UNIVERSIDAD DE COLIMA

University of Colima
Faculty of Mechanical and Electrical Engineering
Intelligent Computer Engineering

Seaborn's Penguin Dataset.
Data analysis and visualization

Larios Bravo Cristian Armando 20188165

6°D

Place: Mexico, Colima, Coquimatlan.
Date: 28/04/2024.

Penguins Dataset

Research about the penguins

- Species
- Habitat
- Consider the emperor penguin
- Images
- Video of the life of penguins
- etc

Challenge: Add at least one species with its anthropomorphic characteristics to the dataset and regenerate the graphs

The list of penguins (order Sphenisciformes, family Spheniscidae) includes aquatic, flightless birds living almost exclusively in the southern hemisphere, especially in Antarctica. Highly adapted for life in the water, members of the penguin familiy have countershaded dark and white plumage, and their wings have become flippers.

Most penguins feed on krill, fish, squid, and other forms of sealife caught while swimming underwater. They spend about half of their life on land and half in the oceans. Although all penguin species are native to the southern hemisphere, they are not found only in cold climates, such as Antarctica. In fact, only a few species of penguin live so far south. Several

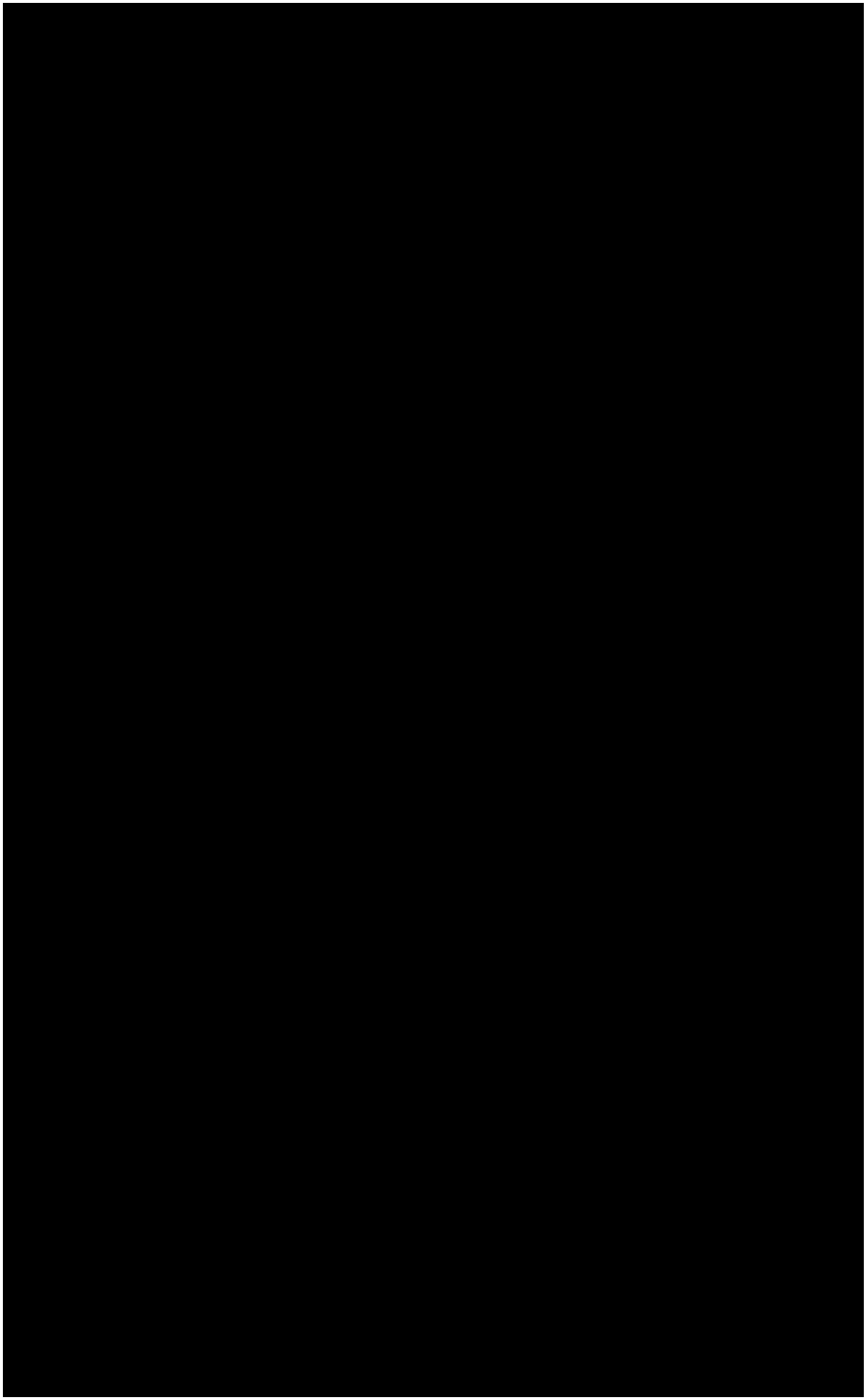
species are found in the temperate zone, and one species, the Galápagos Penguin, lives near the equator.

Penguin species

- King Penguin
- Emperor penguin
- Gentoo Penguin
- Adelie Penguin
- Chinstrap Penguin
- Southern Rockhopper Penguin
- Northern Rockhopper Penguin
- Fiordland Penguin
- Snares Penguin
- Erect-crested Penguin
- Macaroni Penguin
- Royal Penguin
- Yellow-eyed Penguin
- Little Penguin
- African Penguin
- Humboldt Penguin
- Magellanic Penguin
- Galapagos Penguin

Emperor Penguin

Emperor penguins spend their entire lives on Antarctic ice and in its surrounding waters. They are well adapted to thrive in the freezing conditions of the Antarctic. To preserve heat, they have a dense double layer of feathers – about 70 feathers per square inch – large fat reserves and, proportionally, smaller beaks and flippers compared to other penguins. They are also expert divers, with the ability to dive to depths of over 200 meters and stay underwater for up to 20 minutes.



