In [1]:
```
import pandas as pd
import numpy as np
import seaborn as sb
import matplotlib.pyplot as plt
```

Reading the dataset

In [3]:
```
df = pd.read_csv('/content/penguins_size.csv')
df
```

Out [3]:

	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
...	...	...	...	...	...	...	...
339	Gentoo	Biscoe	NaN	NaN	NaN	NaN	NaN
340	Gentoo	Biscoe	46.8	14.3	215.0	4850.0	FEMALE
341	Gentoo	Biscoe	50.4	15.7	222.0	5750.0	MALE
342	Gentoo	Biscoe	45.2	14.8	212.0	5200.0	FEMALE
343	Gentoo	Biscoe	49.9	16.1	213.0	5400.0	MALE

344 rows × 7 columns

In [4]:
```
df.shape
```

Out [4]:
```
(344, 7)
```

In [5]:
```
df['species'].value_counts()
```

Out [5]:
```
Adelie       152
Gentoo       124
Chinstrap     68
Name: species, dtype: int64
```

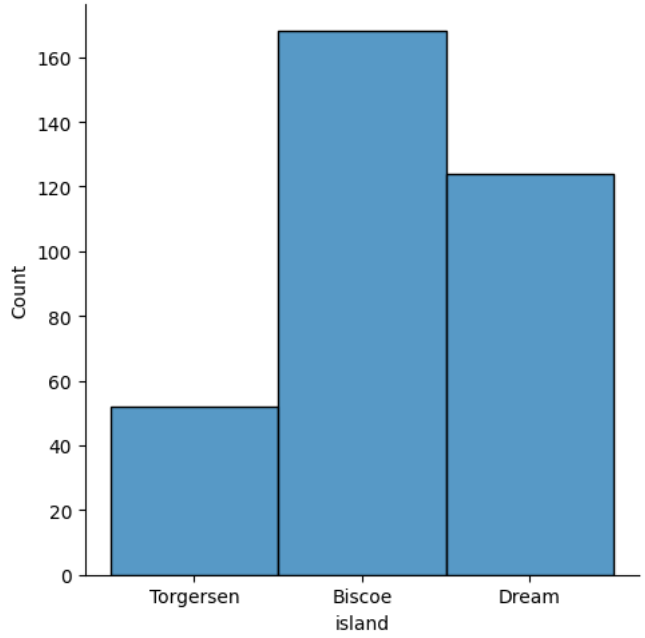
In [6]:
```
df['island'].value_counts()
```

Out [6]:
```
Biscoe       168
Dream        124
Torgersen     52
Name: island, dtype: int64
```

Univariate Analysis

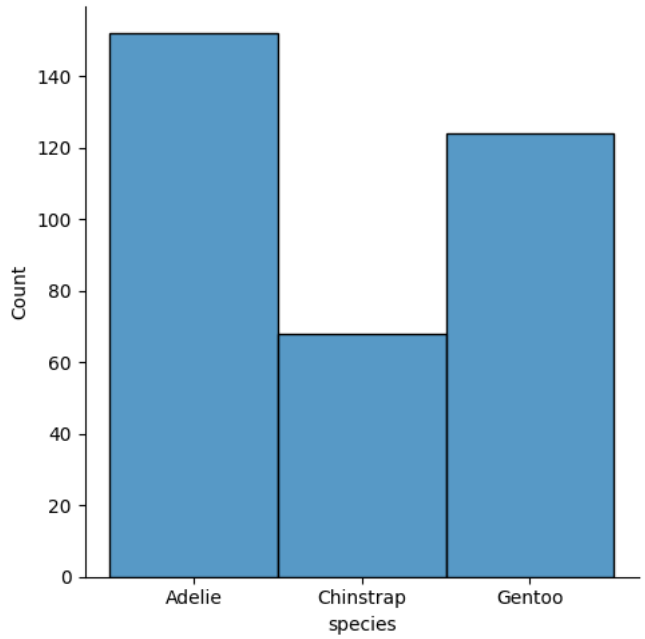
In [7]:
```
sb.displot(df['island'])
```

Out [7]:
```
<seaborn.axisgrid.FacetGrid at 0x7990c2ac22f0>
```

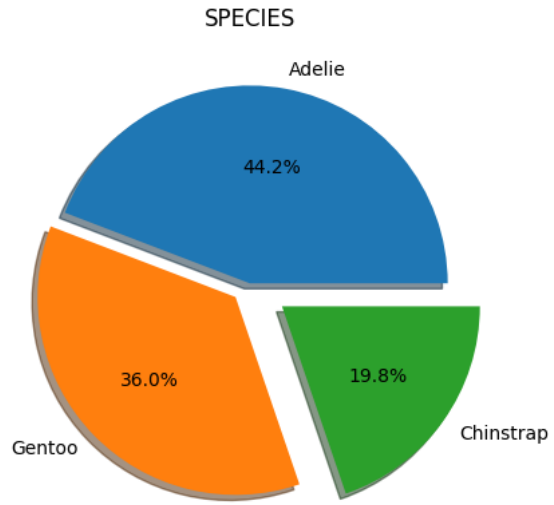


In [9]:
```
sb.displot(df['species'])
```

Out [9]: <seaborn.axisgrid.FacetGrid at 0x7990c2962b00>

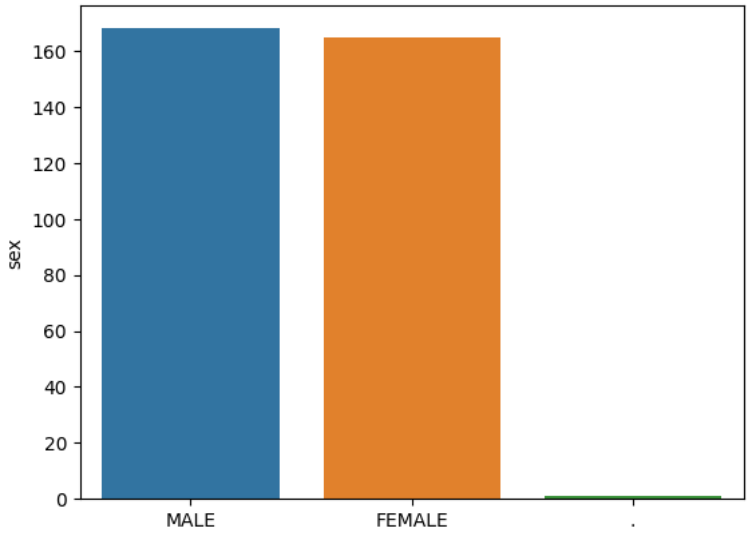


In [19]:
```
plt.pie(df.species.value_counts() ,  [0,0.1,0.2] , labels = [ 'Adelie' ,  'Gentoo','Chinstrap' ] , autopct =
plt.title('SPECIES')
plt.show()
```



In [20]:
```
sb.barplot(x =df.sex.value_counts().index,y =df.sex.value_counts() )
```

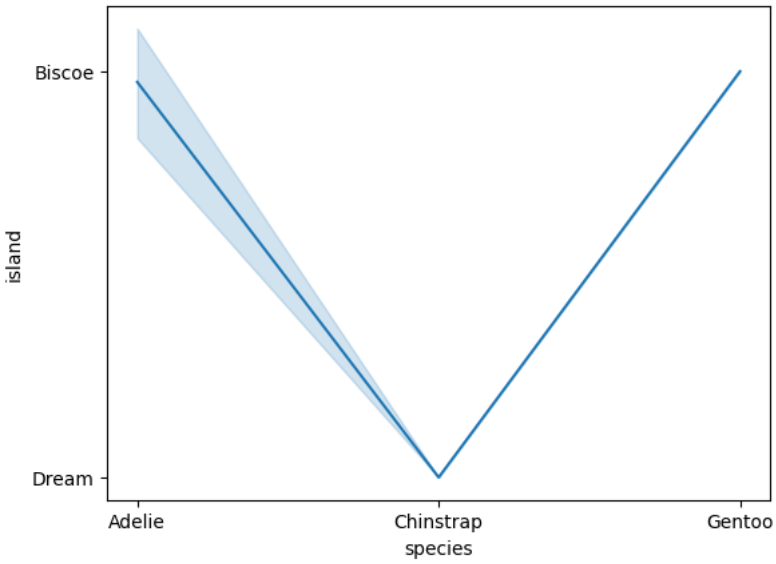
Out [20]: <Axes: ylabel='sex'>



Bi-variate Analysis

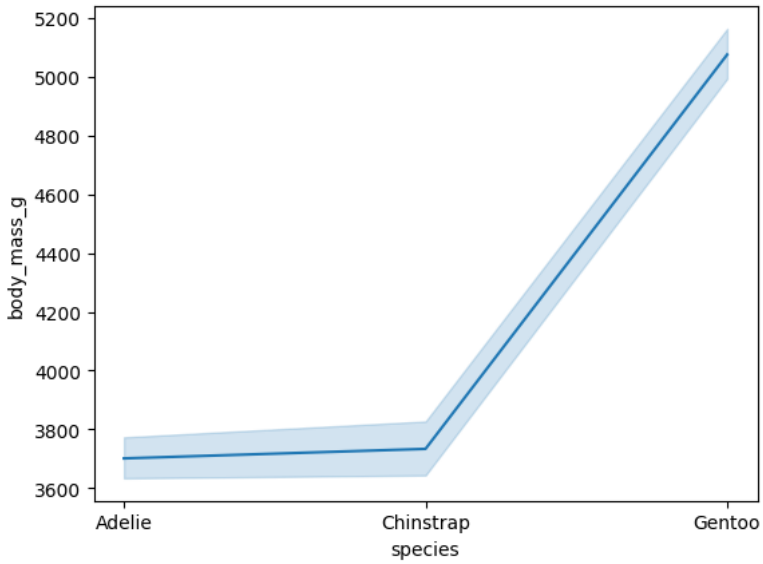
In [22]:
```
sb.lineplot(x = df['species'],y=df['island'])
```

Out [22]: <Axes: xlabel='species', ylabel='island'>



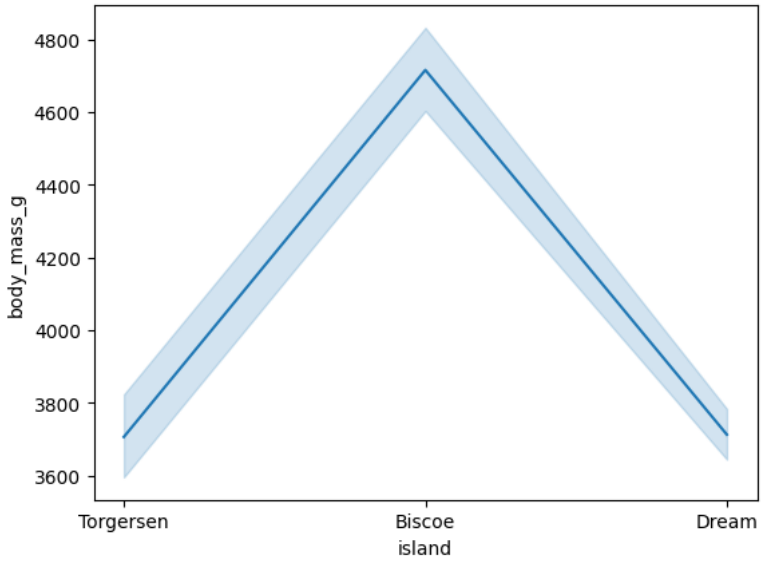
In [23]:
```
sb.lineplot(x = df['species'],y=df['body_mass_g'])
```

Out [23]: <Axes: xlabel='species', ylabel='body_mass_g'>



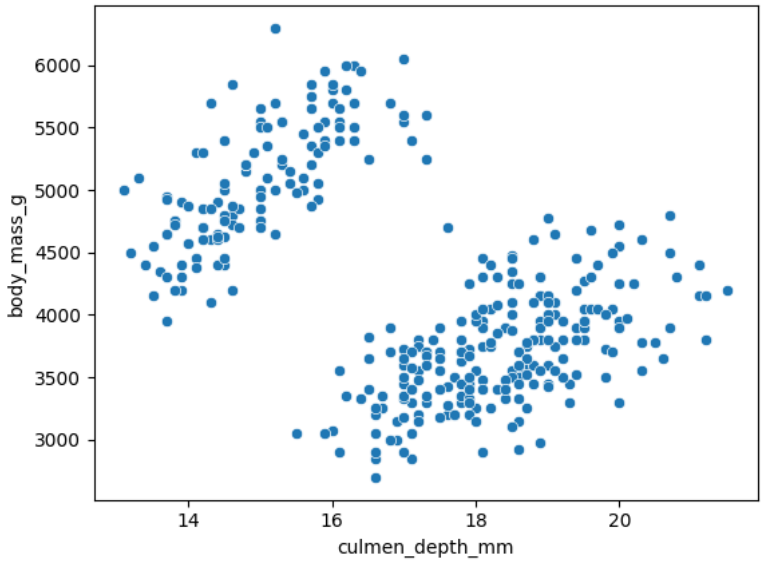
In [26]:
```
sb.lineplot(x = df['island'],y=df['body_mass_g'])
```

Out [26]: <Axes: xlabel='island', ylabel='body_mass_g'>



In [28]:
```
sb.scatterplot(x = df['culmen_depth_mm'],y=df['body_mass_g'])
```

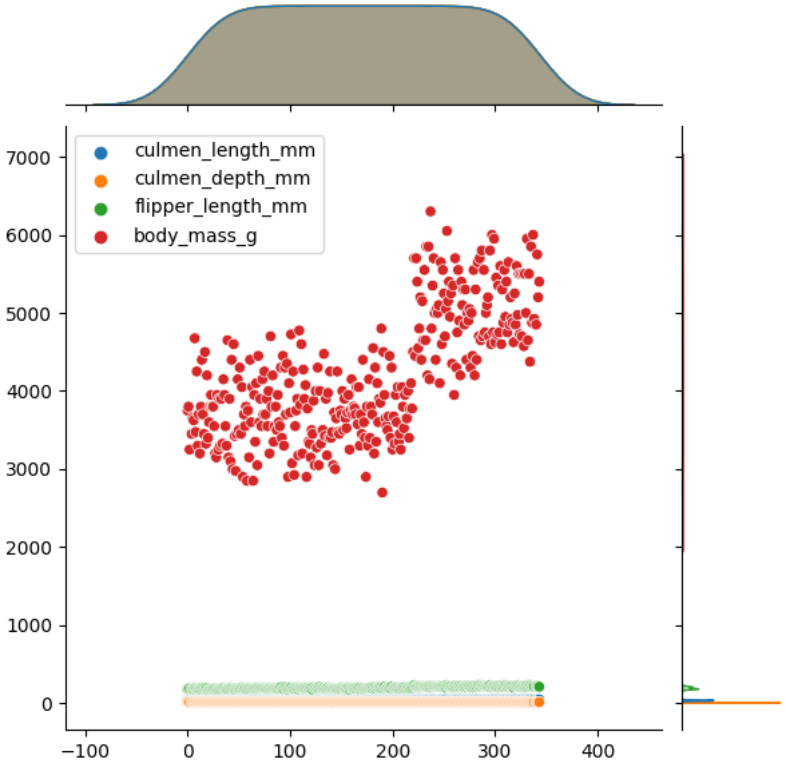
Out [28]: <Axes: xlabel='culmen_depth_mm', ylabel='body_mass_g'>



Multivariate Analysis

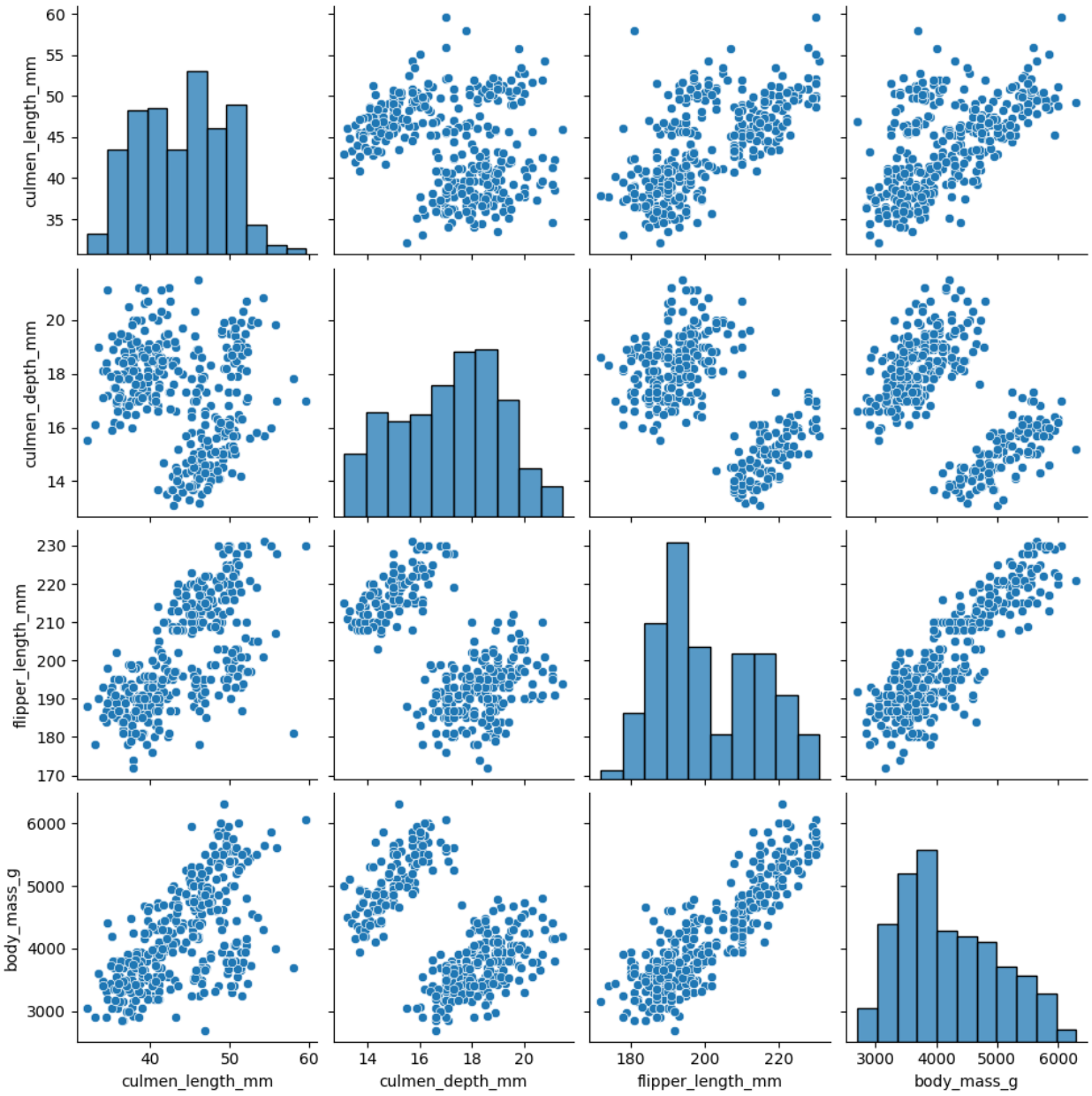
In [34]:
```
sb.jointplot(data=df)
```

Out [34]: <seaborn.axisgrid.JointGrid at 0x7990bf4372e0>



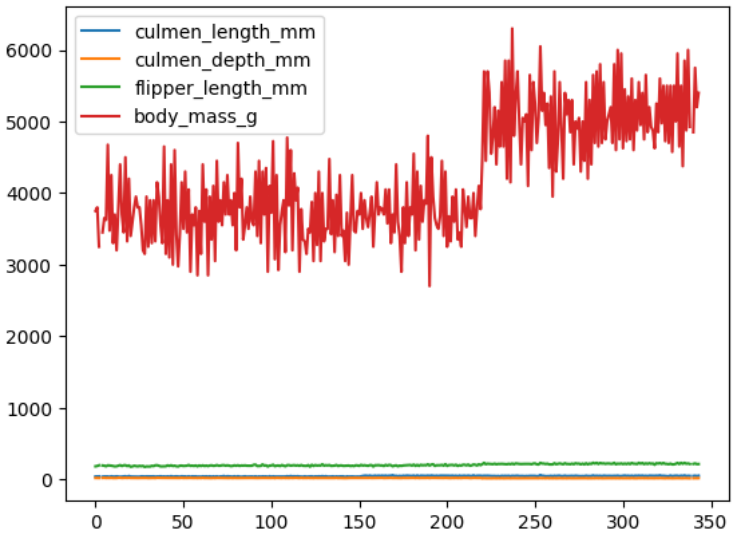
In [29]:
```
sb.pairplot(df)
```

Out [29]: <seaborn.axisgrid.PairGrid at 0x7990bf85a650>



In [30]:
```
df.plot()
```

Out [30]: <Axes: >



Descriptive statistics

In [35]:
```
df.describe()
```

Out [35]:

	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

In [37]:
```
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
```

Check the missing values

In [36]:
```
df.isnull().any()
```

Out [36]:
```
species             False
island              False
culmen_length_mm     True
culmen_depth_mm      True
flipper_length_mm    True
body_mass_g          True
sex                  True
dtype: bool
```

In [38]:
```
df.isnull().sum()
```

Out [38]:
```
species             0
island              0
culmen_length_mm    2
culmen_depth_mm     2
flipper_length_mm   2
body_mass_g         2
sex                10
dtype: int64
```

Label Encoder

In [41]:
```
from sklearn.preprocessing  import LabelEncoder
```

In [42]:
```
le = LabelEncoder()
```

In [44]:
```
df['sex'] = le.fit_transform(df['sex'])
df
```

Out [44]:

	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	2
1	Adelie	Torgersen	39.50000	17.40000	186.000000	3800.000000	1
2	Adelie	Torgersen	40.30000	18.00000	195.000000	3250.000000	1
3	Adelie	Torgersen	43.92193	17.15117	200.915205	4201.754386	3
4	Adelie	Torgersen	36.70000	19.30000	193.000000	3450.000000	1
...	...	...	...	...	...	...	...
339	Gentoo	Biscoe	43.92193	17.15117	200.915205	4201.754386	3
340	Gentoo	Biscoe	46.80000	14.30000	215.000000	4850.000000	1
341	Gentoo	Biscoe	50.40000	15.70000	222.000000	5750.000000	2
342	Gentoo	Biscoe	45.20000	14.80000	212.000000	5200.000000	1
343	Gentoo	Biscoe	49.90000	16.10000	213.000000	5400.000000	2

344 rows × 7 columns

In [45]:
```
df.fillna(df.mean() , inplace= True)
df
```

```
<ipython-input-45-826902893166>:1: FutureWarning: The default value of numeric_only in DataFrame.mean 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.fillna(df.mean() , inplace= True)
```

Out [45]:

	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	2
1	Adelie	Torgersen	39.50000	17.40000	186.000000	3800.000000	1
2	Adelie	Torgersen	40.30000	18.00000	195.000000	3250.000000	1
3	Adelie	Torgersen	43.92193	17.15117	200.915205	4201.754386	3
4	Adelie	Torgersen	36.70000	19.30000	193.000000	3450.000000	1
...	...	...	...	...	...	...	...
339	Gentoo	Biscoe	43.92193	17.15117	200.915205	4201.754386	3
340	Gentoo	Biscoe	46.80000	14.30000	215.000000	4850.000000	1
341	Gentoo	Biscoe	50.40000	15.70000	222.000000	5750.000000	2
342	Gentoo	Biscoe	45.20000	14.80000	212.000000	5200.000000	1
343	Gentoo	Biscoe	49.90000	16.10000	213.000000	5400.000000	2

344 rows × 7 columns

one hot encoding

In [47]:
```
df_main = pd.get_dummies(df,columns =['species' , 'island'  ])
df_main.head()
```

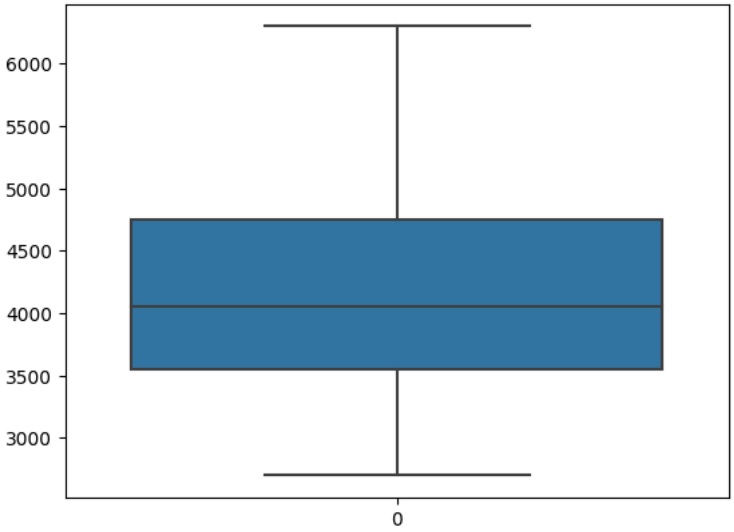
Out [47]:

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

outliers

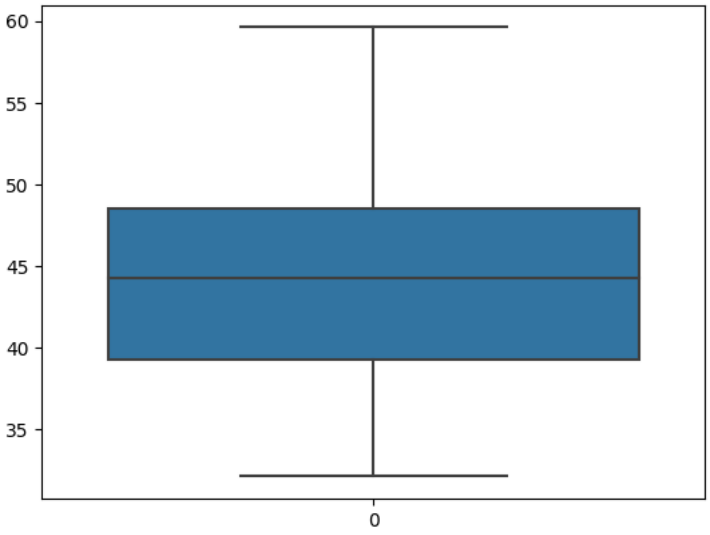
In [49]:
```
sb.boxplot(df['body_mass_g'])
```

Out [49]: <Axes: >



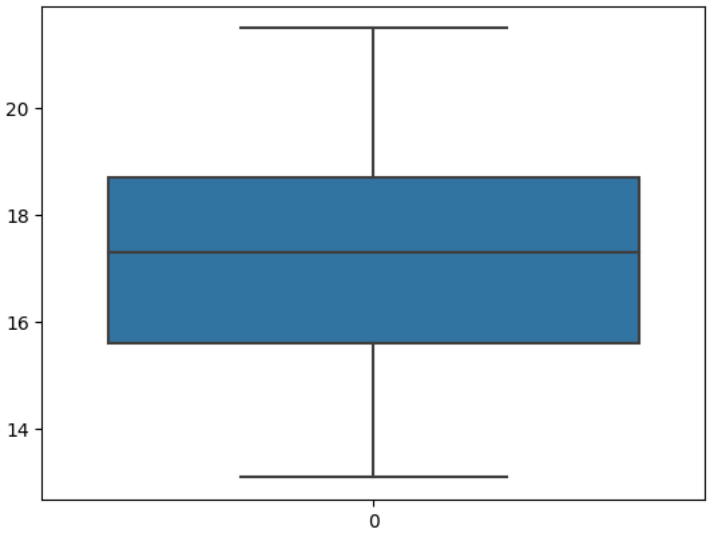
In [50]:
```
sb.boxplot(df['culmen_length_mm'])
```

Out [50]: <Axes: >



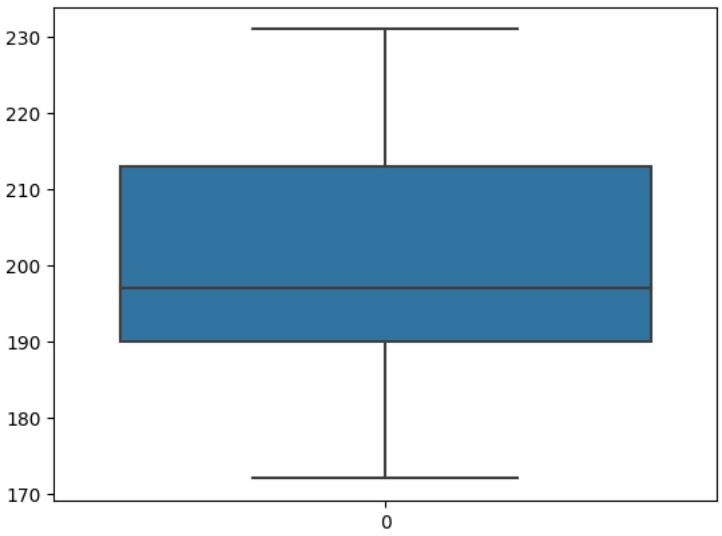
In [51]:
```
sb.boxplot(df['culmen_depth_mm'])
```

Out [51]: <Axes: >



In [52]:
```
sb.boxplot(df['flipper_length_mm'])
```

Out [52]: <Axes: >



correlation of independent variables

In [53]:
```
df_main.corr()
```

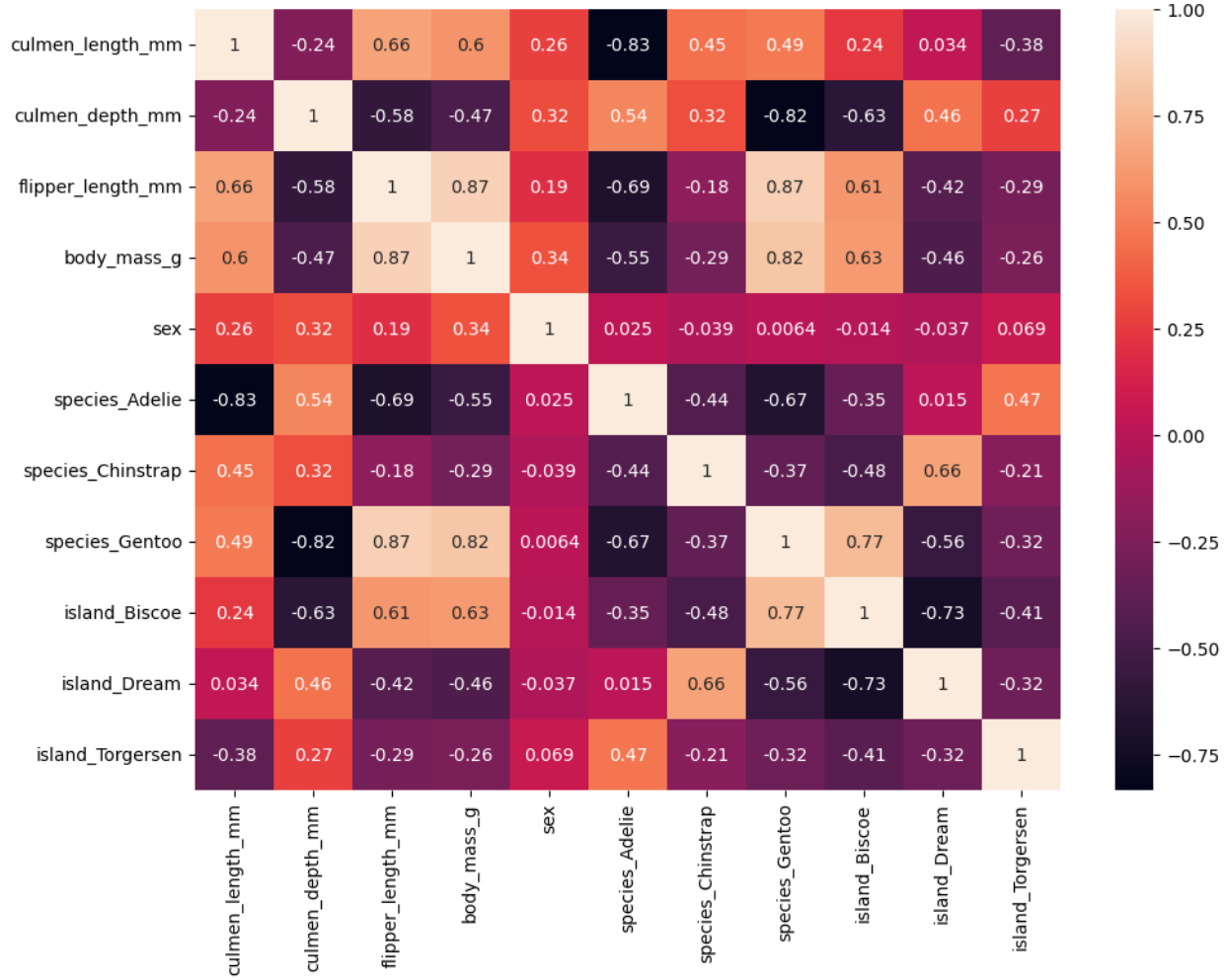
Out [53]:

	culmen_length_mm	culmen_depth_mm	flipper_length_mm	body_mass_g	sex	species_Adelie	species_Chinstra

	culmen_length_mm	culmen_depth_mm	flipper_length_mm	body_mass_g	sex	species_Adelie	species_Chinstra
culmen_length_mm	1.000000	-0.235053	0.656181	0.595110	0.264024	-0.834277	0.448530
culmen_depth_mm	-0.235053	1.000000	-0.583851	-0.471916	0.316379	0.537305	0.320468
flipper_length_mm	0.656181	-0.583851	1.000000	0.871202	0.193476	-0.692055	-0.180520
body_mass_g	0.595110	-0.471916	0.871202	1.000000	0.340402	-0.554721	-0.291351
sex	0.264024	0.316379	0.193476	0.340402	1.000000	0.024857	-0.038745
species_Adelie	-0.834277	0.537305	-0.692055	-0.554721	0.024857	1.000000	-0.441643
species_Chinstrap	0.448530	0.320468	-0.180520	-0.291351	-0.038745	-0.441643	1.000000
species_Gentoo	0.490869	-0.821550	0.865530	0.815411	0.006427	-0.667991	-0.372649
island_Biscoe	0.238622	-0.630442	0.609855	0.625523	-0.013800	-0.354038	-0.484951
island_Dream	0.033950	0.455604	-0.420557	-0.459651	-0.036926	0.014743	0.661151
island_Torgersen	-0.378494	0.269073	-0.287321	-0.256785	0.068753	0.474285	-0.209464

In [55]:
```
plt.figure(figsize=(11,8))
sb.heatmap(df_main.corr(),annot =True)
```

Out [55]: <Axes: >



Split the data into dependent and independent variables.

In [56]:
```
df_main.head()
```

Out [56]:

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

In [58]:
```
y  = df_main['body_mass_g']
y
```

Out [58]:
```
0      3750.000000
1      3800.000000
2      3250.000000
3      4201.754386
4      3450.000000
          ...
339    4201.754386
340    4850.000000
341    5750.000000
342    5200.000000
343    5400.000000
Name: body_mass_g, Length: 344, dtype: float64
```

In [59]:
```
x=df_main.drop(columns  = ['body_mass_g'] , axis=1)
x
```

Out [59]:

	culmen_length_mm	culmen_depth_mm	flipper_length_mm	sex	species_Adelie	species_Chinstrap	species_Gentoo	island_Biscoe
0	39.10000	18.70000	181.000000	2	1	0	0	0
1	39.50000	17.40000	186.000000	1	1	0	0	0
2	40.30000	18.00000	195.000000	1	1	0	0	0
3	43.92193	17.15117	200.915205	3	1	0	0	0
4	36.70000	19.30000	193.000000	1	1	0	0	0
...	...	...	...	...	...	...	...	...
339	43.92193	17.15117	200.915205	3	0	0	1	1
340	46.80000	14.30000	215.000000	1	0	0	1	1
341	50.40000	15.70000	222.000000	2	0	0	1	1
342	45.20000	14.80000	212.000000	1	0	0	1	1
343	49.90000	16.10000	213.000000	2	0	0	1	1

344 rows × 10 columns

Scaling the data

In [60]:
```
from sklearn.preprocessing import MinMaxScaler
scale =MinMaxScaler()
```

In [61]:
```
X_scaled= pd.DataFrame(scale.fit_transform(x),columns =x.columns)
X_scaled.head()
```

Out [61]:

	culmen_length_mm	culmen_depth_mm	flipper_length_mm	sex	species_Adelie	species_Chinstrap	species_Gentoo	island_Biscoe
0	0.254545	0.666667	0.152542	0.666667	1.0	0.0	0.0	0.0
1	0.269091	0.511905	0.237288	0.333333	1.0	0.0	0.0	0.0
2	0.298182	0.583333	0.389831	0.333333	1.0	0.0	0.0	0.0
3	0.429888	0.482282	0.490088	1.000000	1.0	0.0	0.0	0.0
4	0.167273	0.738095	0.355932	0.333333	1.0	0.0	0.0	0.0

Split the data into training and testing

In [62]:
```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test = train_test_split(X_scaled,y,test_size=0.3,random_state=10)
```

Check the training and testing data shape

In [63]:
```
X_train.shape
```

Out [63]:
```
(240, 10)
```

In [64]:
```
X_train.head()
```

Out [64]:

	culmen_length_mm	culmen_depth_mm	flipper_length_mm	sex	species_Adelie	species_Chinstrap	species_Gentoo	island_Bis
258	0.432727	0.059524	0.610169	0.333333	0.0	0.0	1.0	1.0
332	0.414545	0.250000	0.694915	0.333333	0.0	0.0	1.0	1.0
121	0.203636	0.797619	0.440678	0.666667	1.0	0.0	0.0	0.0
61	0.334545	0.952381	0.389831	0.666667	1.0	0.0	0.0	1.0
70	0.050909	0.702381	0.305085	0.333333	1.0	0.0	0.0	0.0

In [67]:
```
y_train.shape
```

Out [67]: (240,)

In [68]:
```
y_train.head()
```

Out [68]:
```
258    4350.0
332    4650.0
121    3500.0
61     4400.0
70     3600.0
Name: body_mass_g, dtype: float64
```

In [65]:
```
X_test.shape
```

Out [65]: (104, 10)

In [66]:
```
X_test.head()
```

Out [66]:

	culmen_length_mm	culmen_depth_mm	flipper_length_mm	sex	species_Adelie	species_Chinstrap	species_Gentoo	island_Bis
229	0.534545	0.273810	0.728814	0.666667	0.0	0.0	1.0	1.0
80	0.090909	0.488095	0.288136	0.333333	1.0	0.0	0.0	0.0
327	0.774545	0.321429	0.796610	0.666667	0.0	0.0	1.0	1.0
6	0.247273	0.559524	0.152542	0.333333	1.0	0.0	0.0	0.0
309	0.727273	0.464286	0.983051	0.666667	0.0	0.0	1.0	1.0

In [69]:
```
y_test.shape
```

Out [69]: (104,)

In [70]:
```
y_test.head()
```

Out [70]:
```
229    5150.0
80     3200.0
327    5500.0
6      3625.0
309    5550.0
Name: body_mass_g, dtype: float64
```