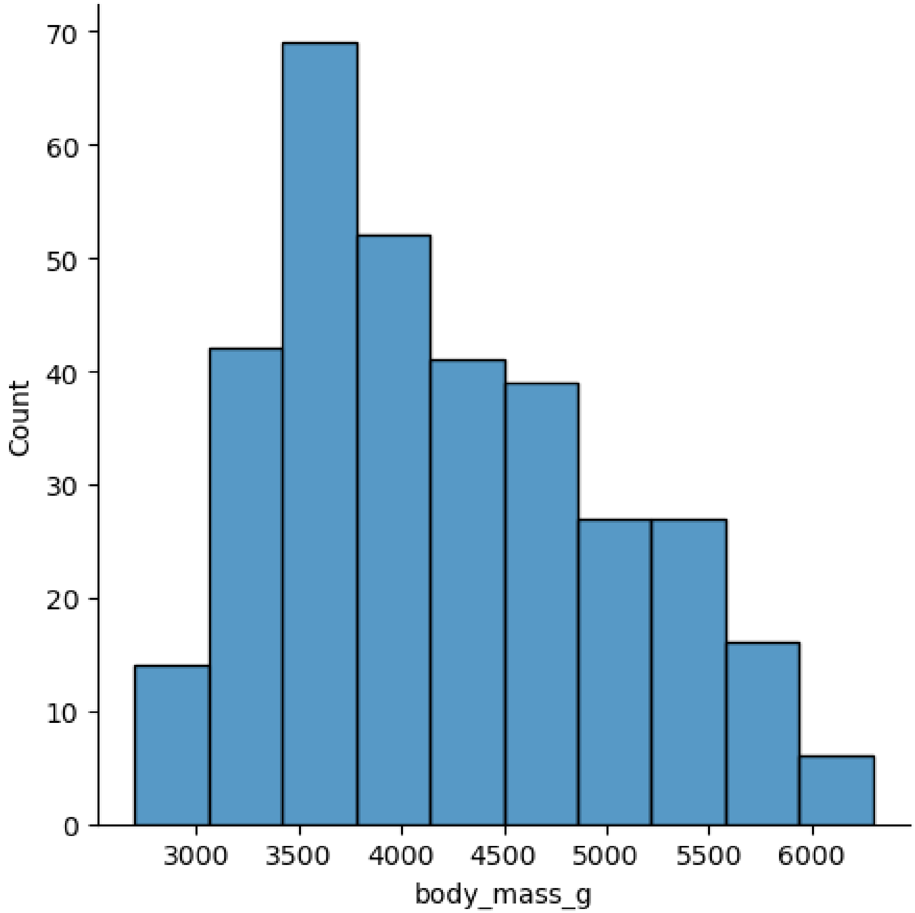
```
[24]: <seaborn.axisgrid.FacetGrid at 0x226b7fedf10>
```



```
[25]: #14. Create a histogram with the flipper_length_mm and bins = 10
      sns.displot(penguins_cleaned["body_mass_g"], kde= False, bins = 10)
```

```