Emperor Penguin

- Beak length: 4-5 inches (10-13 cm)
- Beak width: 1-1.2 inches (2.5-3 cm)
- Flipper length: 15-20 inches (38-50 cm)
- Body weight: 80-100 pounds (36-45 kg)

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

1. Load the Seaborn Penguin Dataset in a new Jupyter Notebook File

In [ ]:
```
penguins = sns.load_dataset("penguins")
penguins
```

Out[ ]:

	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g
0	Adelie	Torgersen	39.1	18.7	181.0	3750.0
1	Adelie	Torgersen	39.5	17.4	186.0	3800.0
2	Adelie	Torgersen	40.3	18.0	195.0	3250.0
3	Adelie	Torgersen	NaN	NaN	NaN	NaN
4	Adelie	Torgersen	36.7	19.3	193.0	3450.0
...	...	...	...	...	...	...
339	Gentoo	Biscoe	NaN	NaN	NaN	NaN
340	Gentoo	Biscoe	46.8	14.3	215.0	4850.0
341	Gentoo	Biscoe	50.4	15.7	222.0	5750.0
342	Gentoo	Biscoe	45.2	14.8	212.0	5200.0
343	Gentoo	Biscoe	49.9	16.1	213.0	5400.0

344 rows × 7 columns

2. Perform basic data exploration:

- Display the first and last few rows of the dataset.
- Explore the data types of each column.
- Calculate summary statistics.
- Check for missing values.
- Investigate the measurements of beak length, beak width, fin length and body weight of the emperor penguin species and add at least 10 rows containing this information to the dataset. You can add random measurements corresponding to the range of values of each parameter.

In [ ]:
```
# add at least 10 rows containing this information to the dataset. You can add rand
# add 10 rows to the dataset
"""
& Emperor Penguin
* Beak length: 4-5 inches (10-13 cm)
* Beak width: 1-1.2 inches (2.5-3 cm)
* Flipper length: 15-20 inches (38-50 cm)
* Body weight: 80-100 pounds (36-45 kg)
"""
new_rows = {
    "species": ["Emperor"] * 10,
    "island": ["Antarctic"] * 10,
    "bill_length_mm": np.random.uniform(100, 130, 10),
    "bill_depth_mm": np.random.uniform(25, 30, 10),
    "flipper_length_mm": np.random.uniform(380, 500, 10),
    "body_mass_g": np.random.uniform(36000,45000,10),
    "sex": [np.random.choice(["Male","Female"]) for _ in range(10)]
}
new_df = pd.DataFrame(new_rows)
# penguins = penguins.append(new_rows, ignore_index=True)
penguins = pd.concat([penguins, new_df], ignore_index=True)
penguins
```

Out[ ]:

	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g
0	Adelie	Torgersen	39.100000	18.700000	181.000000	3750.000000
1	Adelie	Torgersen	39.500000	17.400000	186.000000	3800.000000
2	Adelie	Torgersen	40.300000	18.000000	195.000000	3250.000000
3	Adelie	Torgersen	NaN	NaN	NaN	NaN
4	Adelie	Torgersen	36.700000	19.300000	193.000000	3450.000000
...	...	...	...	...	...	...
349	Emperor	Antarctic	102.727678	26.925566	466.915824	44241.891064
350	Emperor	Antarctic	120.179639	25.704479	435.229647	41065.831102
351	Emperor	Antarctic	124.358282	28.259777	492.886645	44728.089629
352	Emperor	Antarctic	115.379472	25.774184	467.355390	39867.729823
353	Emperor	Antarctic	129.860548	26.642180	489.863843	39418.621734

354 rows × 7 columns

In [ ]:
```
# Initial Exploration
penguins.head()
```

Out[ ]:

	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	
0	Adelie	Torgersen	39.1	18.7	181.0	3750.0	I
1	Adelie	Torgersen	39.5	17.4	186.0	3800.0	Fe
2	Adelie	Torgersen	40.3	18.0	195.0	3250.0	Fe
3	Adelie	Torgersen	NaN	NaN	NaN	NaN	
4	Adelie	Torgersen	36.7	19.3	193.0	3450.0	Fe

In [ ]:
```
# How many rows and columns?
penguins.shape
```

Out[ ]:
```
(354, 7)
```

In [ ]:
```
penguins.describe()
```

Out[ ]:

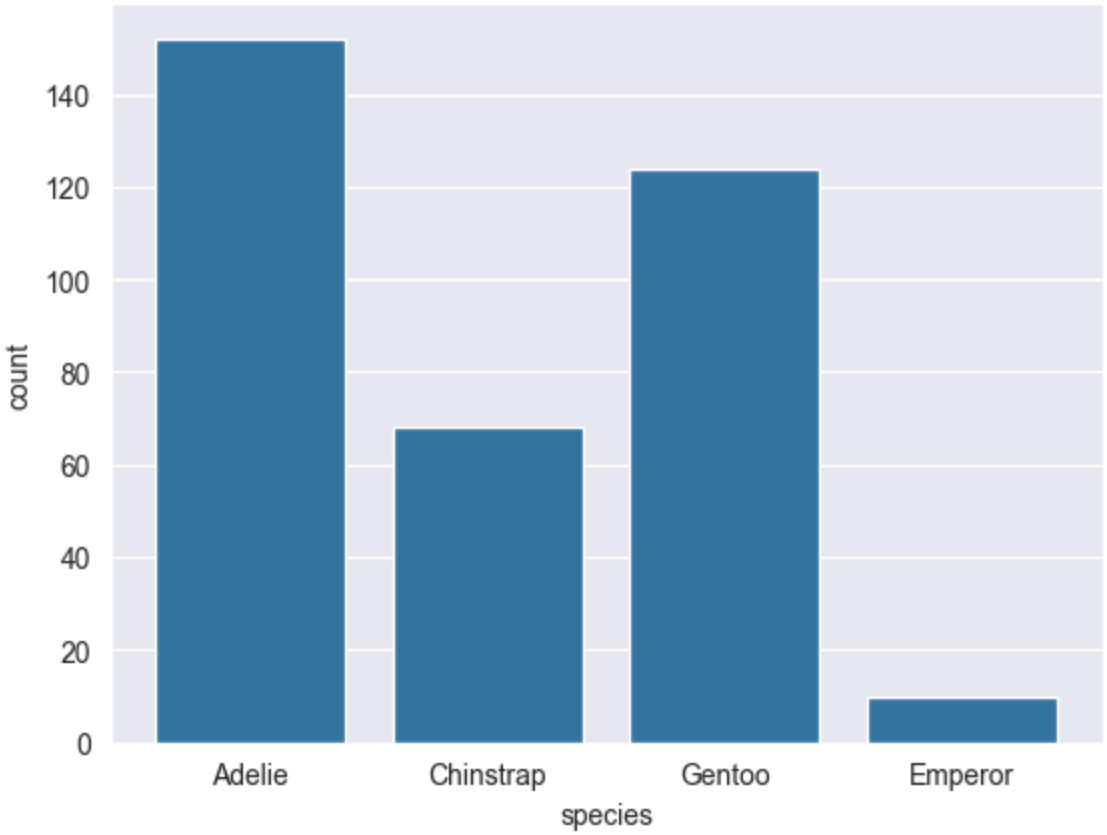
	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g
count	352.000000	352.000000	352.000000	352.000000
mean	45.957910	17.440099	207.926744	5259.829061
std	13.143830	2.588339	43.629082	6259.489993
min	32.100000	13.100000	172.000000	2700.000000
25%	39.500000	15.675000	190.000000	3550.000000
50%	44.950000	17.450000	197.500000	4062.500000
75%	49.000000	18.800000	215.000000	4850.000000
max	129.860548	29.763769	492.886645	44728.089629

2. Data Visualization

Create a count plot to visualize the distribution of penguin species.

In [ ]:
```
# Create a count plot to visualize the distribution of penguin species.
sns.countplot(data=penguins, x="species")
plt.plot()
```

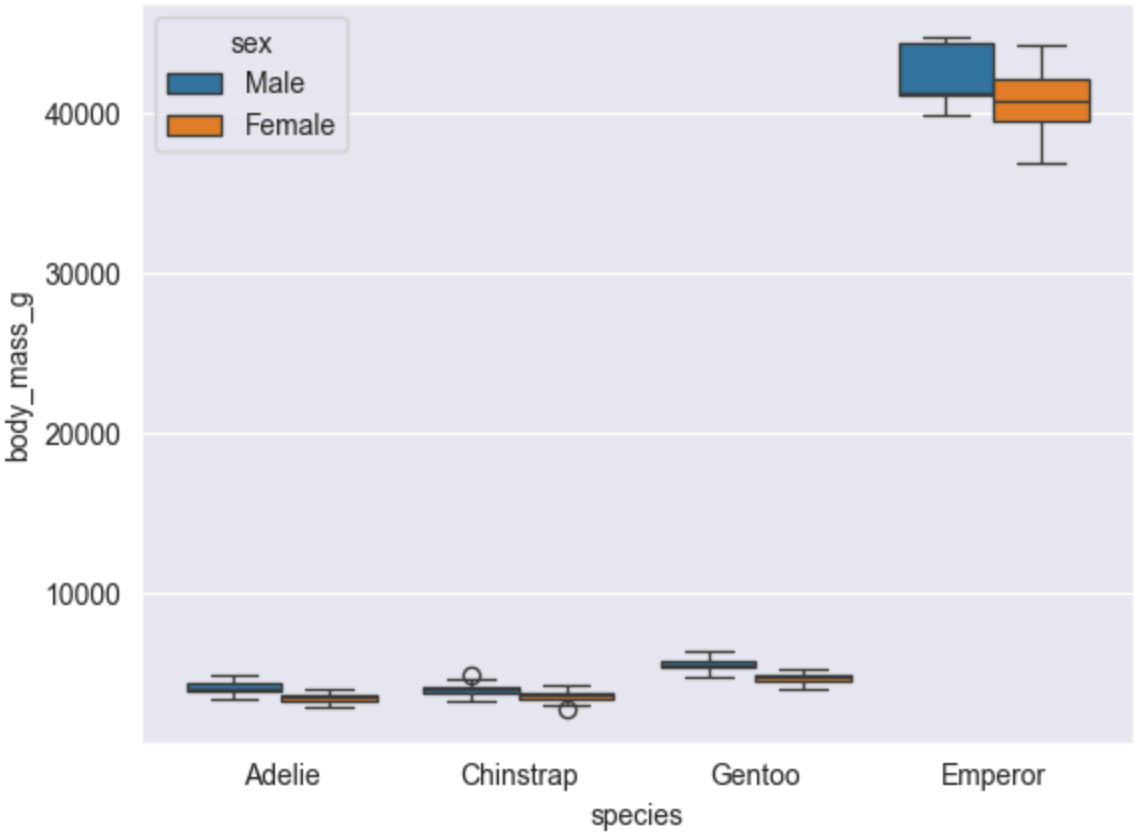
Out[ ]:
```
[]
```



Generate a box plot to compare the body mass of penguins based on their species and sex.

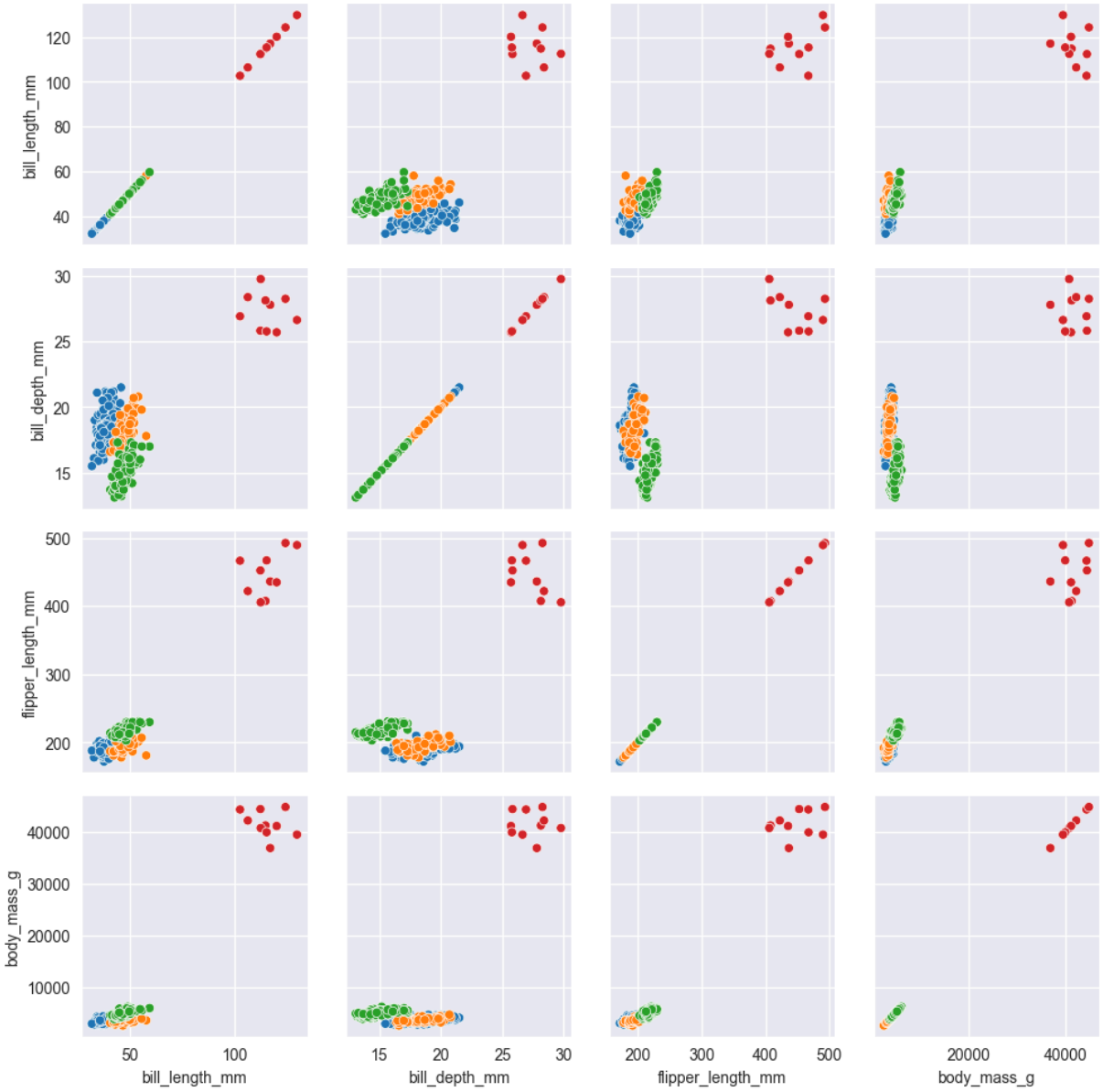
In [ ]:
```
# Generate a box plot to compare the body mass of penguins based on their species a
sns.boxplot(data=penguins, x="species", y="body_mass_g",
            hue="sex")
plt.plot()
```

Out[ ]:
```
[]
```



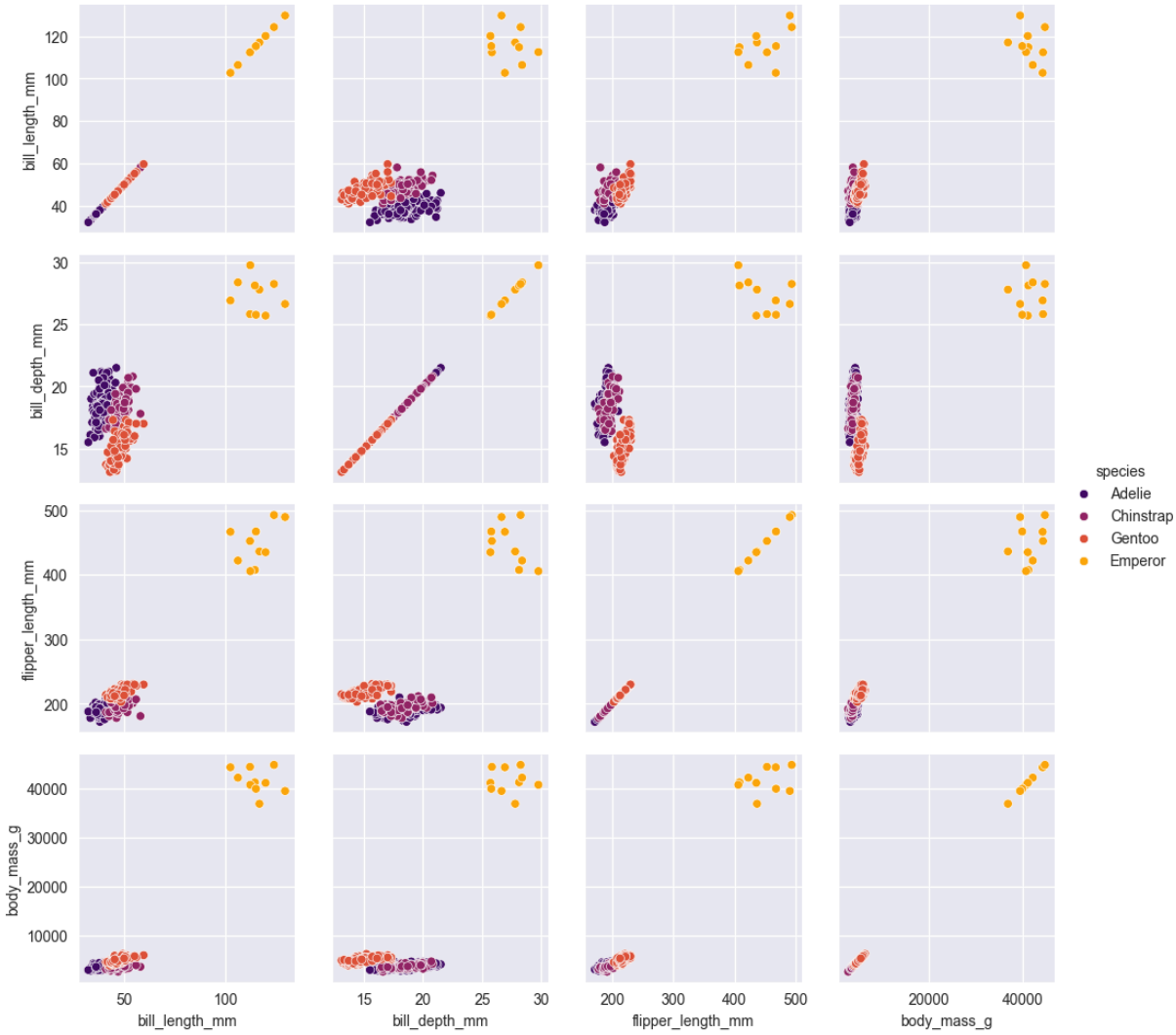
How can we tell penguin species apart

In [ ]:
```
# Change the seaborn style
sns.set_style('darkgrid')
# Create a pairplot of the penguins data (numeric columns only)
g = sns.PairGrid(penguins, hue='species')
# Visualize the pairplot
g.map(sns.scatterplot)
plt.show()
```



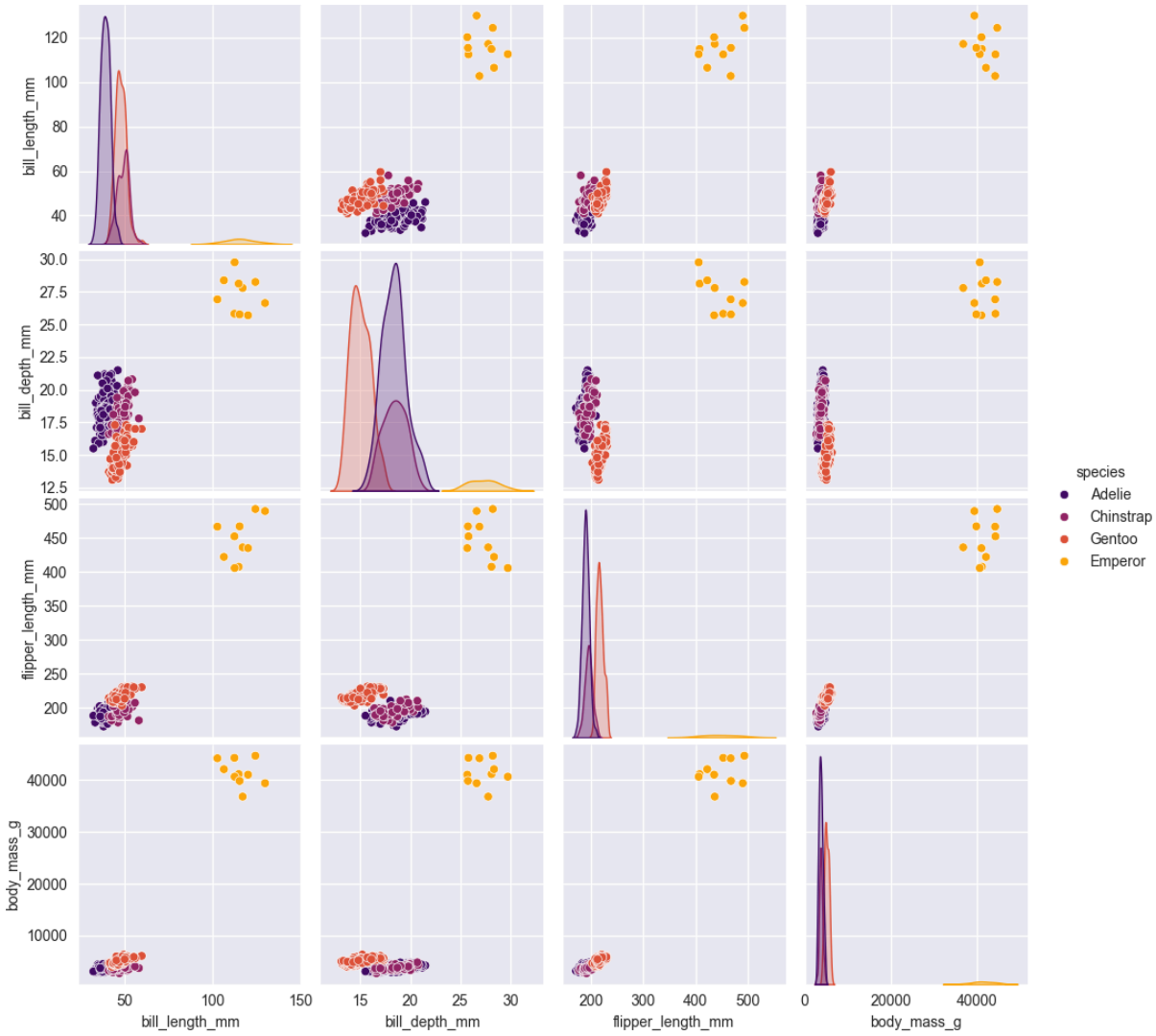
Add color to the data of graphic and a legend

In [ ]:
```
# Change the seaborn style
sns.set_style('darkgrid')
# Create a pairplot of the penguins data (numeric columns only)
# Use the 'species' column to color the points
# Use the 'inferno' palette
# 4 variables: bill_length_mm, bill_depth_mm, flipper_length_mm, body_mass_g
g = sns.PairGrid(penguins, hue='species', palette='inferno')
# Visualize the pairplot
g.map(sns.scatterplot)
g.add_legend()
# plt.legend()
plt.show()
```

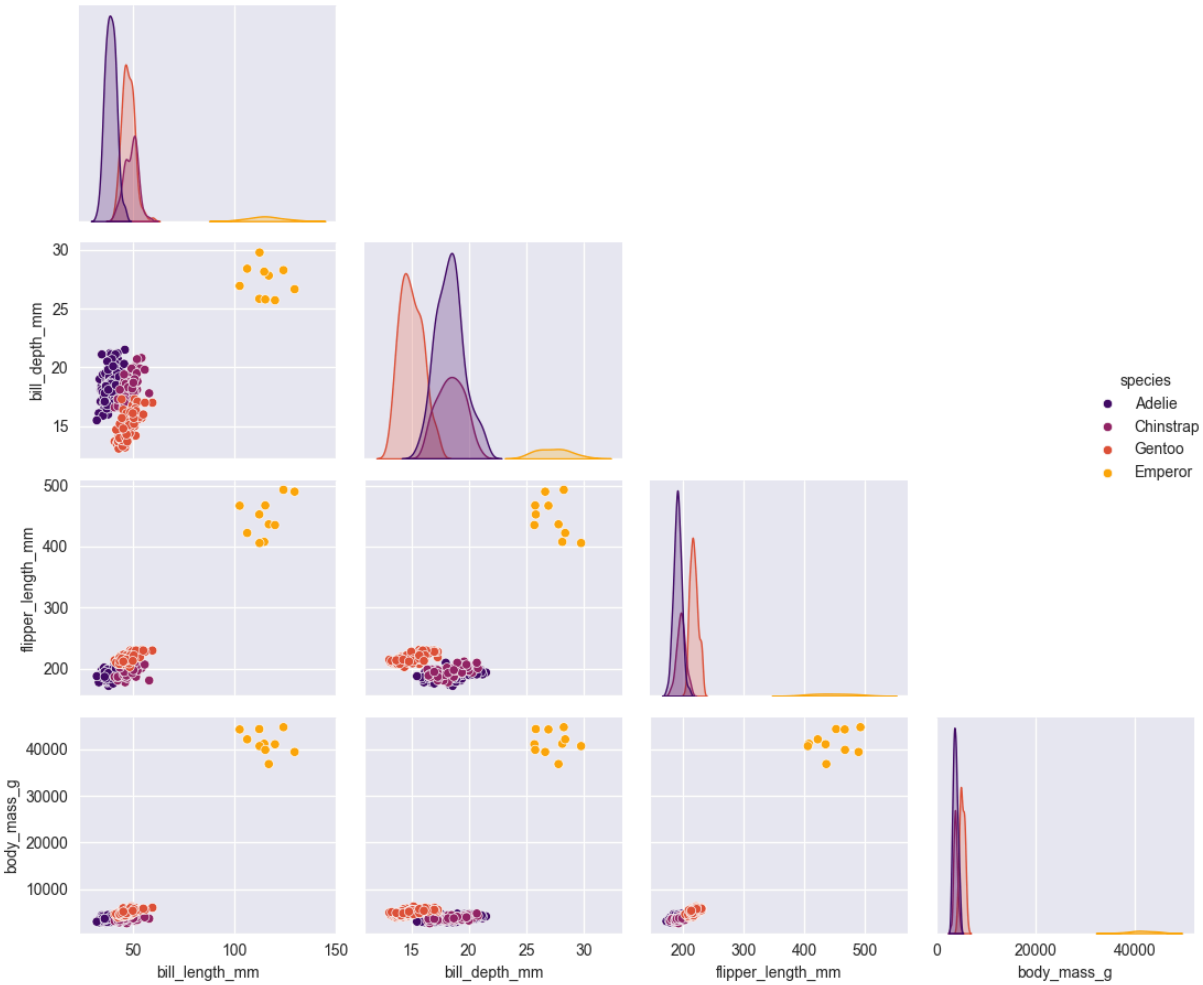


In [ ]:
```
sns.pairplot(data=penguins, hue='species', palette='inferno')
```

Out[ ]: <seaborn.axisgrid.PairGrid at 0x2a92b176ab0>



In [ ]:
```
# No repeated plots
grid = sns.pairplot(data=penguins, hue='species', palette='inferno', corner=True)
grid.fig.suptitle('Features Comparison in Penguins Species', y=1.02, size=20)
# Adjust the plot
plt.tight_layout()
grid.fig.subplots_adjust(top=0.9)
plt.show()
```



In [ ]:
```
# Drop nan values
penguins.dropna(inplace=True)
```

In [ ]:
```
# Show nan values
# Print "sex" columns nan values
if (penguins["sex"].isnull().sum()) == 0 :
    print("No nan values")
else:
    print("Nan values in the data")
```

```
No nan values
```

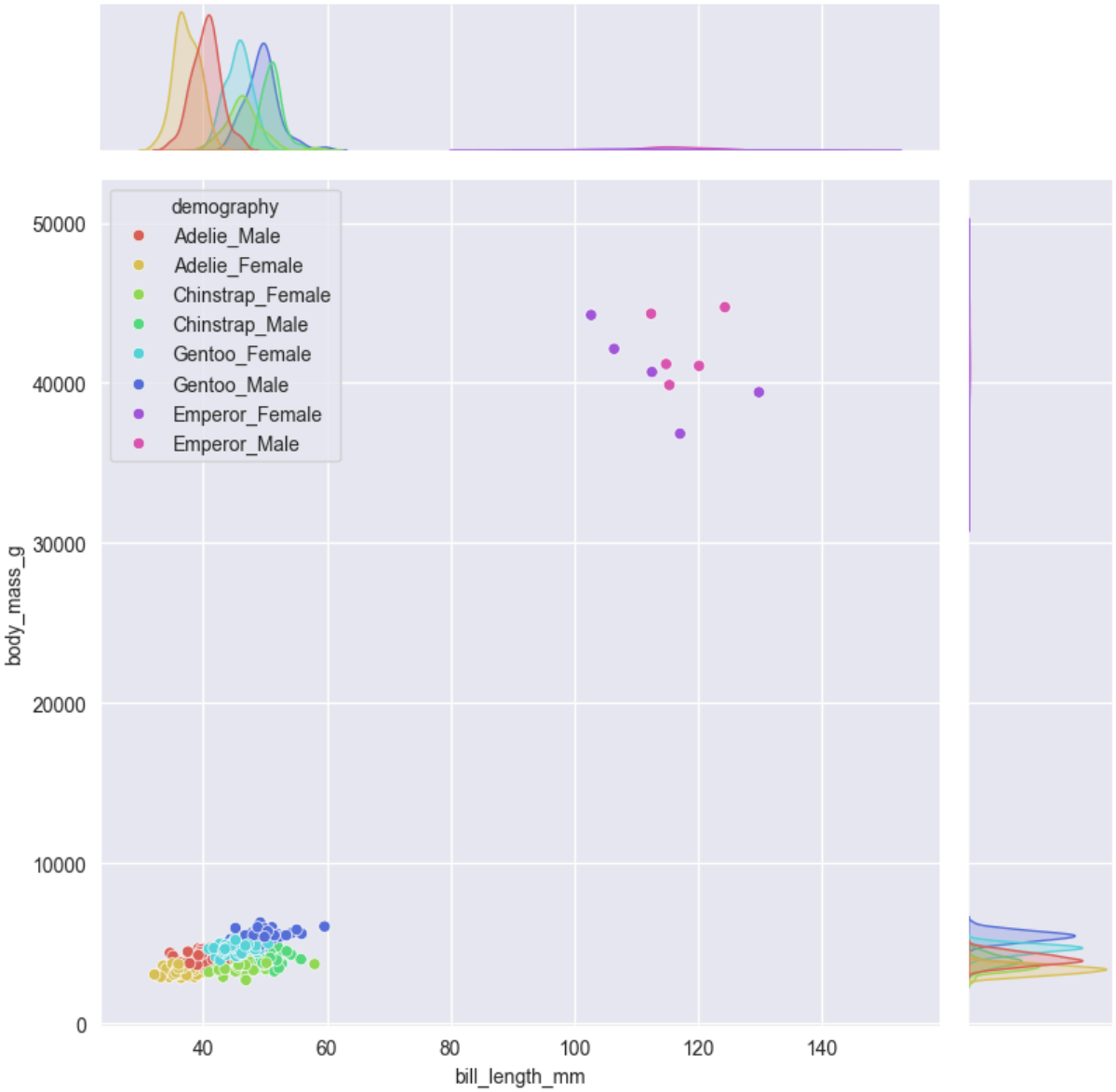
In [ ]:
```
# Clasify the penguins species based on the sex
# Create a new column to describe the sex and species
penguins["demography"] = penguins.apply(lambda row: '%s_%s' % (row["species"], row[
```

In [ ]:
```
penguins.head()
```

Out[ ]:

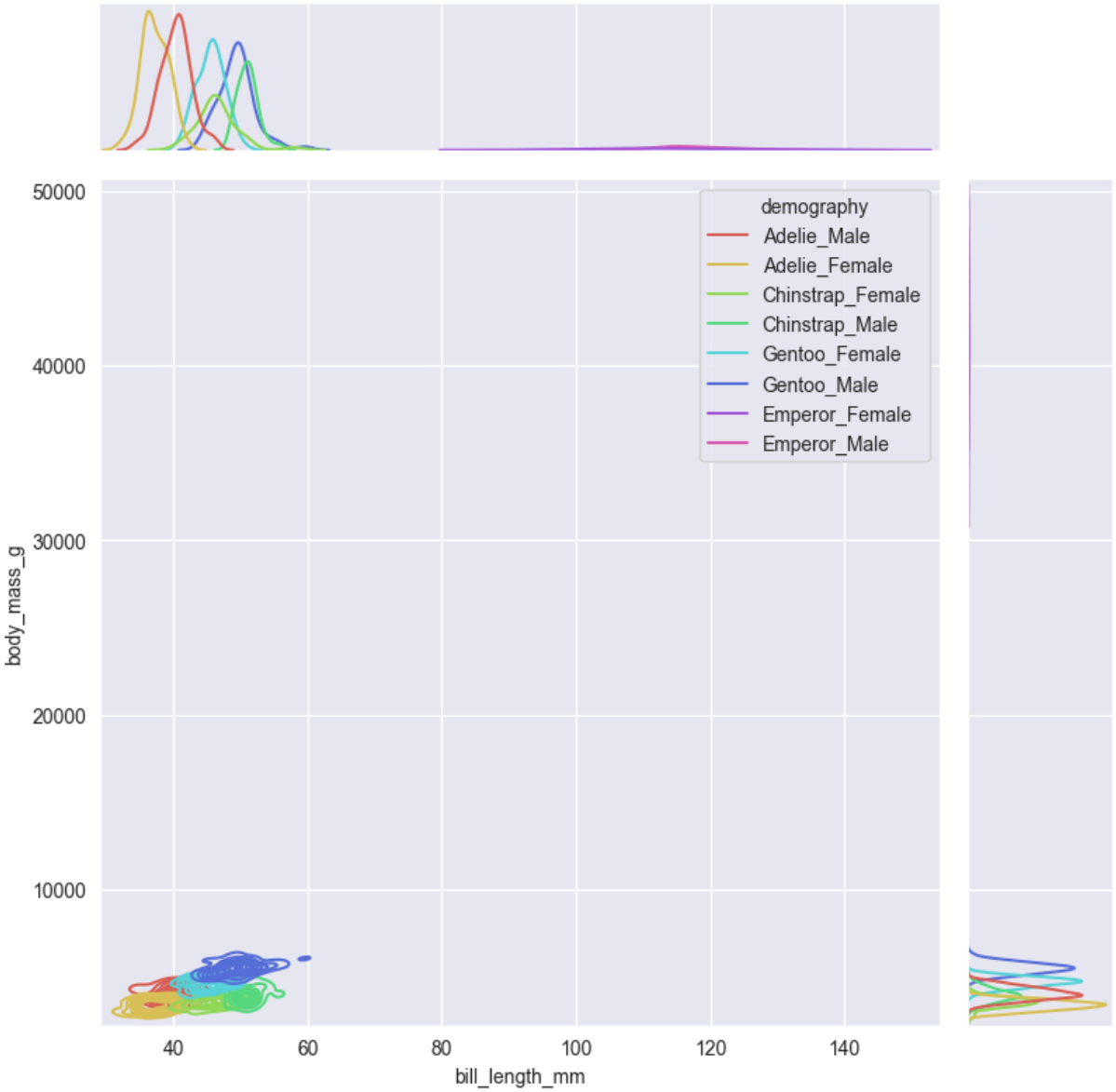
	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	
0	Adelie	Torgersen	39.1	18.7	181.0	3750.0	I
1	Adelie	Torgersen	39.5	17.4	186.0	3800.0	Fe
2	Adelie	Torgersen	40.3	18.0	195.0	3250.0	Fe
4	Adelie	Torgersen	36.7	19.3	193.0	3450.0	Fe
5	Adelie	Torgersen	39.3	20.6	190.0	3650.0	I

In [ ]:
```
# Scatter plot of the penguins data
sns.jointplot(data=penguins, x='bill_length_mm', y='body_mass_g', hue='demography',
plt.show()
```

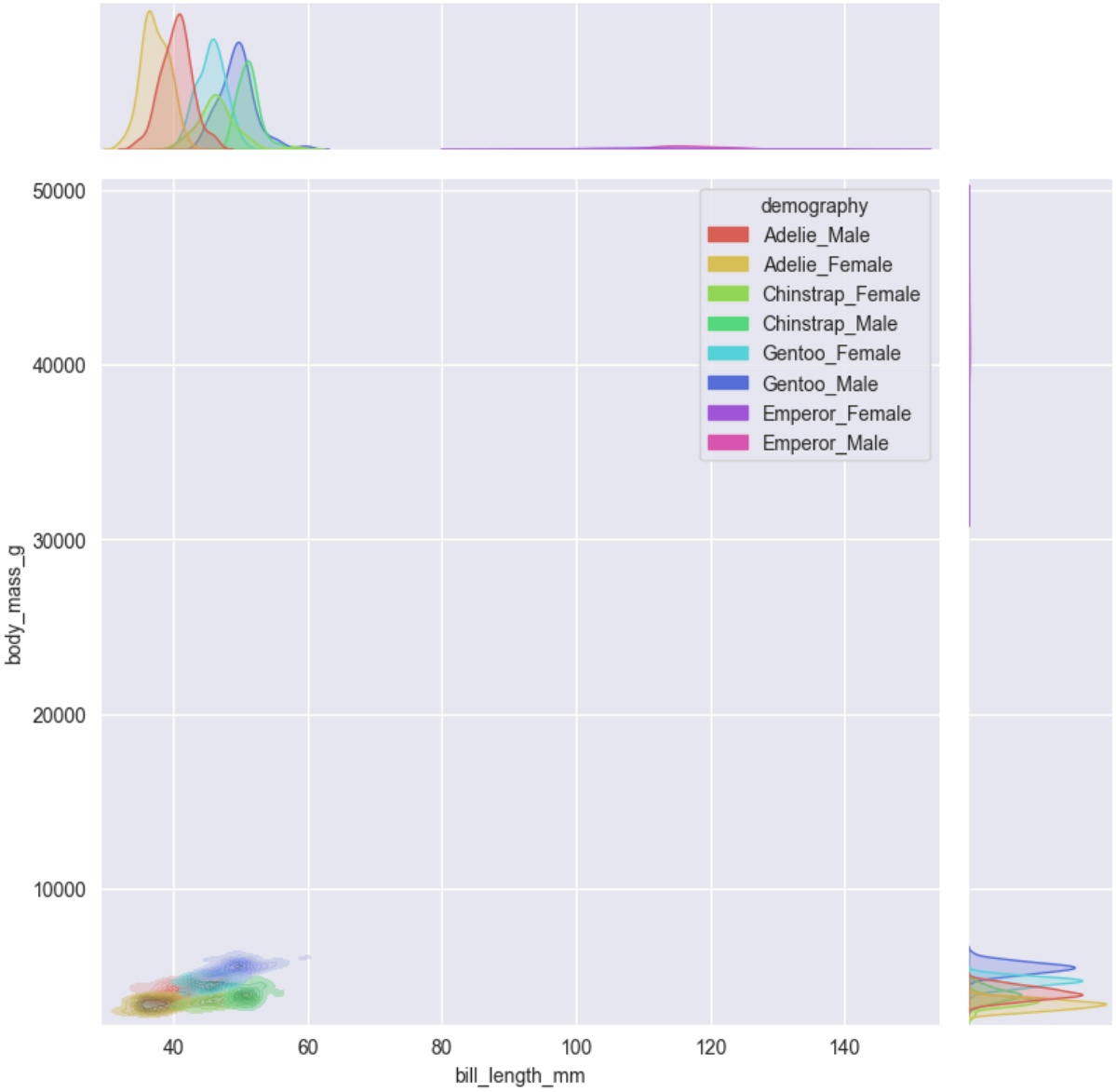


In [ ]:
```
# kde plot of the penguins data
sns.jointplot(data=penguins, x='bill_length_mm', y='body_mass_g', hue='demography',
```

```
plt.show()
```



In [ ]:
```
# Affinity of density plot
sns.jointplot(data=penguins, x='bill_length_mm', y='body_mass_g', hue='demography',
plt.show()
```



How the development of this activity has helped me to improve my skills in the domain of analysis and visualization of datasets?

Working with the penguin dataset enhances skills in data analysis and visualization. Through tasks like cleaning, exploring relationships, and statistical analysis, one can glean insights into data structures and patterns. Visualization techniques, including scatter plots and histograms, aid in communicating findings effectively. Additionally, the dataset provides opportunities for predictive modeling, enabling the application of machine learning algorithms to classify penguin species based on attributes. Overall, this hands-on experience fosters proficiency in handling real-world datasets, a vital skill set for data-driven decision-making in diverse domains.