PYTHON PROGRAMMING

DIGITAL ASSIGNMENT

Name: SOUMYADEEP GANGULY

Reg No: 24MDT0082

Code: PMDS508L

Seaborn

Installation

Install seaborn using pip

```
  pip install seaborn

import seaborn as sns
import matplotlib.pyplot as plt

import numpy as np
import pandas as pd
```

Set a default color theame

```
# Set a theme
sns.set_theme(style="whitegrid")
```

Load a inbuild dataset from seaborm module

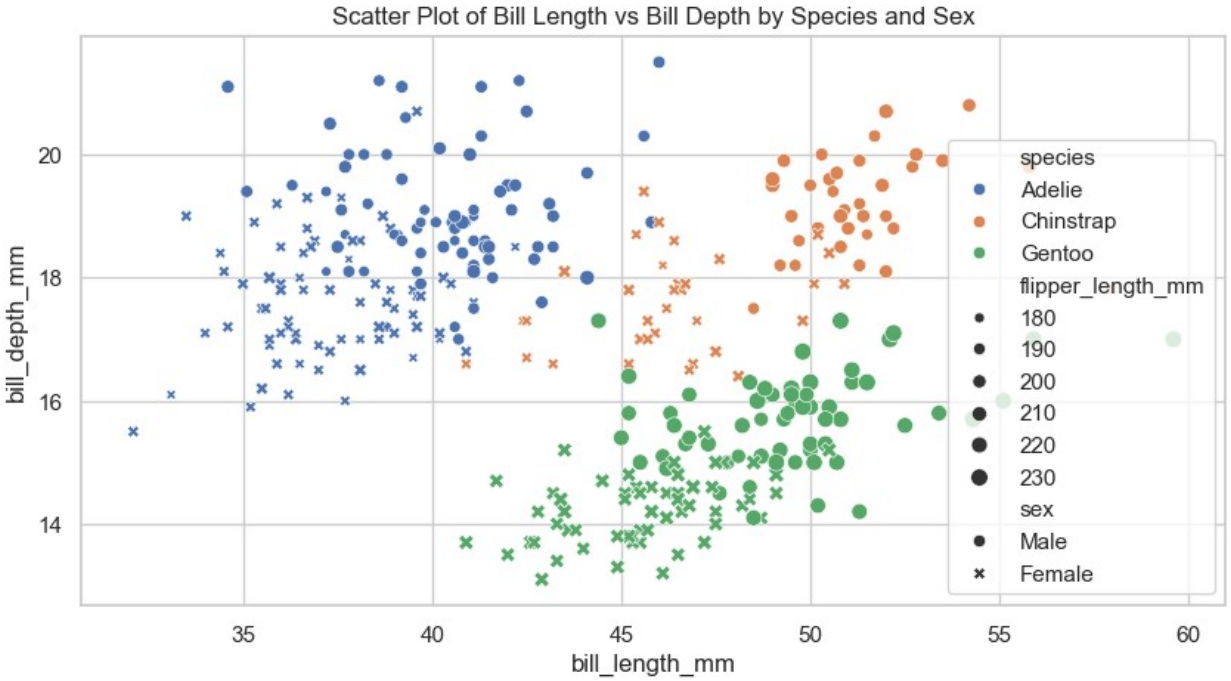
```
# Load the penguins dataset
penguins = sns.load_dataset("penguins")
```

1. Scatter Plot

Scatter plots are useful for examining relationships between two numerical variables. The scatter plot can also include visual encodings like color `(hue)`, marker style `(style)`, and size `(size)` to represent additional dimensions.

```
plt.figure(figsize=(10,5))
sns.scatterplot(data=penguins, x="bill_length_mm", y="bill_depth_mm",
hue="species", style="sex", size="flipper_length_mm")
```

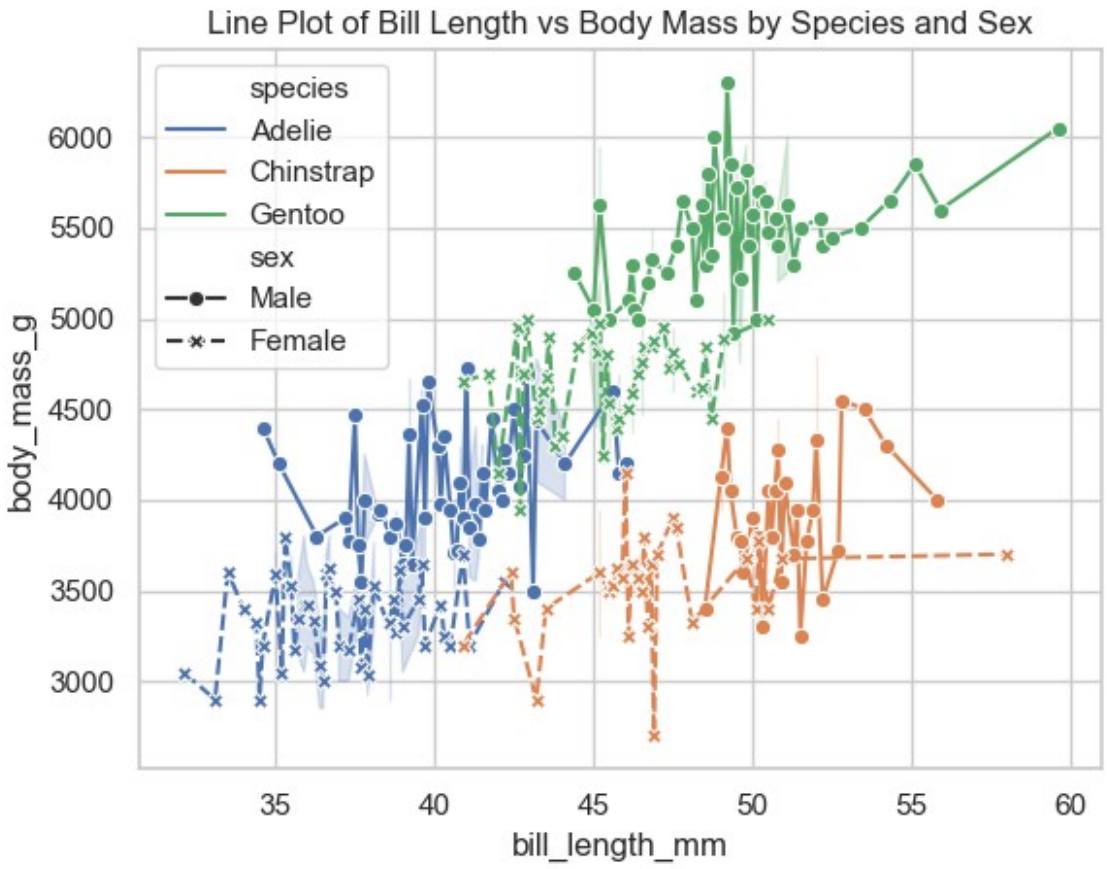
```
plt.title("Scatter Plot of Bill Length vs Bill Depth by Species and
Sex")
plt.show()
```



