Zybooks includes a case study on the Palmer Penguin dataset. Be sure to complete the interactive reading in your Zybooks before completing this notebook.

Steps:

1. Complete the Palmer Penguin Case Study (Interactive) in your Zybooks.
2. Follow the Instructions in this notebook. Click on execution arrow to the left of the code cells to execute code.
3. Questions that you need to answer will appear in a markdown or text cell. Place your answer in the cell (double click the cell to open).
4. Questions that require code will have a code cell immediately below the markdown or text cell. Enter and execute your code in the code cell, adding additional blocks for code if needed. Draw on the knowledge you have gained in Datacamp and in Zybooks to complete the code.
5. Save your work in your Google Drive (File . . . Save a copy to Drive) or you can save the notebook (File . . . Download .ipynb). Notebooks have the extension .ipynb, just the python code without the markdown can be saved as a python file with the extension .py but you will lose the markdown.
6. TURN IN A PDF: Generate a PDF by selecting File . . . Print . . . and change the destination to .PDF.

NOTE: students can experiment with generating code with AI, a feature provided in Google Colab. Be careful! You need to be able to verify the code that is generated as it is not always accurate! Be sure to leave in the documentation that shows that the code was generated.

Reference:

https://pypi.org/project/palmerpenguins/

https://github.com/allisonhorst/palmerpenguins

Pandas for Python Cheat Sheet:

https://pandas.pydata.org/Pandas_Cheat_Sheet.pdf

Import Necessary Libraries

In [ ]:

```
%matplotlib inline
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
import warnings
warnings.filterwarnings('ignore')
sns.set()
sns.set_style('whitegrid')
```

Load the Penguins Data

In [ ]:

```
penguins = pd.read_csv('https://raw.githubusercontent.com/rfordatascience/tidytuesday/
```

In [ ]:

```
# view the shape
penguins.shape
```

Out[ ]:

```
(344, 8)
```

In [ ]:

```
# write the file to csv
# click on the folder in the left sidebar to see the file
# select the three dots to download the file locally
penguins.to_csv('penguins2.csv')
```

In [ ]:

```
# view the first 10 rows
penguins.head(10)
```

Out[ ]:

	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex	year
0	Adelie	Torgersen	39.1	18.7	181.0	3750.0	male	2007
1	Adelie	Torgersen	39.5	17.4	186.0	3800.0	female	2007
2	Adelie	Torgersen	40.3	18.0	195.0	3250.0	female	2007
3	Adelie	Torgersen	NaN	NaN	NaN	NaN	NaN	2007
4	Adelie	Torgersen	36.7	19.3	193.0	3450.0	female	2007
5	Adelie	Torgersen	39.3	20.6	190.0	3650.0	male	2007
6	Adelie	Torgersen	38.9	17.8	181.0	3625.0	female	2007
7	Adelie	Torgersen	39.2	19.6	195.0	4675.0	male	2007
8	Adelie	Torgersen	34.1	18.1	193.0	3475.0	NaN	2007
9	Adelie	Torgersen	42.0	20.2	190.0	4250.0	NaN	2007

In [ ]:

```
# use value counts to count the number of rows with each unique value
penguins.species.value_counts()
```

Out[ ]:

```
species
Adelie       152
Gentoo       124
Chinstrap     68
Name: count, dtype: int64
```

Question 1:

Use value counts to count the number of rows with unique values for the "island" column.

In [ ]:

```
#Question 1 code
penguins["island"].value_counts()
```

Out[ ]:

```
island
Biscoe       168
Dream        124
Torgersen     52
Name: count, dtype: int64
```

In [ ]:

```
# count missing values
print(penguins.isna().sum())
```

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

In [ ]:

```
# use info to count missing values
penguins.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 344 entries, 0 to 343
Data columns (total 8 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 
 7   year               344 non-null    int64  
dtypes: float64(4), int64(1), object(3)
memory usage: 21.6+ KB
```

In [ ]:

```
# use describe to get basic statistical information on the dataframe
penguins.describe()
```

Out[ ]:

	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	year
count	342.000000	342.000000	342.000000	342.000000	344.000000
mean	43.921930	17.151170	200.915205	4201.754386	2008.029070
std	5.459584	1.974793	14.061714	801.954536	0.818356
min	32.100000	13.100000	172.000000	2700.000000	2007.000000
25%	39.225000	15.600000	190.000000	3550.000000	2007.000000
50%	44.450000	17.300000	197.000000	4050.000000	2008.000000
75%	48.500000	18.700000	213.000000	4750.000000	2009.000000
max	59.600000	21.500000	231.000000	6300.000000	2009.000000

In [ ]:

```
# select a subset of the dataframe
island_sex = penguins[["island", "sex"]]
island_sex.head()
```

Out[ ]:

	island	sex
0	Torgersen	male
1	Torgersen	female
2	Torgersen	female
3	Torgersen	NaN
4	Torgersen	female

In [ ]:

```
# select rows 3 and 4, just the bill_length_mm and bill_depth_mm columns
penguins[['bill_length_mm','bill_depth_mm']][3:5]
```

Out[ ]:

	bill_length_mm	bill_depth_mm
3	NaN	NaN
4	36.7	19.3

Question 2:

Enter code below to select rows 10, 11 and 12, just the island and sex.