C:\Users\PC\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning:
use_inf_as_na option is deprecated and will be removed in a future version.
Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

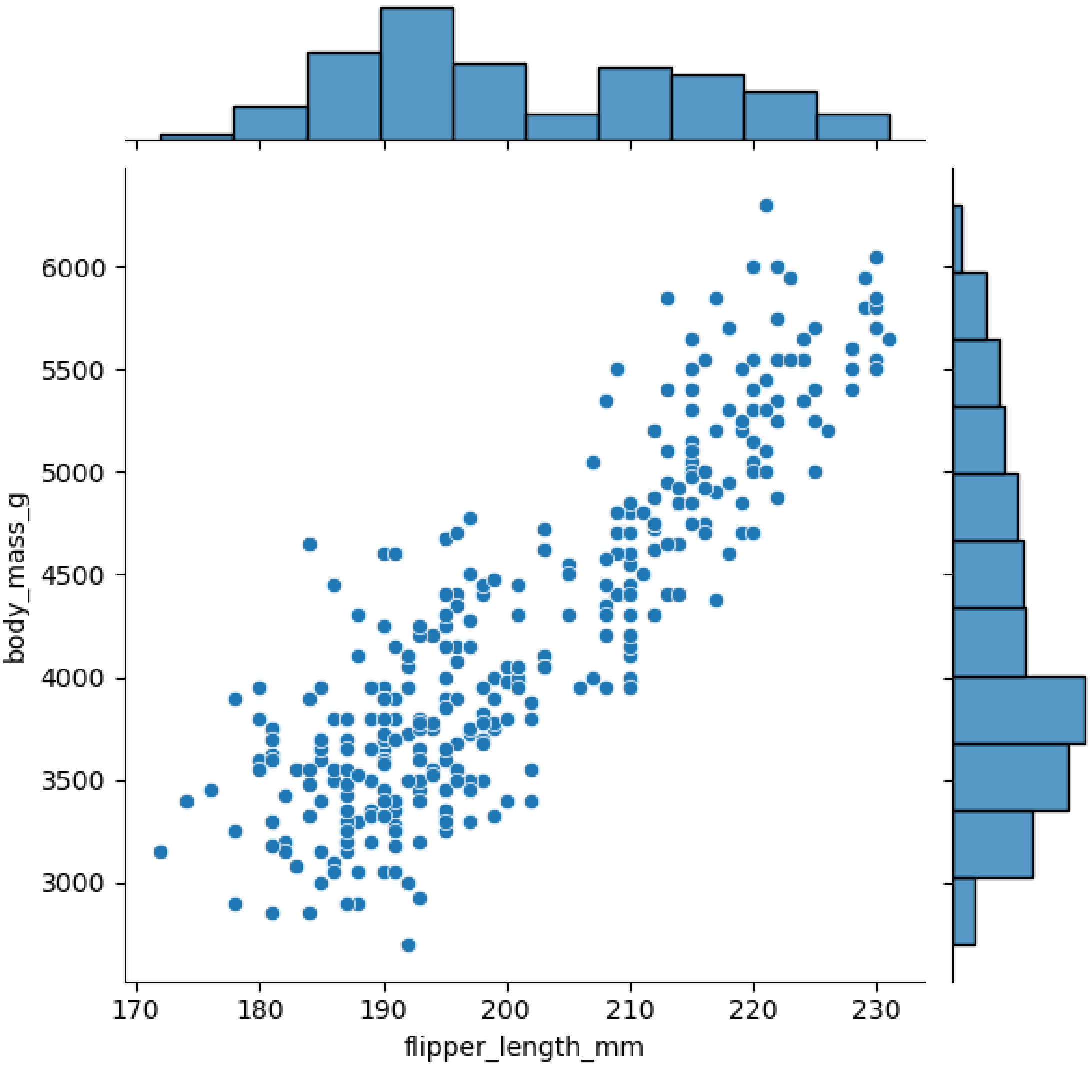
```
[25]: <seaborn.axisgrid.FacetGrid at 0x226b76a82d0>
```