2. Line Plot

Line plots are commonly used to show trends over time or continuous variables. While it's typically time-based, it can also be used for ordered data.

```
sns.lineplot(data=penguins, x="bill_length_mm", y="body_mass_g",
hue="species", style="sex", markers=True)
plt.title("Line Plot of Bill Length vs Body Mass by Species and Sex")
plt.show()
```

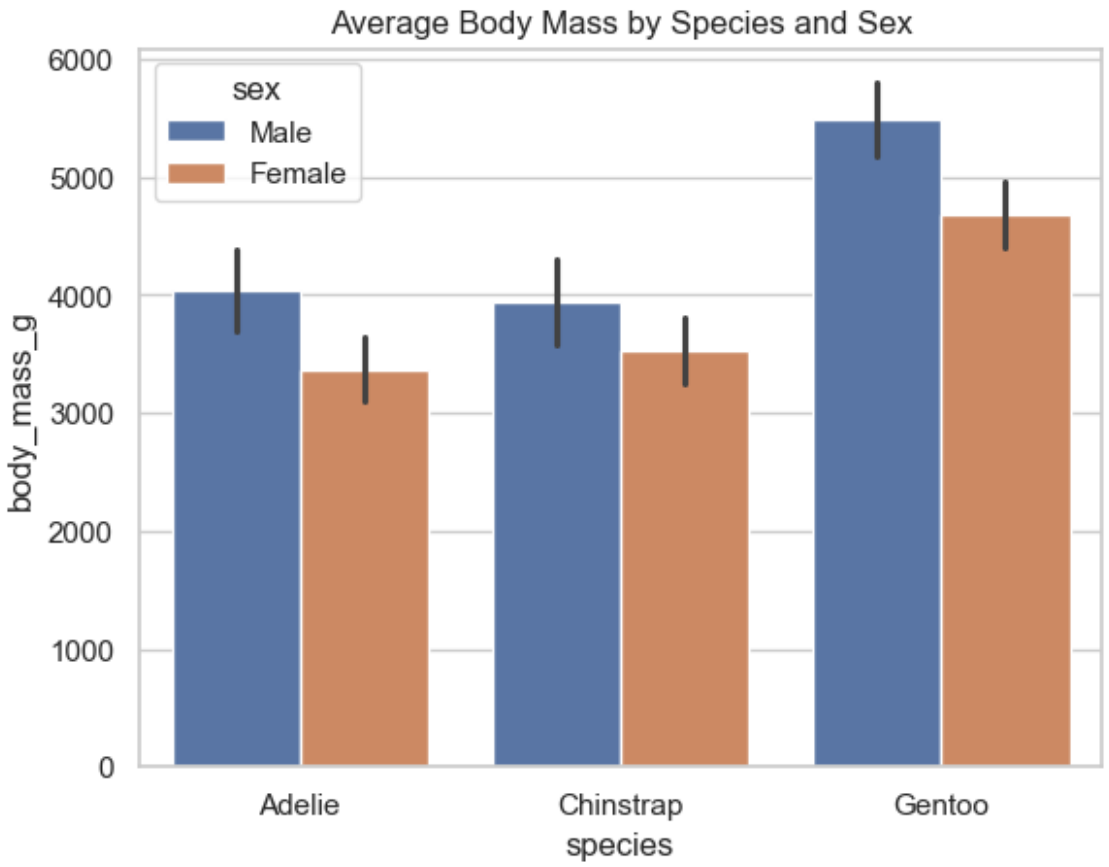


3. Bar Plot

A bar plot is excellent for comparing categories. Here, it represents the average body mass for each penguin species.

```
sns.barplot(data=penguins, x="species", y="body_mass_g", hue="sex",
ci="sd")
plt.title("Average Body Mass by Species and Sex")
plt.show()
```