In [ ]:

```
# Question 2 Code
penguins[['island', 'sex']][10:13]
```

Out[ ]:

	island	sex
10	Torgersen	NaN
11	Torgersen	NaN
12	Torgersen	female

In [ ]:

```python
# Filter records based on a condition
penguins[penguins['body_mass_g'] > 6000]
```

Out[ ]:

	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex	year
169	Gentoo	Biscoe	49.2	15.2	221.0	6300.0	male	2007
185	Gentoo	Biscoe	59.6	17.0	230.0	6050.0	male	2007

Question 3:

Enter code below to filter just the rows where island is equal to Biscoe.

In [ ]:

```python
# Question 3 code
penguins[penguins['island'] == 'Biscoe']
```

Out[ ]:

	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex	year
20	Adelie	Biscoe	37.8	18.3	174.0	3400.0	female	2007
21	Adelie	Biscoe	37.7	18.7	180.0	3600.0	male	2007
22	Adelie	Biscoe	35.9	19.2	189.0	3800.0	female	2007
23	Adelie	Biscoe	38.2	18.1	185.0	3950.0	male	2007
24	Adelie	Biscoe	38.8	17.2	180.0	3800.0	male	2007
...	...	...	...	...	...	...	...	...
271	Gentoo	Biscoe	NaN	NaN	NaN	NaN	NaN	2009
272	Gentoo	Biscoe	46.8	14.3	215.0	4850.0	female	2009
273	Gentoo	Biscoe	50.4	15.7	222.0	5750.0	male	2009
274	Gentoo	Biscoe	45.2	14.8	212.0	5200.0	female	2009
275	Gentoo	Biscoe	49.9	16.1	213.0	5400.0	male	2009

168 rows × 8 columns

In [ ]:

```python
# Filter with && and == operators
bodymass = penguins["body_mass_g"] < 3400
```

```
sexm = penguins["sex"] == "male"
penguins[bodymass & sexm]
```

Out[ ]:

	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex	yea
119	Adelie	Torgersen	41.1	18.6	189.0	3325.0	male	200
292	Chinstrap	Dream	50.3	20.0	197.0	3300.0	male	200
324	Chinstrap	Dream	51.5	18.7	187.0	3250.0	male	200

In [ ]:

```
# What percentage of penguins are on Island "Dream"?
penguins['island'].value_counts(normalize=True)*100
```

Out[ ]:

```
island
Biscoe       48.837209
Dream        36.046512
Torgersen    15.116279
Name: proportion, dtype: float64
```

In [ ]:

```
# use "Group by" to get the mean flipper_length_mm by sex and species
penguins.groupby(["sex", "species"])["flipper_length_mm"].mean()
```

Out[ ]:

```
sex     species
female  Adelie       187.794521
        Chinstrap    191.735294
        Gentoo       212.706897
male    Adelie       192.410959
        Chinstrap    199.911765
        Gentoo       221.540984
Name: flipper_length_mm, dtype: float64
```

Question 4:

Enter the code below to use "Group by" to get the mean bill_length_mm by Island and species

In [ ]:

```
# Question 4 Code
penguins.groupby(["island", 'species'])["bill_length_mm"].mean()
```

Out[ ]:

```
island     species
Biscoe     Adelie       38.975000
           Gentoo       47.504878
Dream      Adelie       38.501786
           Chinstrap    48.833824
Torgersen  Adelie       38.950980
Name: bill_length_mm, dtype: float64
```

In [ ]:

```
# use Group By with describe
penguins.groupby(['sex','island']).describe()
```

Out[ ]:

		bill_length_mm								bill_depth_mm		...
		count	mean	std	min	25%	50%	75%	max	count	mean	...
sex	**island**											
female	**Biscoe**	80.0	43.307500	4.177631	34.5	39.675	44.9	46.500	50.5	80.0	15.191250	...
	Dream	61.0	42.296721	5.533834	32.1	37.000	42.5	46.400	58.0	61.0	17.601639	...
	Torgersen	24.0	37.554167	2.207887	33.5	35.850	37.6	39.125	41.1	24.0	17.550000	...
male	**Biscoe**	83.0	47.119277	4.691000	37.6	43.800	48.5	50.050	59.6	83.0	16.597590	...
	Dream	62.0	46.116129	5.767211	36.3	40.625	49.1	51.225	55.8	62.0	19.066129	...
	Torgersen	23.0	40.586957	3.027496	34.6	38.850	41.1	42.650	46.0	23.0	19.391304	...

6 rows × 40 columns

Question 5:

Place the code in the cell below to Use group by with describe to gain insight on the year and island

In [ ]:

```python
# Question 5 code
penguins.groupby(['year','island']).describe()
```

Out[ ]:

		bill_length_mm								bill_depth_mm		...
		count	mean	std	min	25%	50%	75%	max	count	mean	...
year	**island**											
2007	**Biscoe**	44.0	45.038636	4.746515	35.3	41.725	46.10	48.250	59.6	44.0	15.540909	...
	Dream	46.0	44.539130	5.677225	36.0	39.525	45.30	49.800	58.0	46.0	18.573913	...
	Torgersen	19.0	38.800000	2.931628	34.1	37.250	38.90	39.900	46.0	19.0	19.021053	...
2008	**Biscoe**	64.0	44.620312	4.551789	34.5	41.550	45.25	48.250	54.3	64.0	15.825000	...
	Dream	34.0	43.755882	6.215094	33.1	38.450	42.85	49.375	54.2	34.0	18.397059	...
	Torgersen	16.0	38.768750	3.651432	33.5	35.800	38.40	41.875	45.8	16.0	18.118750	...
2009	**Biscoe**	59.0	46.111864	4.975980	35.0	42.950	47.20	49.850	55.9	59.0	16.177966	...
	Dream	44.0	44.097727	6.142210	32.1	38.775	44.35	50.125	55.8	44.0	18.063636	...
	Torgersen	16.0	39.312500	2.580407	35.2	37.600	38.90	41.175	44.1	16.0	18.037500	...

9 rows × 32 columns

DATA VISUALIZATION

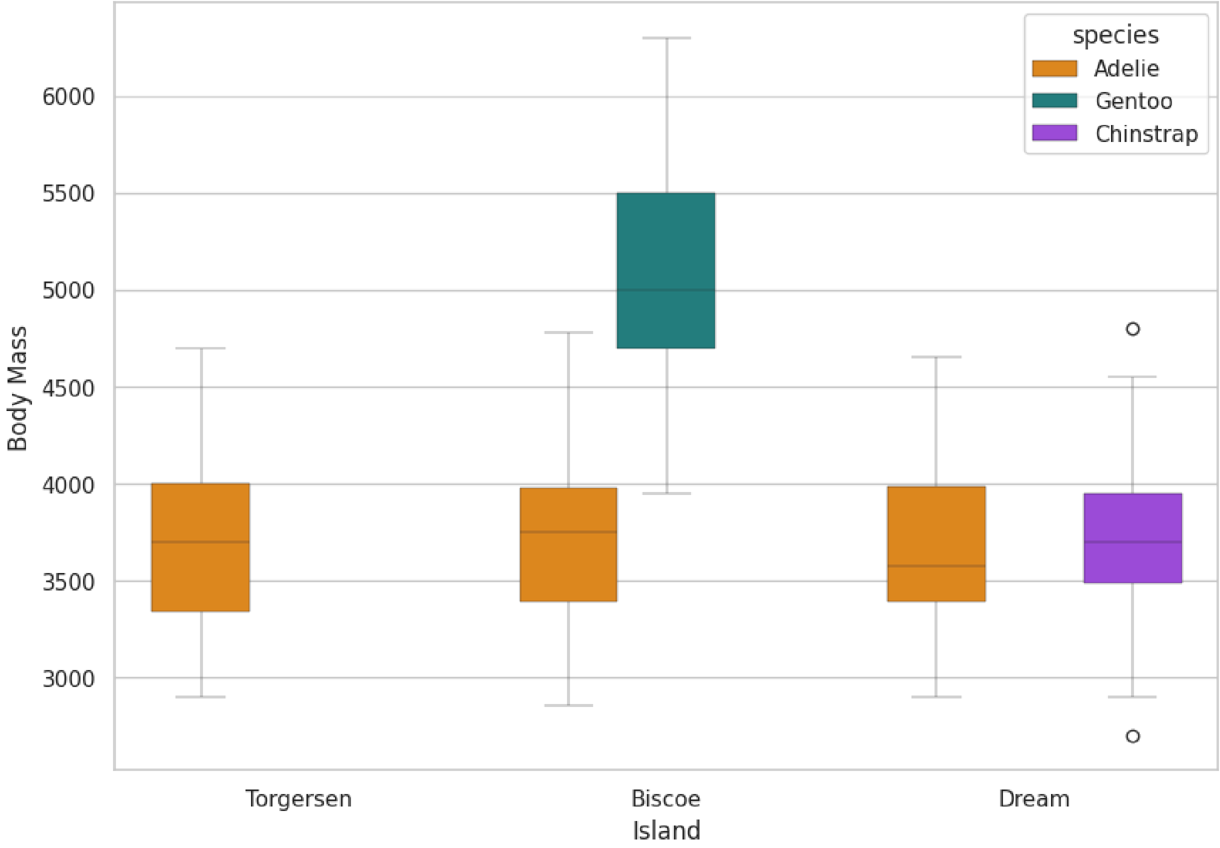
Examine the examples below for data visualization of the penguins data. Review examples in DataCamp as well. Question 6 will ask you to generate your own interesting data visualizations for the penguin data.

In [ ]:

```
plt.figure(figsize = [10,7])
g = sns.boxplot(x = 'island',
            y ='body_mass_g',
            hue = 'species',
            data = penguins,
            palette=['#FF8C00','#159090','#A034F0'],
            linewidth=0.3)
g.set_xlabel('Island')
g.set_ylabel('Body Mass')
```

Out[ ]:

```
Text(0, 0.5, 'Body Mass')
```

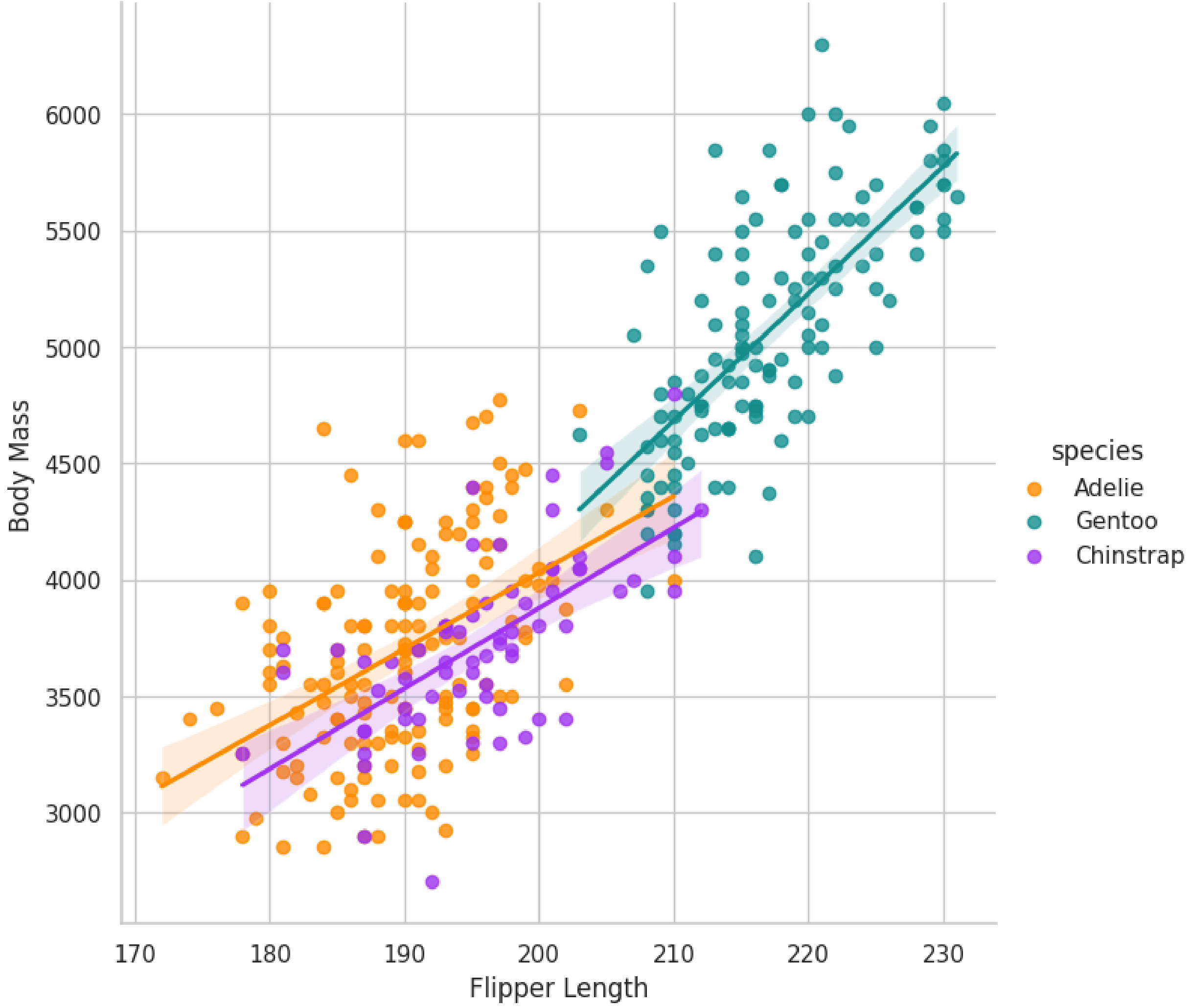


In [ ]:

```
g = sns.lmplot(x="flipper_length_mm",
               y="body_mass_g",
               hue="species",
               height=7,
               data=penguins,
               palette=['#FF8C00','#159090','#A034F0'])
g.set_xlabels('Flipper Length')
g.set_ylabels('Body Mass')
```

Out[ ]:

```
<seaborn.axisgrid.FacetGrid at 0x7fb157b1da80>
```



```
In [ ]: # heat map of the penguins data
        sns.heatmap(penguins.corr(), annot=True)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-84-0a0b56e36a58> in <cell line: 2>()
      1 # heat map of the penguins data
----> 2 sns.heatmap(penguins.corr(), annot=True)

/usr/local/lib/python3.10/dist-packages/pandas/core/frame.py in corr(self, method, mi
n_periods, numeric_only)
  10052         cols = data.columns
  10053         idx = cols.copy()
> 10054         mat = data.to_numpy(dtype=float, na_value=np.nan, copy=False)
  10055 
  10056         if method == "pearson":

/usr/local/lib/python3.10/dist-packages/pandas/core/frame.py in to_numpy(self, dtype,
copy, na_value)
   1836         if dtype is not None:
   1837             dtype = np.dtype(dtype)
-> 1838         result = self._mgr.as_array(dtype=dtype, copy=copy, na_value=na_valu
e)
   1839         if result.dtype is not dtype:
   1840             result = np.array(result, dtype=dtype, copy=False)

/usr/local/lib/python3.10/dist-packages/pandas/core/internals/managers.py in as_array
(self, dtype, copy, na_value)
   1730                 arr.flags.writeable = False
   1731         else:
-> 1732             arr = self._interleave(dtype=dtype, na_value=na_value)
   1733             # The underlying data was copied within _interleave, so no need
   1734             # to further copy if copy=True or setting na_value

/usr/local/lib/python3.10/dist-packages/pandas/core/internals/managers.py in _interle
ave(self, dtype, na_value)
   1792             else:
   1793                 arr = blk.get_values(dtype)
-> 1794             result[rl.indexer] = arr
   1795             itemmask[rl.indexer] = 1
   1796 

ValueError: could not convert string to float: 'Adelie'
```

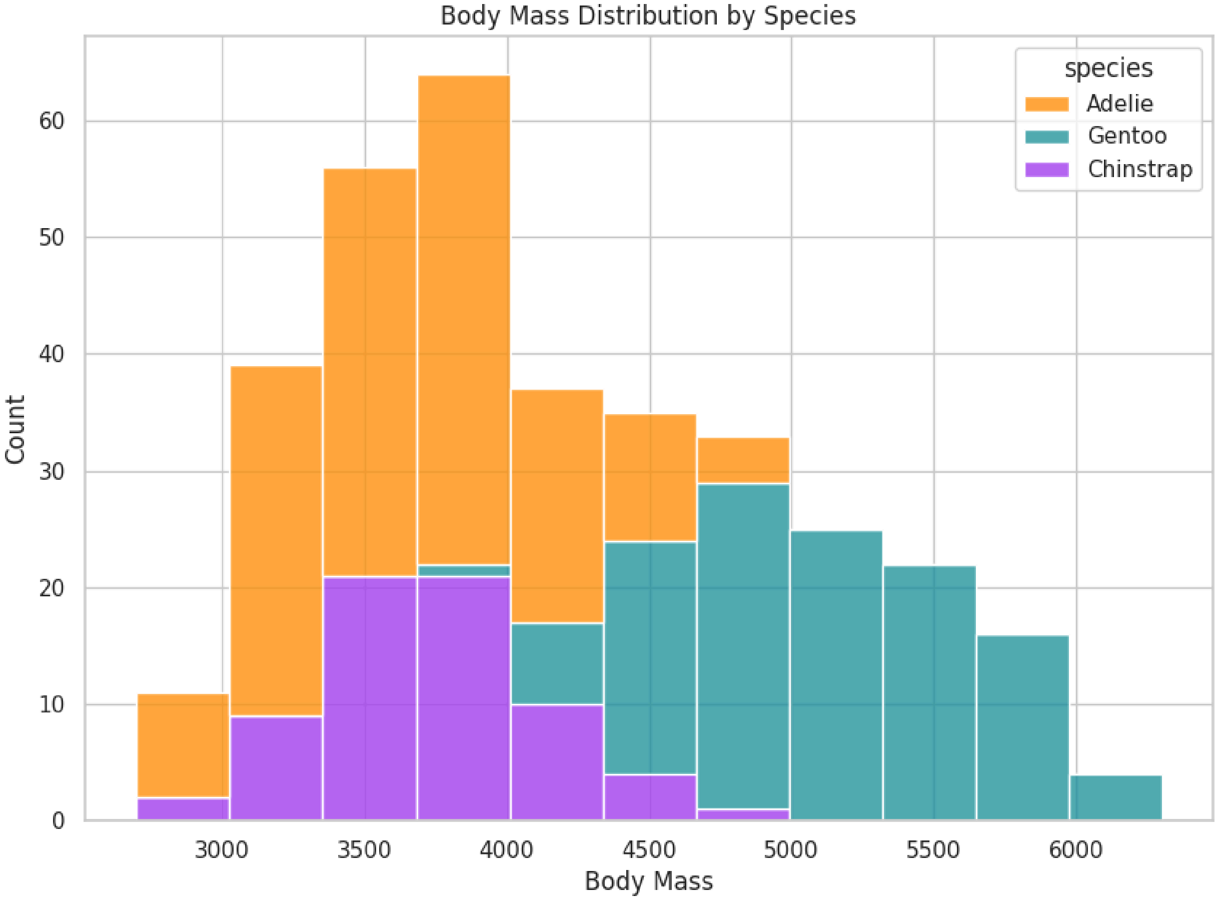
Question 6:

Using the code cells below labed Visualization 1 through Visualization 5, create 5 additional visualizations for the Penguin data. The final question asks for a summary of the findings for the Penguins data based on the exporation in this notebook and your visualizations. Present five findings in narrative form, for example, "Based on body mass and flipper length, Adelie and Chinstrap are similar, where Gentoo tends to have a larger body mass and flipper length."

In [ ]:

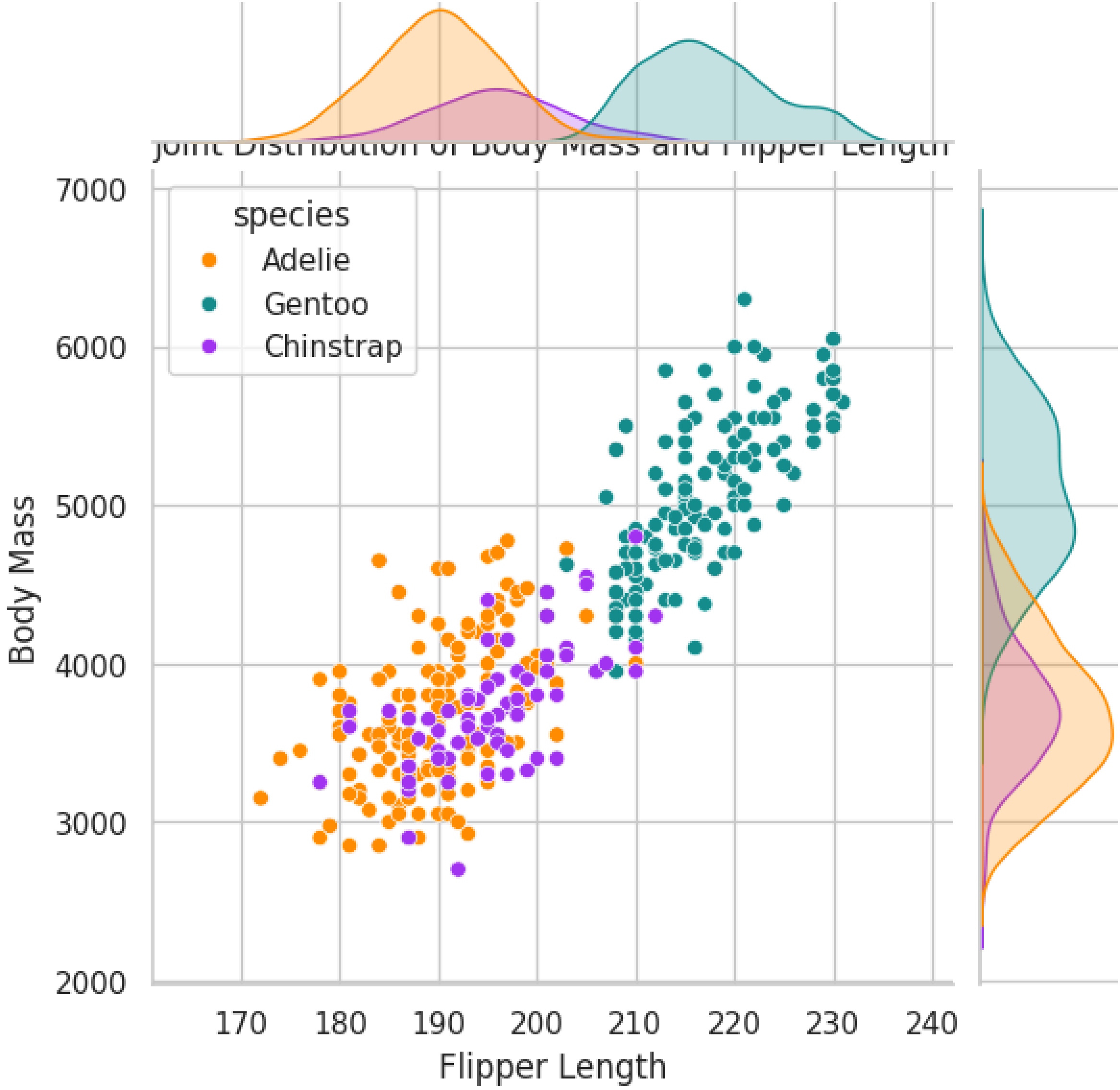
```
# Question 6 Visualization 1

plt.figure(figsize=(10,7))
sns.histplot(data=penguins, x='body_mass_g', hue='species', multiple='stack', palette=
plt.xlabel('Body Mass')
plt.ylabel('Count')
plt.title('Body Mass Distribution by Species')
plt.show()
```



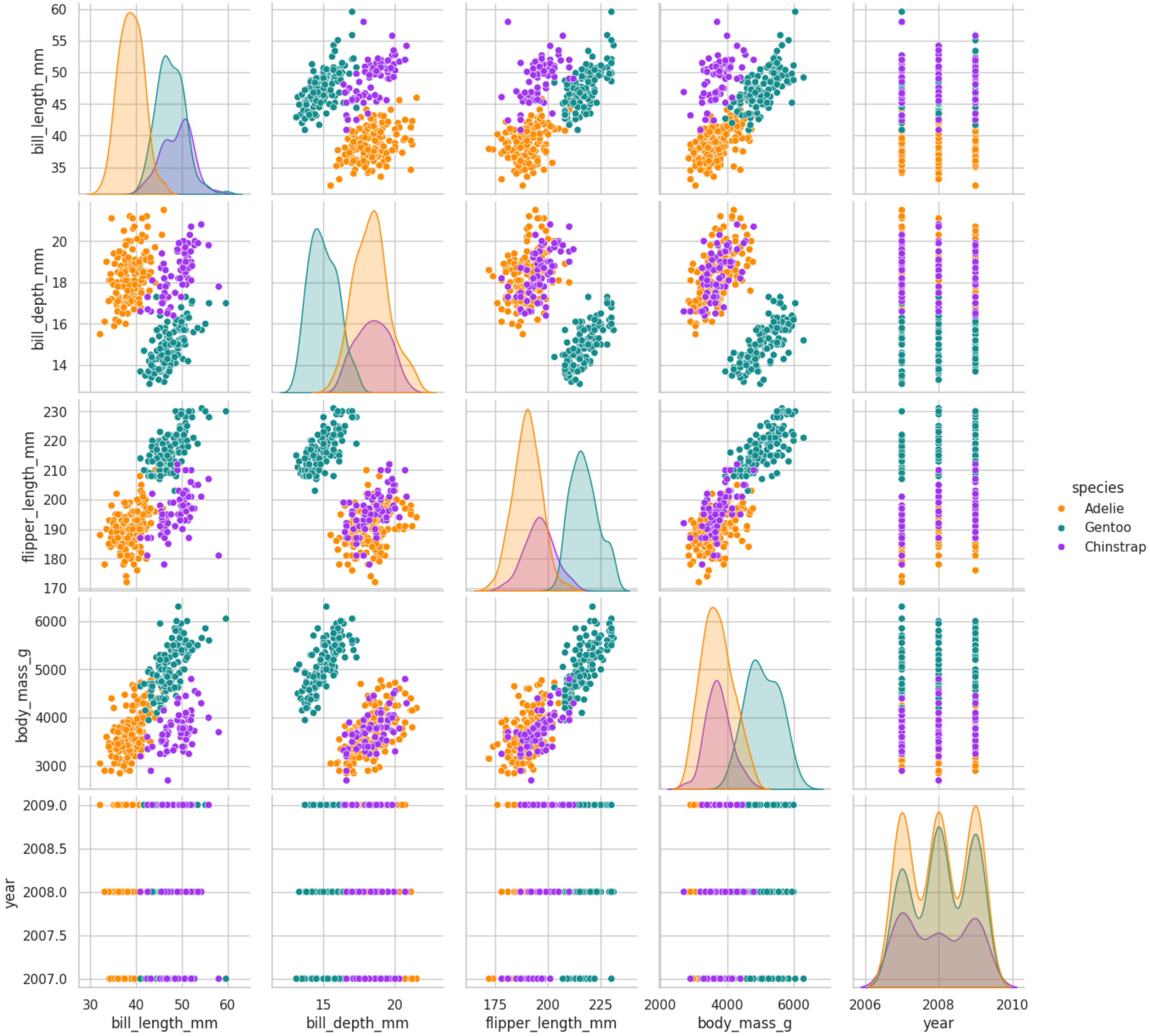
In [ ]:

```
# Question 6 Visualization 2
sns.jointplot(data=penguins, x='flipper_length_mm', y='body_mass_g', hue='species', pa
plt.xlabel('Flipper Length')
plt.ylabel('Body Mass')
plt.title('Joint Distribution of Body Mass and Flipper Length')
plt.show()
```

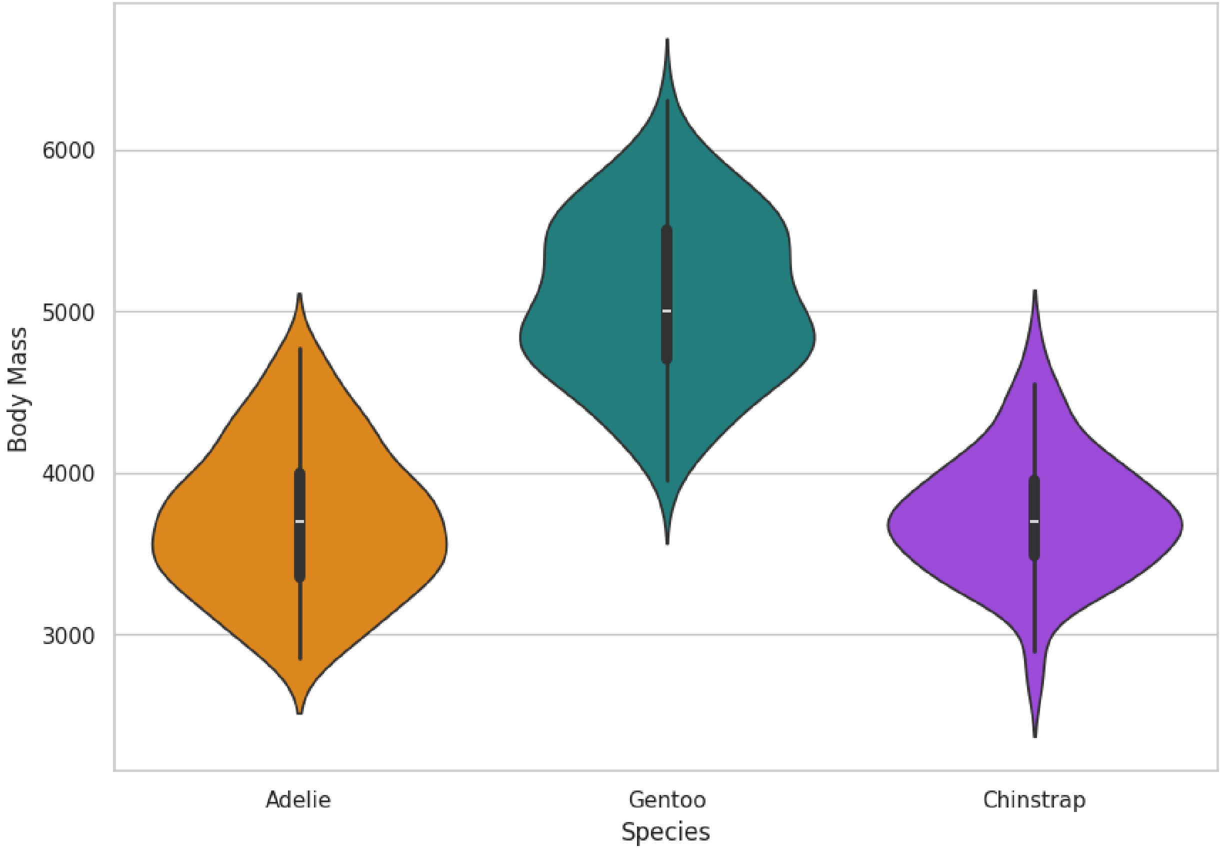


In [ ]:

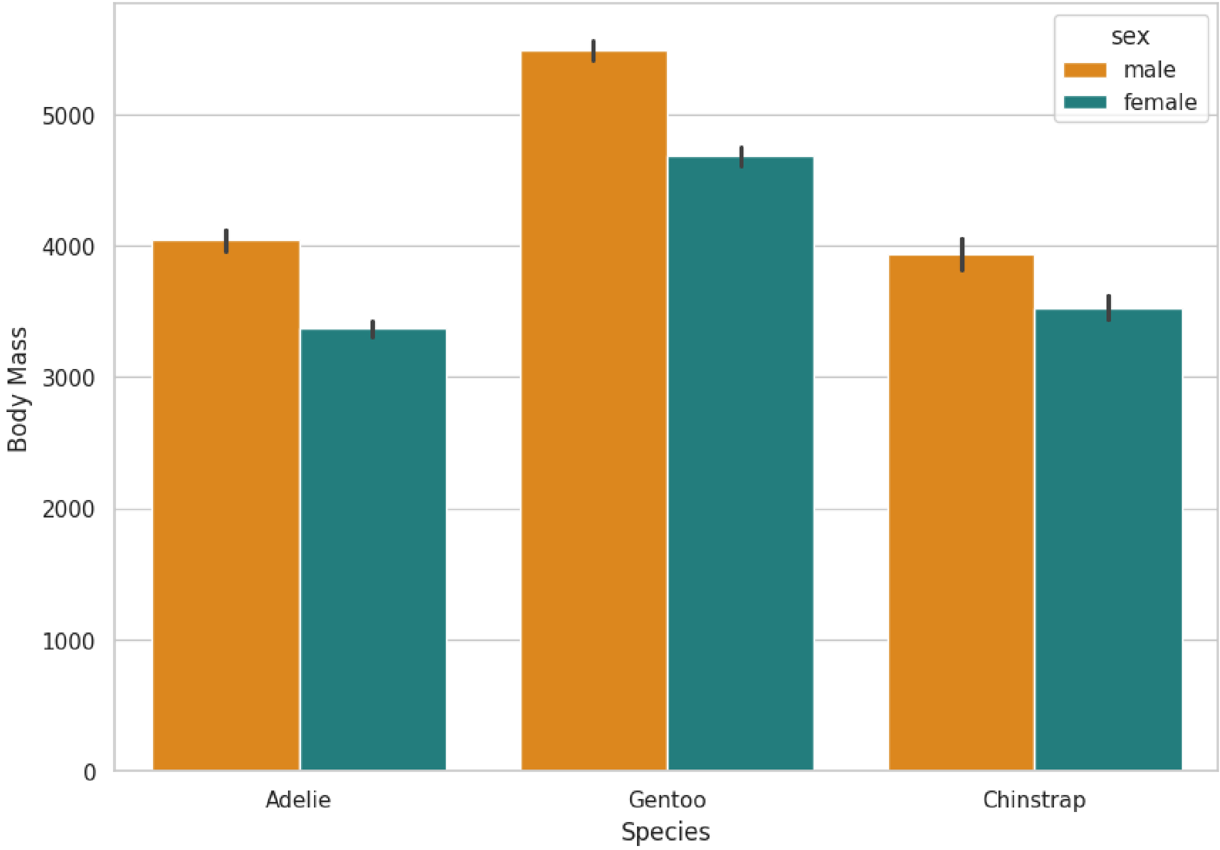
```
# Question 6 Visualization 3
sns.pairplot(penguins, hue='species', palette=['#FF8C00', '#159090', '#A034F0'])
plt.show()
```



```
In [ ]: # Question 6 Visualization 4
        plt.figure(figsize=(10,7))
        sns.violinplot(x='species', y='body_mass_g', data=penguins, palette=['#FF8C00', '#159
        plt.xlabel('Species')
        plt.ylabel('Body Mass')
        plt.show()
```



In [ ]:
```
# Question 6 Visualization 5
plt.figure(figsize=(10,7))
sns.barplot(x='species', y='body_mass_g', hue='sex', data=penguins, palette=['#FF8C00'
plt.xlabel('Species')
plt.ylabel('Body Mass')
plt.show()
```



Question 6 Narrative:

In this markdown or text cell, explain what you have learned about the Penguin data based on the exploration in this notebook.

My Findings

Body Mass Distribution: Gentoo penguins generally have a higher body mass compared to Adelie and Chinstrap.

Flipper Length Correlation: There is a positive correlation between flipper length and body mass across all species.

Island Differences: Body mass varies significantly across different islands, with Gentoo penguins on Biscoe Island being the heaviest.

Species Characteristics: Adelie and Chinstrap penguins have similar body mass distributions, while Gentoo penguins are distinctively heavier.

Sex Differences: Male penguins tend to have a higher body mass than females across all species.

Finishing Up and Submitting Your Work:

1) Save your work - you can download the .ipynb file (it can be reopened), and save it to your google drive. 2) Use File . . .Print . . PDF to generate a PDF version of your notebook (make sure all cells have been executed and show output). Turn in the PDF version of your notebook for our class assignment.

This notebook can be added to a Github repo that showcases your work for class.

In [91]:

```
!jupyter nbconvert --to html /content/Exploring_Penguins.ipynb
```

```
[NbConvertApp] Converting notebook /content/Exploring_Penguins.ipynb to html
[NbConvertApp] Writing 1843748 bytes to /content/Exploring_Penguins.html
```