```
[26]: #15. Create a joint point with body_length_mm, body_depth_mm for␣
        ↪penguins_cleaned dataset
      sns.jointplot(x= 'flipper_length_mm' , y= 'body_mass_g' , data =␣
        ↪penguins_cleaned)
```

```
C:\Users\PC\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning:
use_inf_as_na option is deprecated and will be removed in a future version.
Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\PC\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning:
use_inf_as_na option is deprecated and will be removed in a future version.
Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

```
[26]: <seaborn.axisgrid.JointGrid at 0x226b76af810>
```

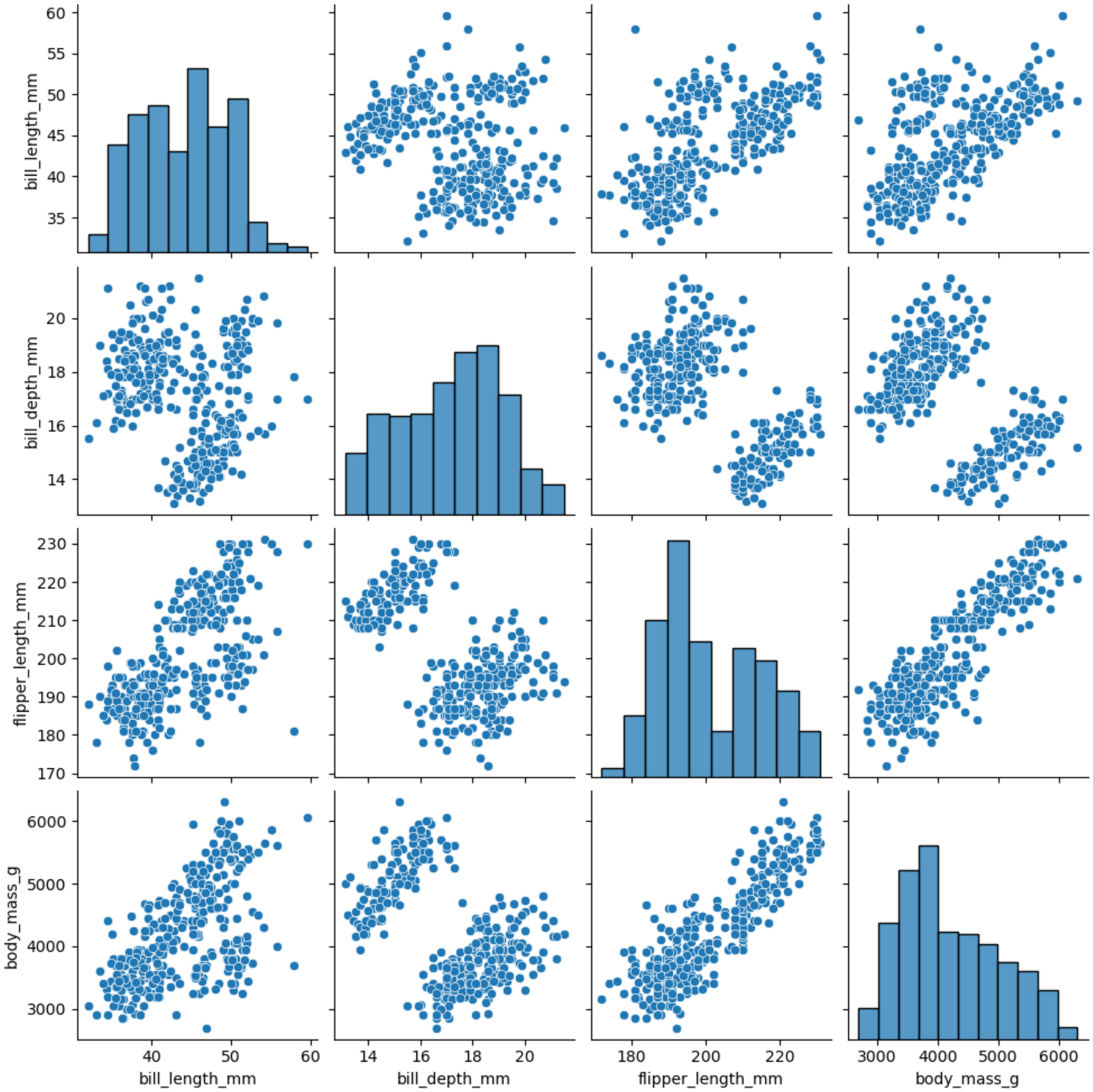


```
[27]: #16. Use a pairwise plot to display all the numerical values of the dataset␣
        ↪penguins_cleaned
      sns.pairplot(penguins_cleaned)
```

```
C:\Users\PC\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning:
use_inf_as_na option is deprecated and will be removed in a future version.
Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\PC\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning:
use_inf_as_na option is deprecated and will be removed in a future version.
Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\PC\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning:
use_inf_as_na option is deprecated and will be removed in a future version.
```