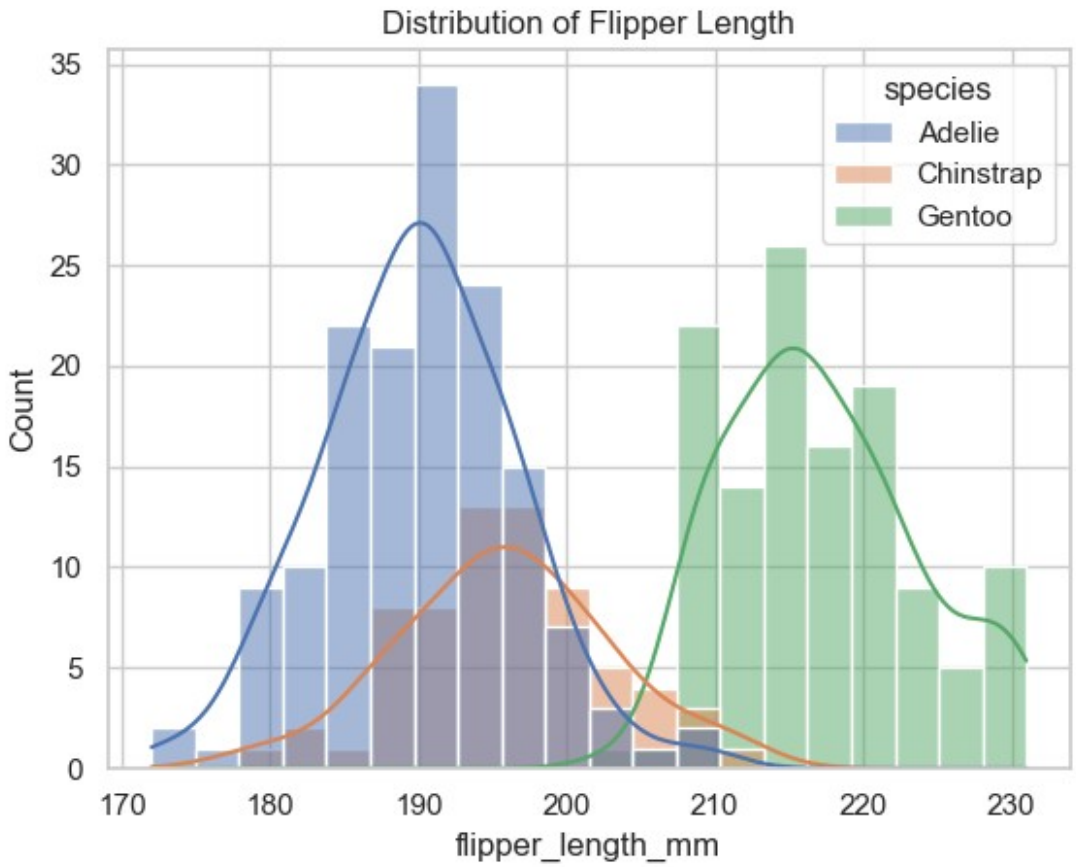
```
C:\Users\sambh\AppData\Local\Temp\ipykernel_14388\3577134162.py:1:
FutureWarning:

The `ci` parameter is deprecated. Use `errorbar='sd'` for the same
effect.

  sns.barplot(data=penguins, x="species", y="body_mass_g", hue="sex",
ci="sd")
```



4. Histogram

Histograms are great for visualizing the distribution of a single numerical variable, such as flipper length.

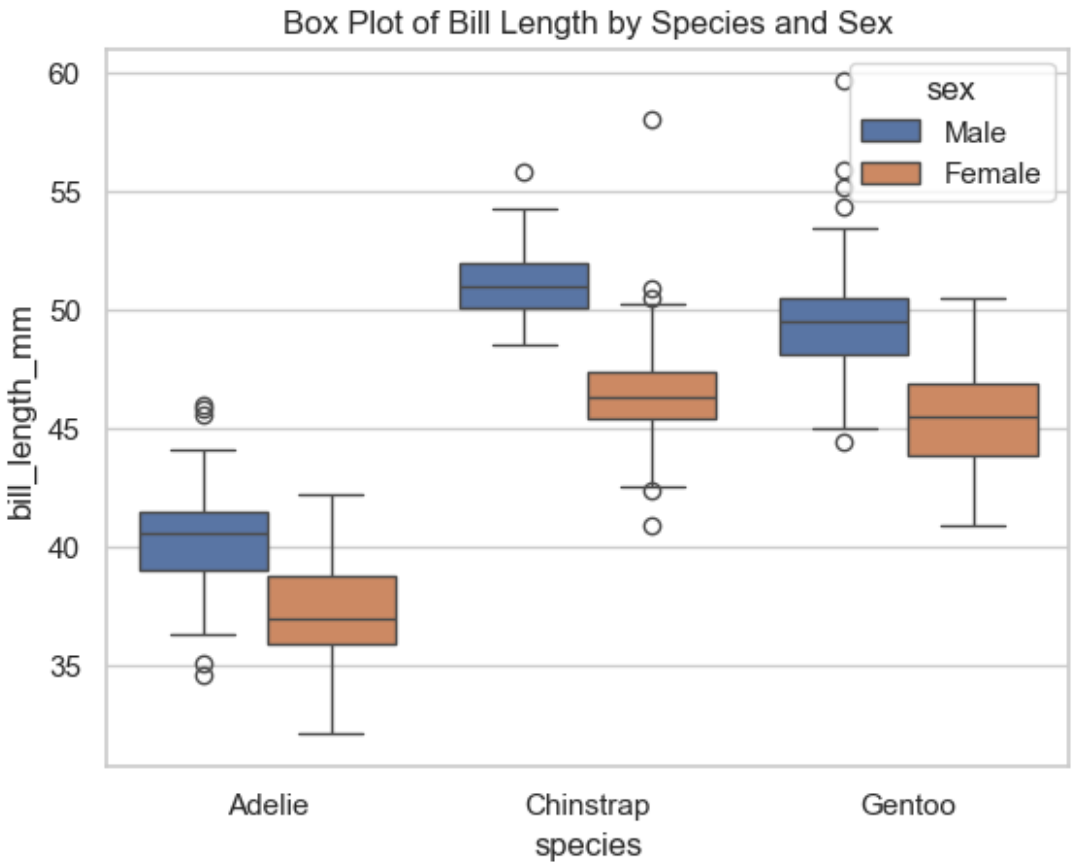
```
sns.histplot(data=penguins, x="flipper_length_mm", hue="species",
bins=20, kde=True)
plt.title("Distribution of Flipper Length")
plt.show()
```



5. Box Plot

Box plots summarize data with quartiles and are useful for identifying outliers.

```
sns.boxplot(data=penguins, x="species", y="bill_length_mm", hue="sex")
plt.title("Box Plot of Bill Length by Species and Sex")
plt.show()
```

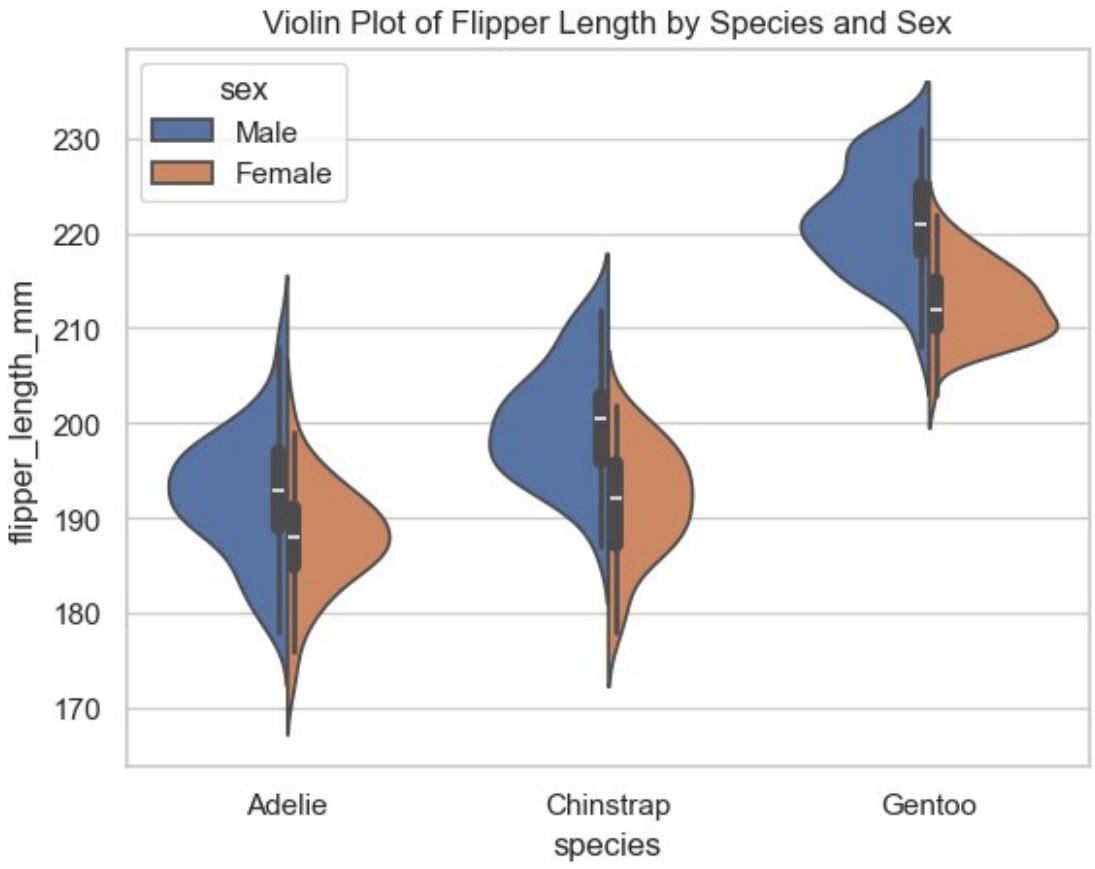


Each box plot shows the interquartile range (IQR) of bill length for each species, with the line inside representing the median. Outliers are shown as points outside the "whiskers."

6. Violin Plot

Violin plots combine features of box plots and KDE to show the distribution and probability density.

```
sns.violinplot(data=penguins, x="species", y="flipper_length_mm",
hue="sex", split=True)
plt.title("Violin Plot of Flipper Length by Species and Sex")
plt.show()
```



The violin plot shows the distribution of flipper length, with width indicating density. This plot helps visualize both the central tendency and variability.

7. Strip Plot

A strip plot shows individual data points, which is useful for understanding the data's granularity.

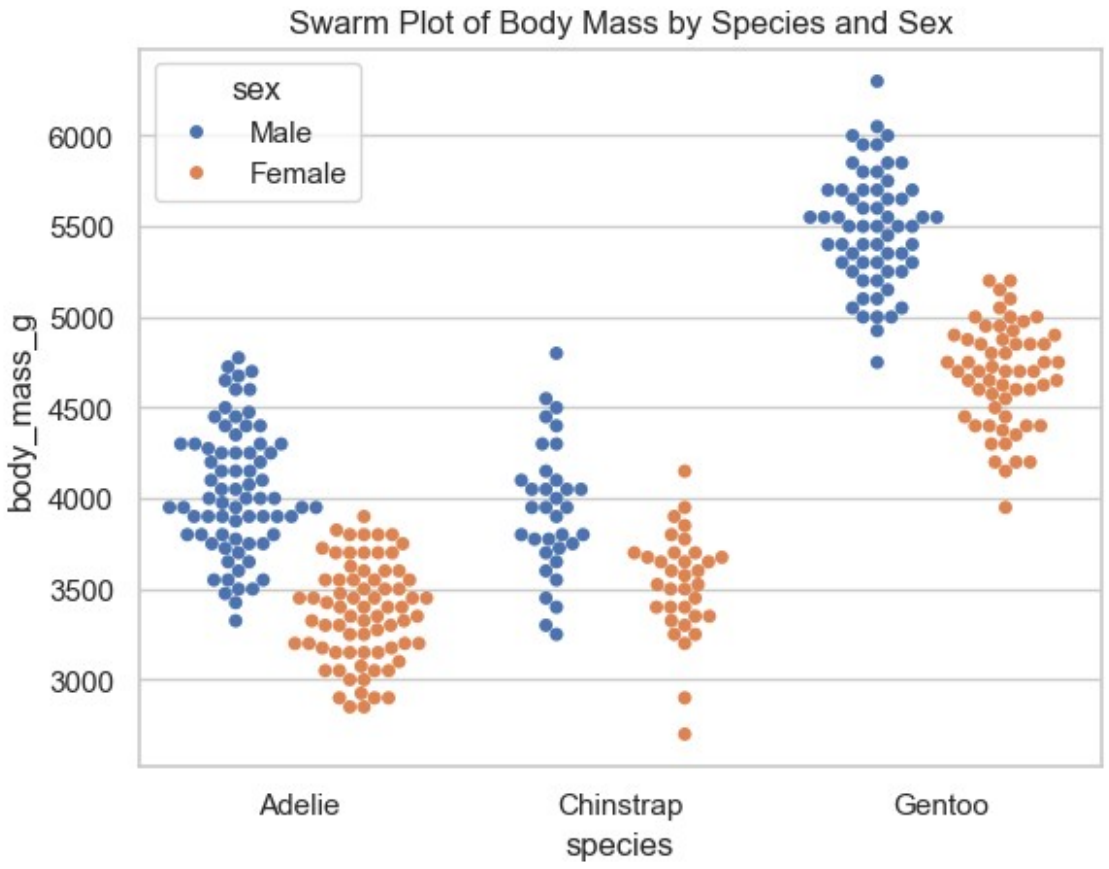
```
sns.stripplot(data=penguins, x="species", y="body_mass_g", hue="sex",
dodge=True)
plt.title("Strip Plot of Body Mass by Species and Sex")
plt.show()
```

Each dot represents an individual penguin, showing body mass across species and sex. `Dodge=True` offsets the points for clarity between categories.

8. Swarm Plot

Swarm plots are similar to strip plots but avoid overlapping points, making it easier to see individual values.

```
sns.swarmplot(data=penguins, x="species", y="body_mass_g", hue="sex",
dodge=True)
plt.title("Swarm Plot of Body Mass by Species and Sex")
plt.show()
```

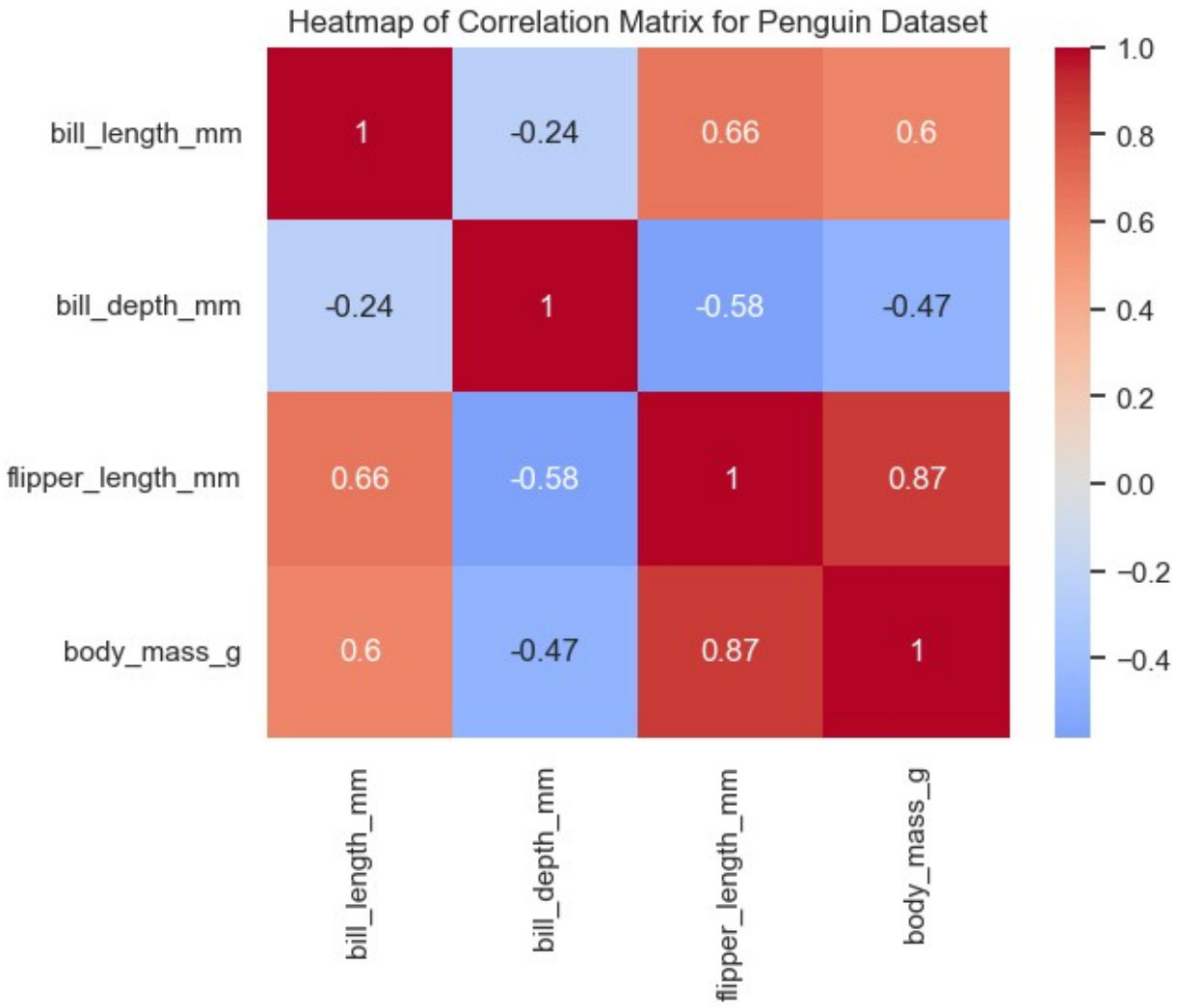


The swarm plot arranges points without overlapping, which allows for a clear comparison of individual data points in each category.

9. Heatmap

Heatmaps are used to visualize the intensity of data at the intersection of two categories, often useful for correlation matrices.

```
penguins1 = penguins.drop(['species', 'island', 'sex'], axis=1)
corr = penguins1.dropna().corr()  # Drop NaN values to calculate
correlation
sns.heatmap(corr, annot=True, cmap="coolwarm", center=0)
plt.title("Heatmap of Correlation Matrix for Penguin Dataset")
plt.show()
```

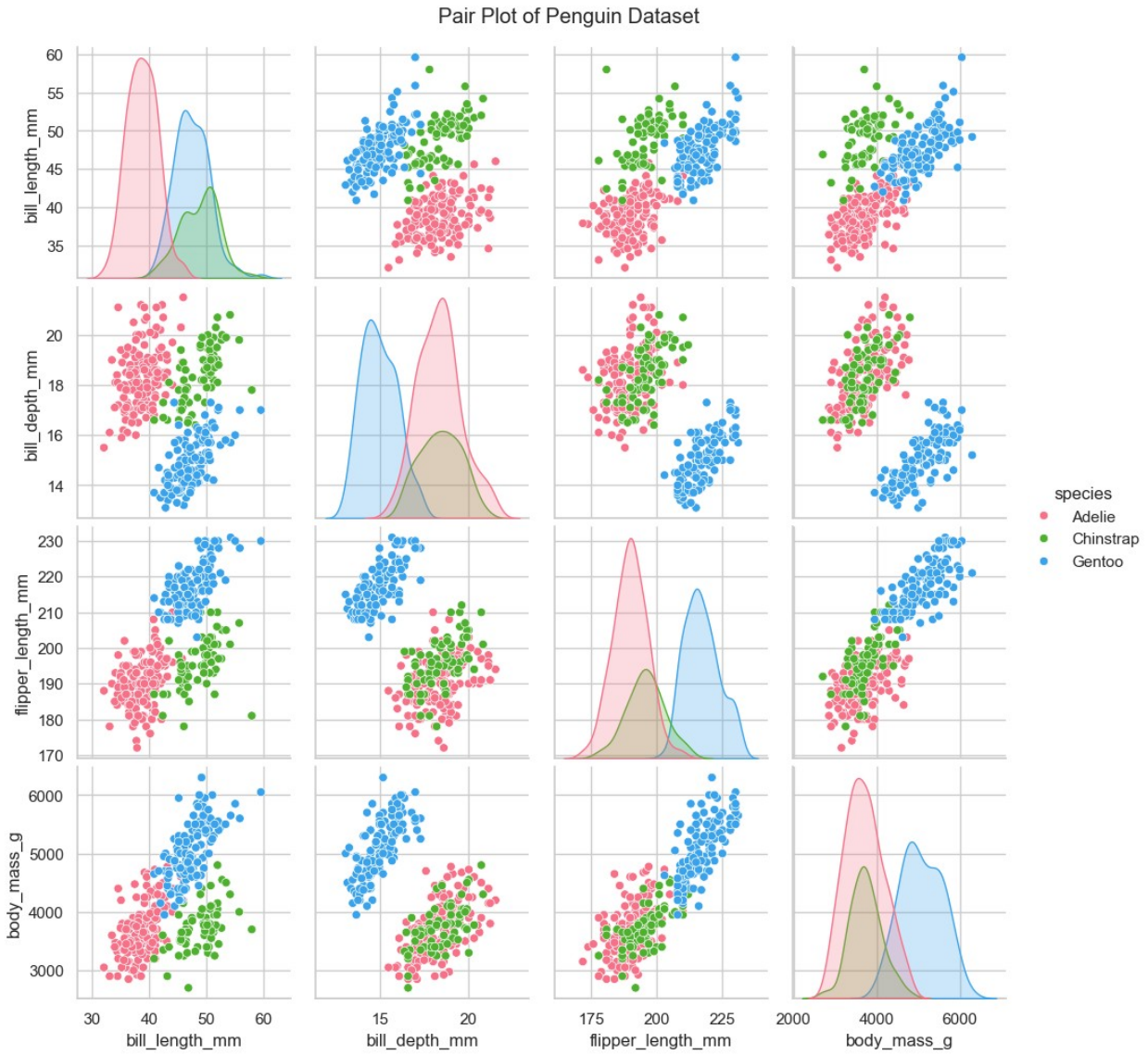


This heatmap visualizes the correlation between variables. Values closer to `1 or -1` indicate a strong relationship, while values near 0 suggest weak or no correlation.

10. Pair Plot

A pair plot shows scatter plots for all variable pairs, providing an overview of relationships in the dataset.

```
sns.pairplot(penguins, hue="species", palette="husl")
plt.suptitle("Pair Plot of Penguin Dataset", y=1.02)
plt.show()
```

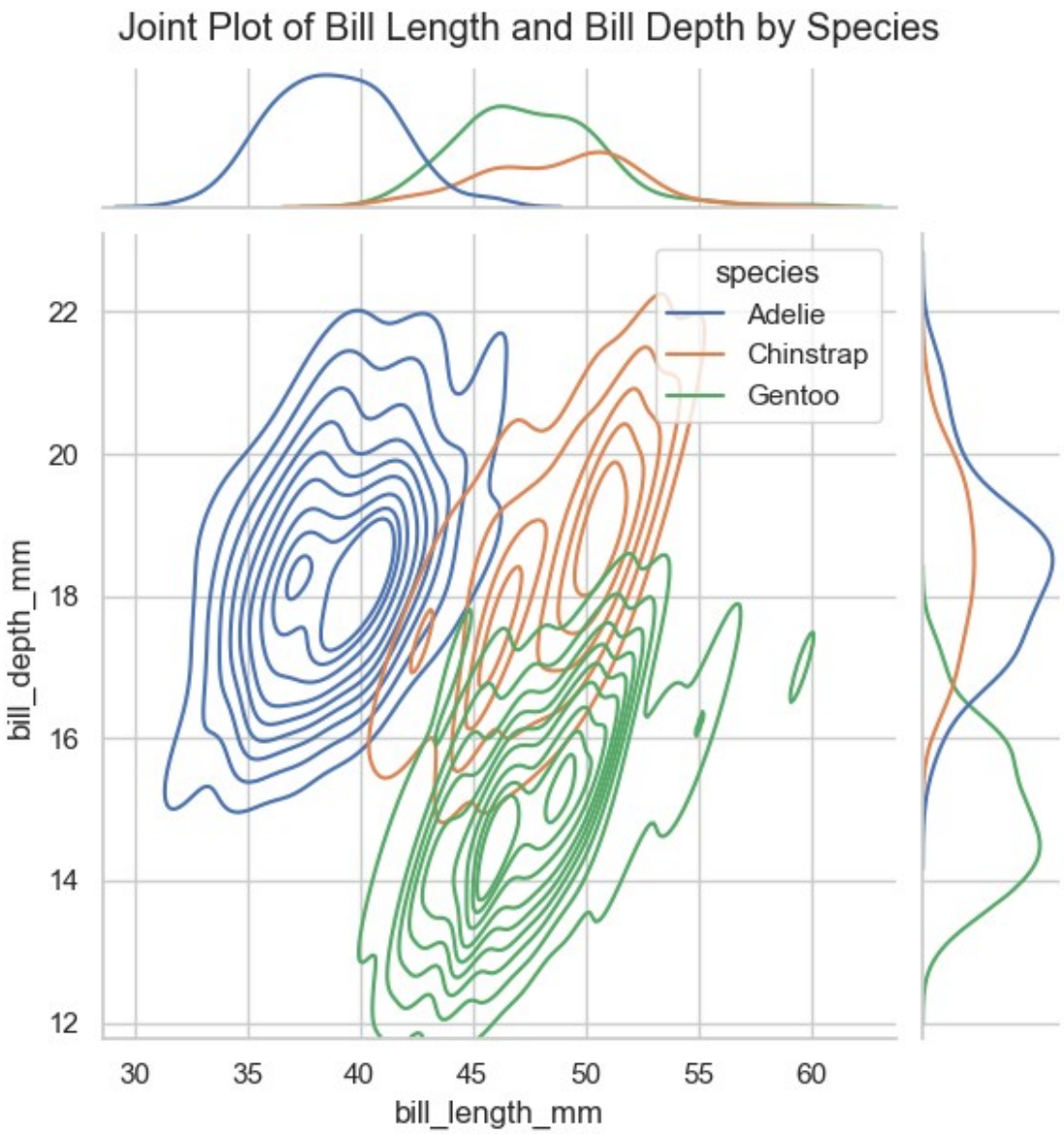


The pair plot gives a grid of scatter plots between each pair of variables, with KDEs on the diagonal. It's useful for spotting trends, correlations, or clusters.

11. Joint Plot

Joint plots show the relationship between two variables along with their distributions.

```
sns.jointplot(data=penguins, x="bill_length_mm", y="bill_depth_mm",
hue="species", kind="kde")
plt.suptitle("Joint Plot of Bill Length and Bill Depth by Species",
y=1.02)
plt.show()
```

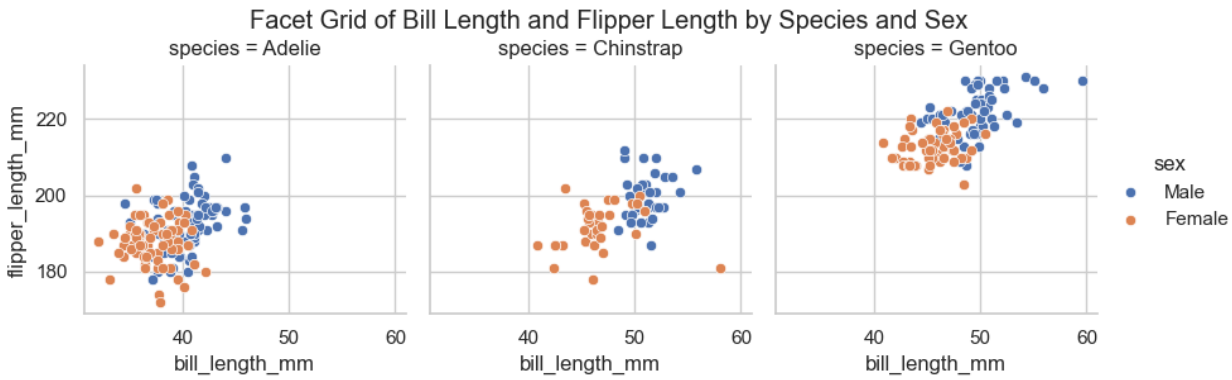


This joint plot combines a scatter plot of bill_length_mm vs bill_depth_mm with KDEs for each variable. It helps us visualize both the relationship and the distributions.

12. Facet Grid

A FacetGrid allows creating multiple plots across subsets of data, which is useful for visualizing categorical variables.

```
g = sns.FacetGrid(penguins, col="species", hue="sex")
g.map(sns.scatterplot, "bill_length_mm", "flipper_length_mm")
g.add_legend()
plt.suptitle("Facet Grid of Bill Length and Flipper Length by Species and Sex", y=1.02)
plt.show()
```

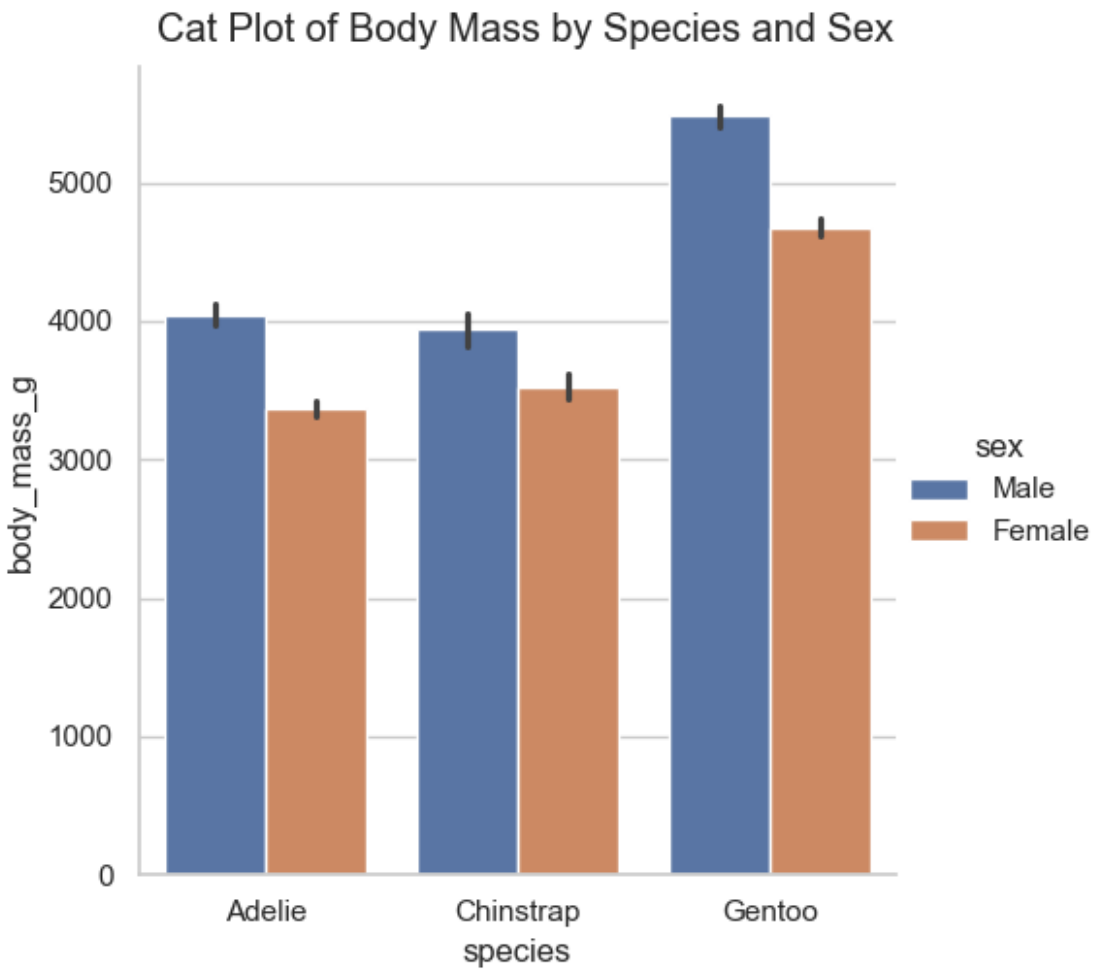


Here, we create separate scatter plots for each species. The FacetGrid provides a quick way to compare data across multiple categories.

13. Cat Plot

A categorical plot `(catplot)` is versatile for plotting categorical data types and can produce bar, point, and box plots in one command.

```
sns.catplot(data=penguins, x="species", y="body_mass_g", hue="sex",
kind="bar")
plt.suptitle("Cat Plot of Body Mass by Species and Sex", y=1.02)
plt.show()
```

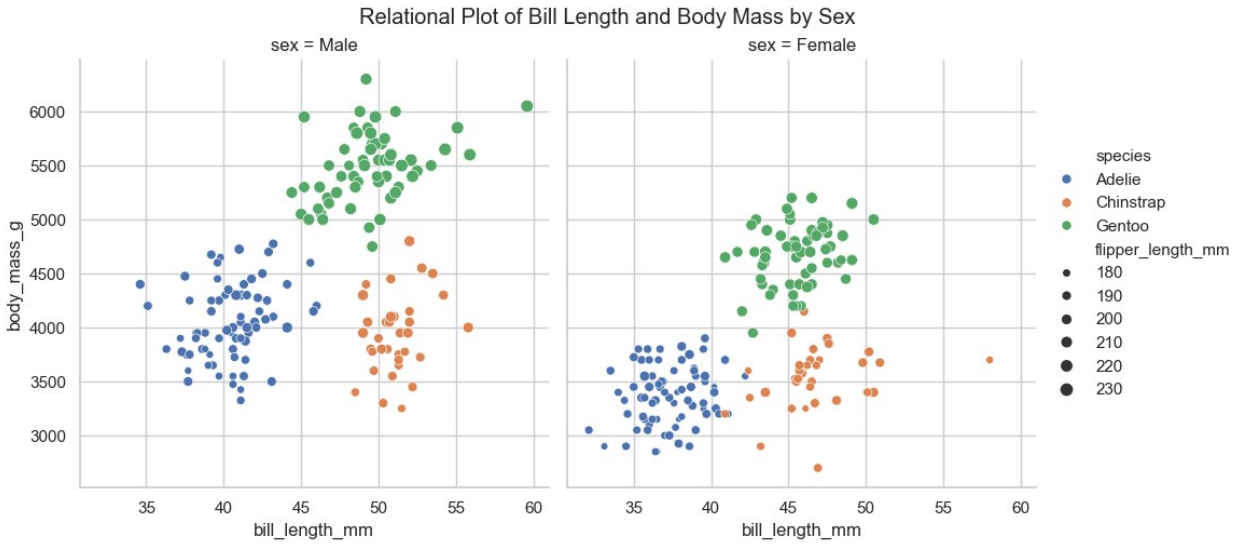


This bar plot (a type of catplot) shows body mass by species and sex. Catplot is useful when comparing different categorical plot types in a unified framework.

14. Rel Plot

Relplot is used to create both scatter and line plots in a facet grid.

```
sns.relplot(data=penguins, x="bill_length_mm", y="body_mass_g",
hue="species", size="flipper_length_mm", col="sex", kind="scatter")
plt.suptitle("Relational Plot of Bill Length and Body Mass by Sex",
y=1.02)
plt.show()
```



This relational plot shows bill length vs. body mass, with data split by sex into separate plots. It provides a visual summary of multiple dimensions, like species and flipper length, in one plot.