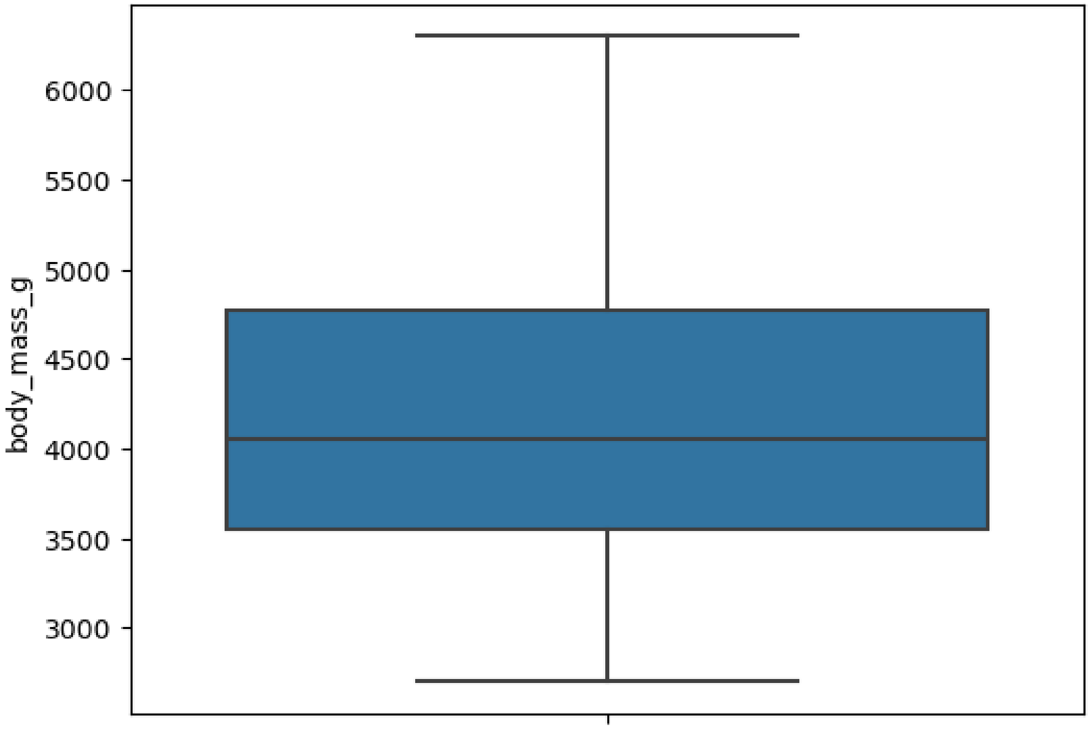
```
Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\PC\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning:
use_inf_as_na option is deprecated and will be removed in a future version.
Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

[27]:
```
<seaborn.axisgrid.PairGrid at 0x226b80e2e50>
```



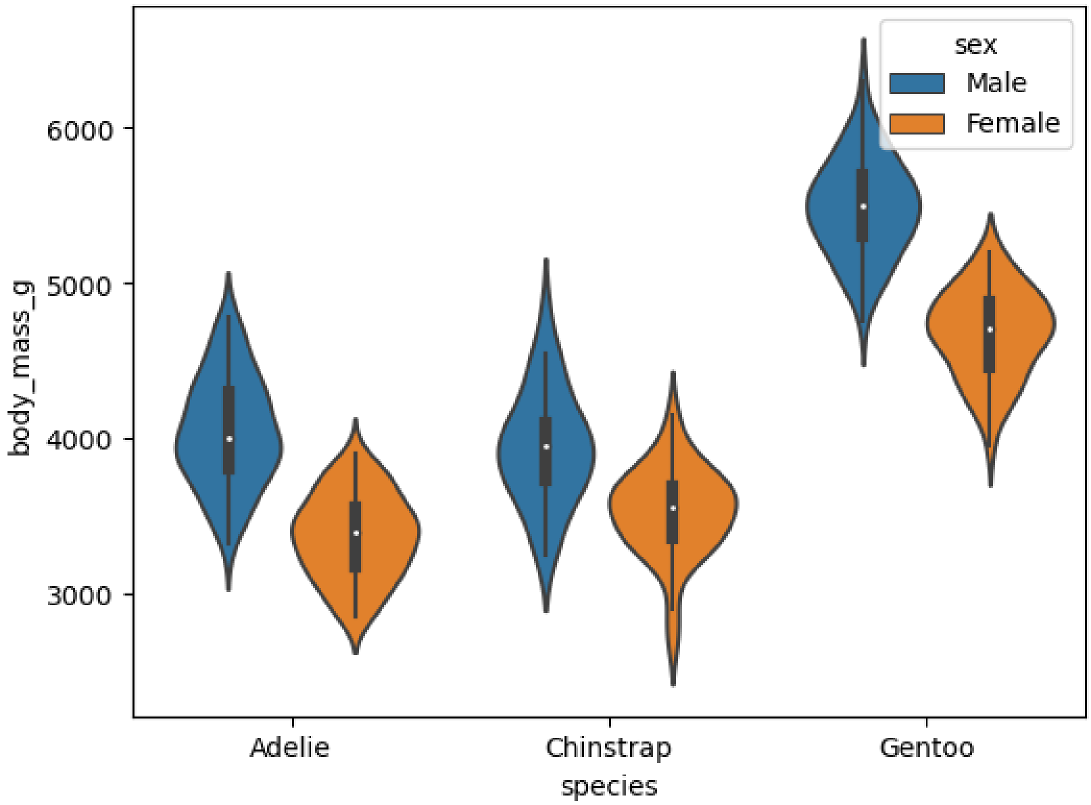
[28]:
```
#17.Check if there exists any outliers based on the boxplot based on numerical␣
 ↪values  of penguins_cleaned dataset
sns.boxplot(y='body_mass_g', data= penguins_cleaned )
```

```
[28]: <Axes: ylabel='body_mass_g'>
```



```
[30]: #18. Create a violin plot to display the body_mass_g of different species based␣
       ↪on sex
      sns.violinplot(x='species', y='body_mass_g', data = penguins_cleaned, hue='sex'␣
       ↪)
```

```
[30]: <Axes: xlabel='species', ylabel='body_mass_g'>
```


sex
Male
Female
6000
5000
4000
3000
body_mass_g
Adelie
Chinstrap
Gentoo
species

[ ]: