In [1]:

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
```

In [41]:

```
df = pd.read_csv("penguins_size.csv")
```

In [3]:

```
df.head()
```

Out[3]:

	species	island	culmen_length_mm	culmen_depth_mm	flipper_length_mm	body_mas
0	Adelie	Torgersen	39.1	18.7	181.0	37
1	Adelie	Torgersen	39.5	17.4	186.0	38
2	Adelie	Torgersen	40.3	18.0	195.0	32
3	Adelie	Torgersen	NaN	NaN	NaN	N
4	Adelie	Torgersen	36.7	19.3	193.0	34

In [4]:

```
df.isnull().sum()
```

Out[4]:

```
species               0
island                0
culmen_length_mm      2
culmen_depth_mm       2
flipper_length_mm     2
body_mass_g           2
sex                  10
dtype: int64
```

In [5]:

```
df.shape
```

Out[5]:

```
(344, 7)
```

In [6]:

```
df.dropna(inplace = True)
```

In [7]:

```
df.shape
```

Out[7]:

```
(334, 7)
```

In [8]:

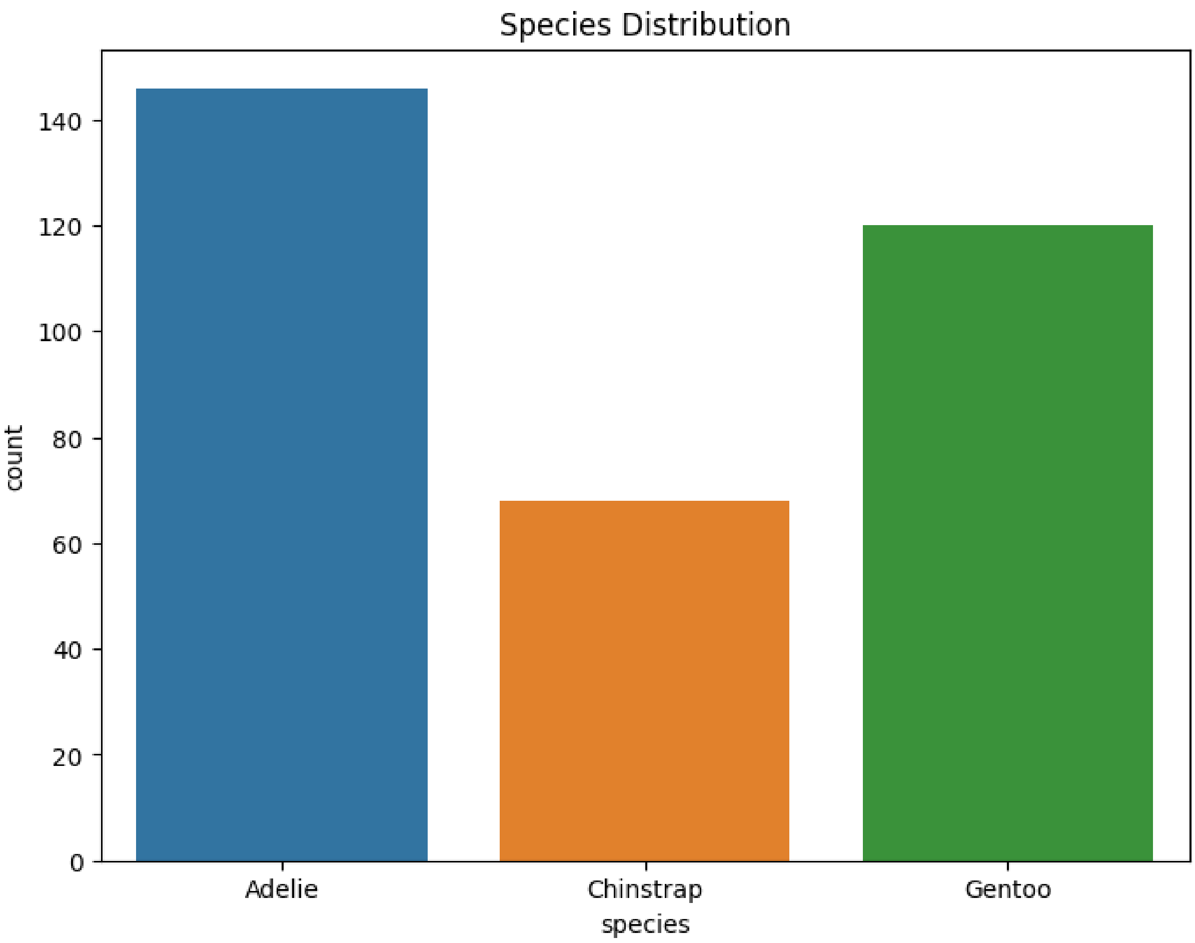
```
df.isnull().sum()
```

Out[8]:
```
species              0
island               0
culmen_length_mm     0
culmen_depth_mm      0
flipper_length_mm    0
body_mass_g          0
sex                  0
dtype: int64
```

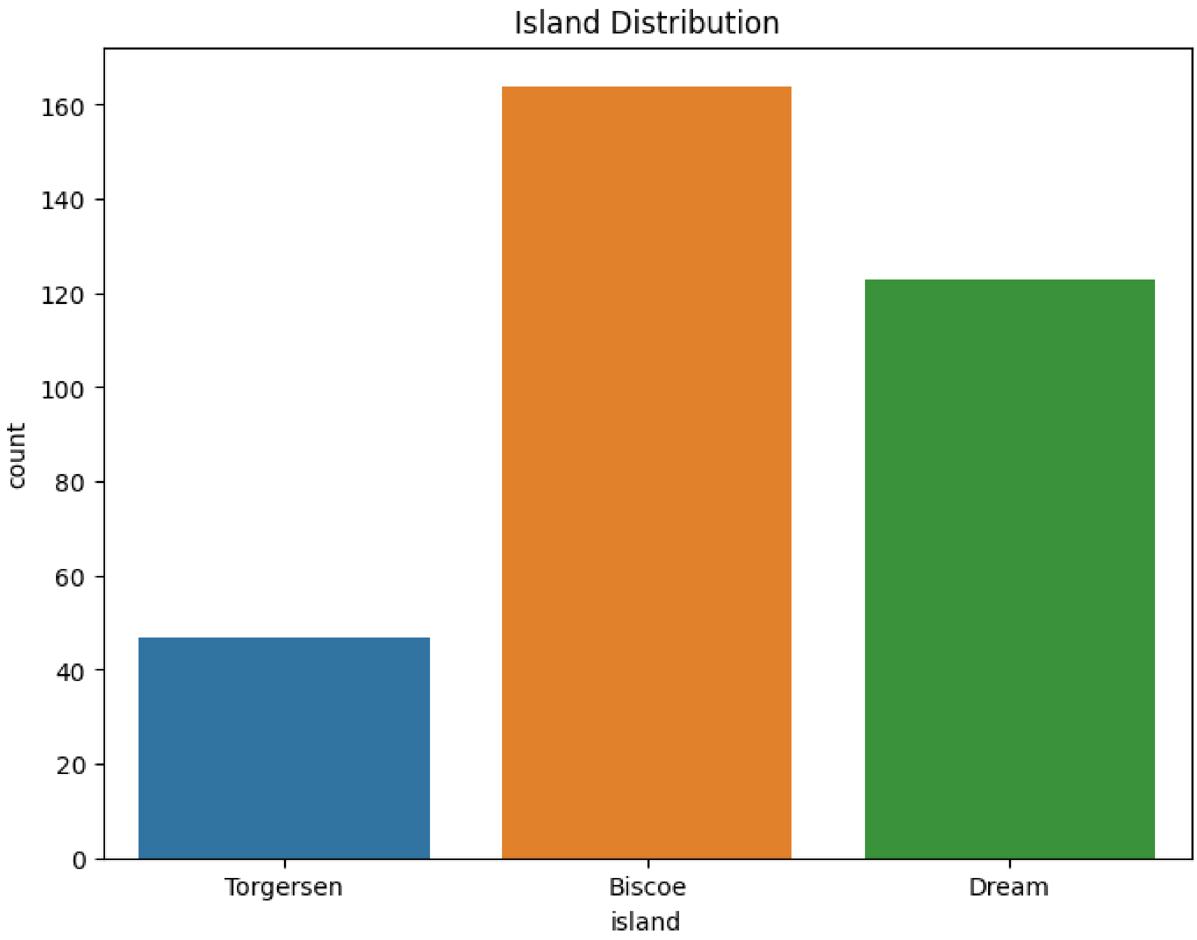
In [9]:
```
df.columns
```

Out[9]:
```
Index(['species', 'island', 'culmen_length_mm', 'culmen_depth_mm',
       'flipper_length_mm', 'body_mass_g', 'sex'],
      dtype='object')
```

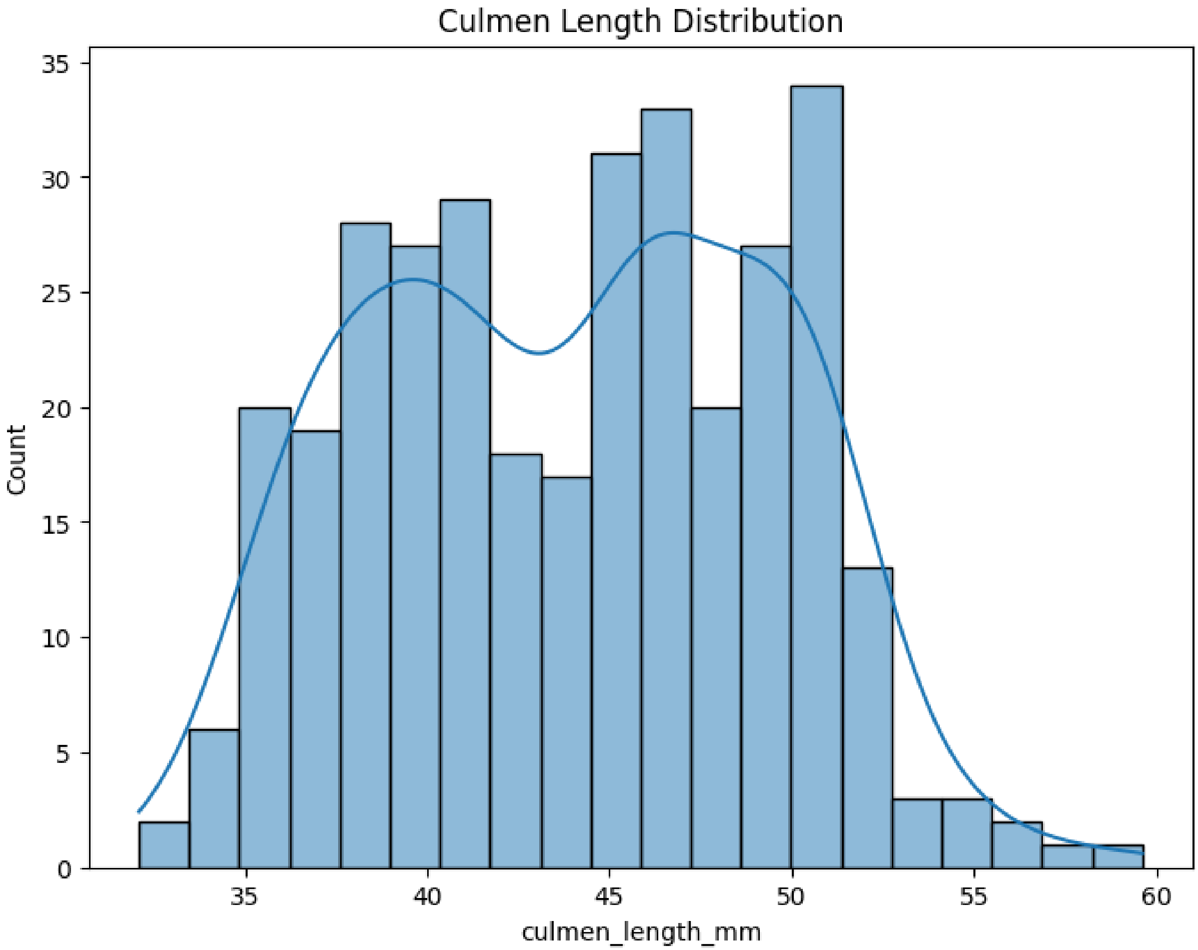
In [10]:
```
# Species Distribution
plt.figure(figsize=(8, 6))
sns.countplot(data=df, x='species')
plt.title('Species Distribution')
plt.show()
```



In [11]:
```
# Island Distribution
plt.figure(figsize=(8, 6))
sns.countplot(data=df, x='island')
plt.title('Island Distribution')
plt.show()
```

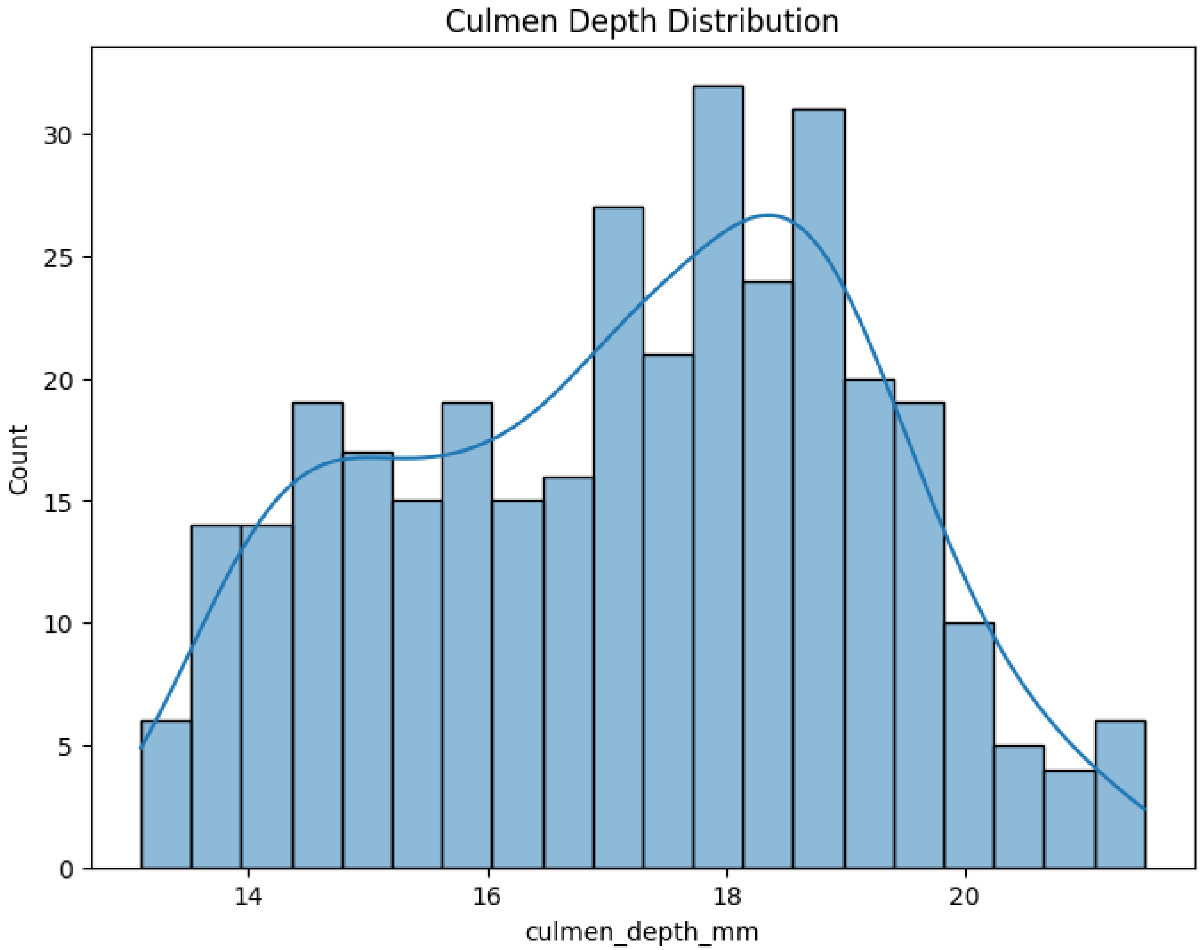


In [12]:
```
# Culmen Length Distribution
plt.figure(figsize=(8, 6))
sns.histplot(data=df, x='culmen_length_mm', bins=20, kde=True)
plt.title('Culmen Length Distribution')
plt.show()
```

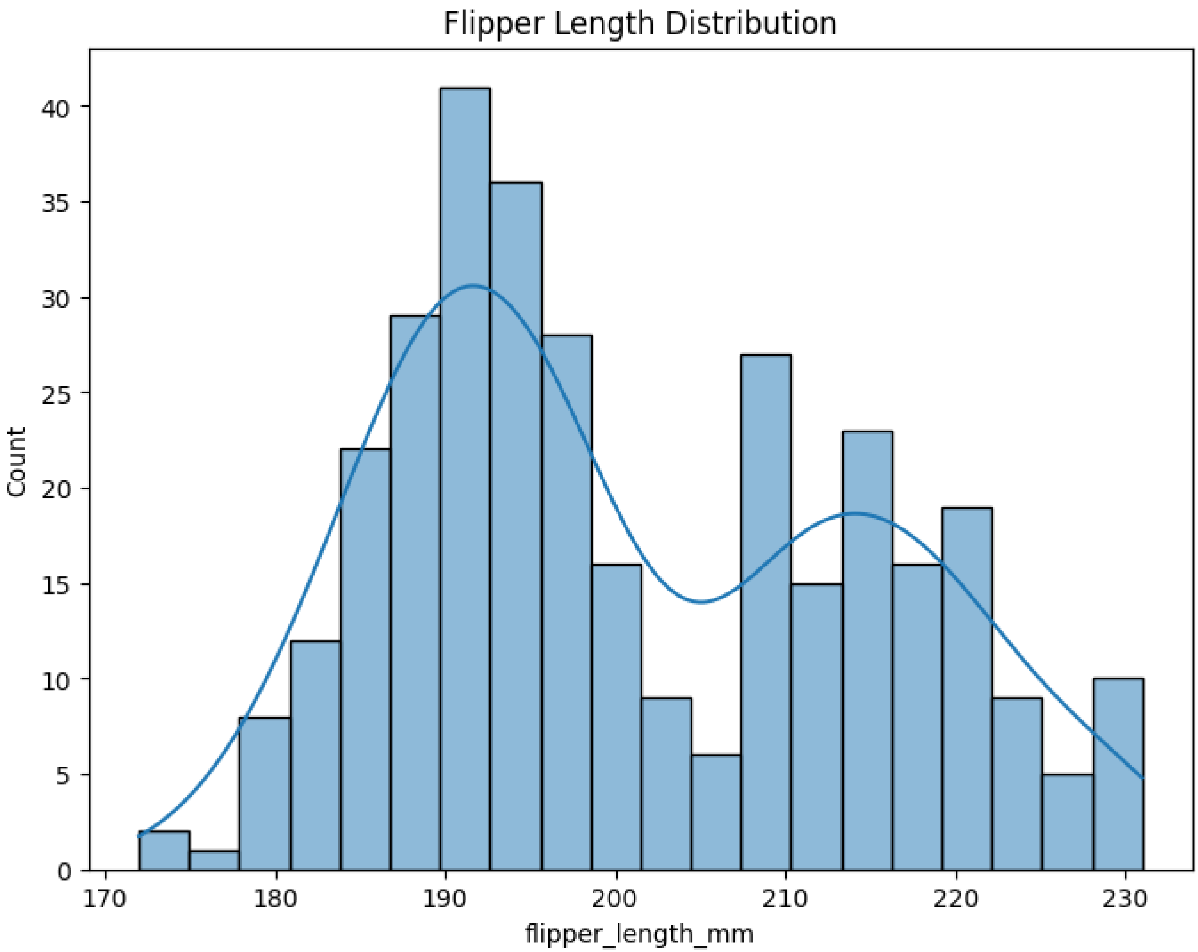


In [13]:

```
# Culmen Depth Distribution
plt.figure(figsize=(8, 6))
sns.histplot(data=df, x='culmen_depth_mm', bins=20, kde=True)
plt.title('Culmen Depth Distribution')
plt.show()
```

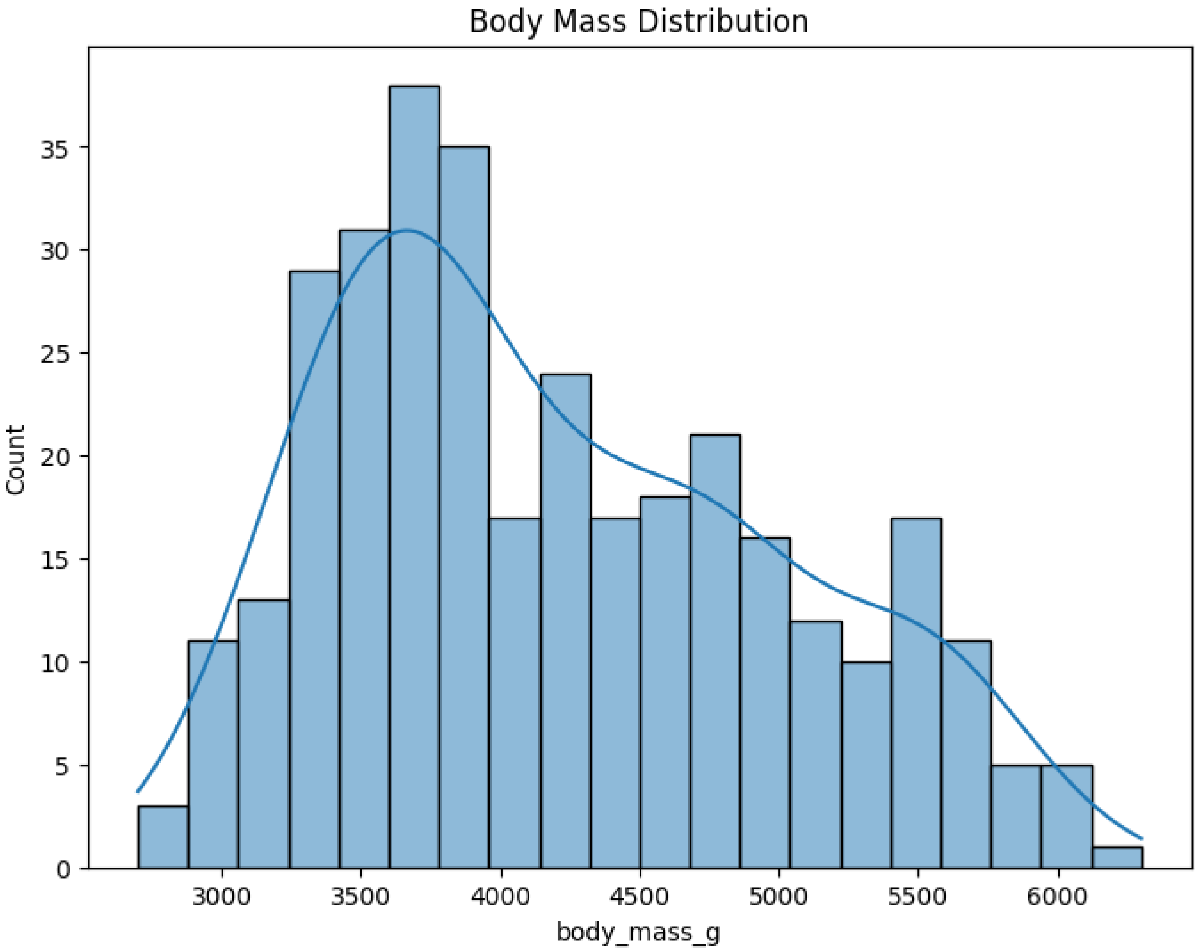


In [14]:
```
# Flipper Length Distribution
plt.figure(figsize=(8, 6))
sns.histplot(data=df, x='flipper_length_mm', bins=20, kde=True)
plt.title('Flipper Length Distribution')
plt.show()
```

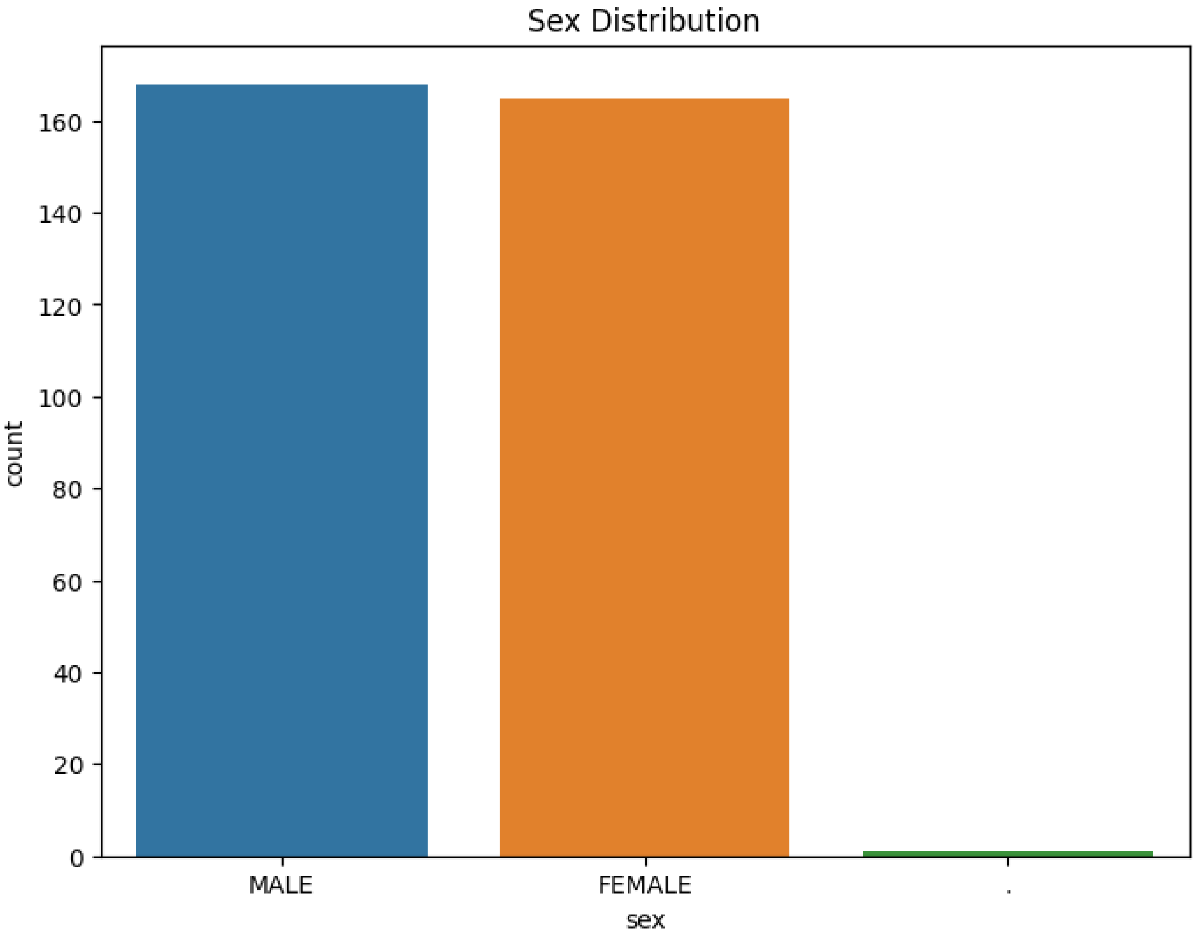


In [15]:

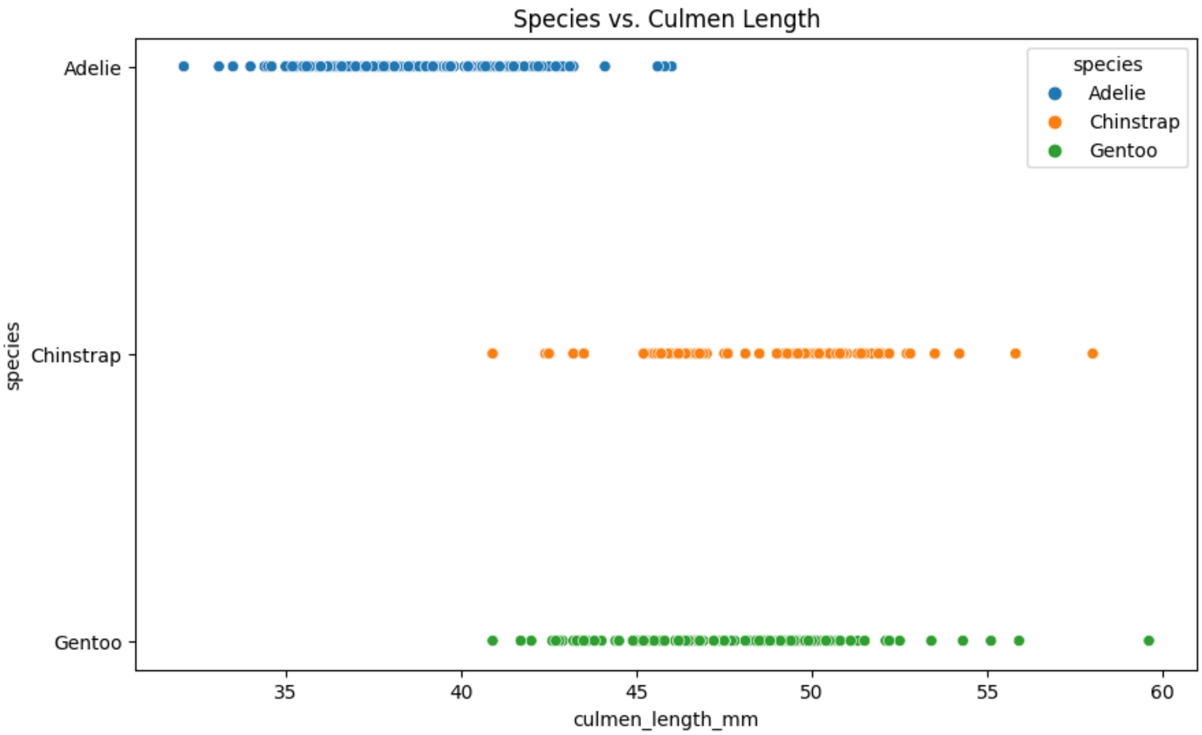
```
# Body Mass Distribution
plt.figure(figsize=(8, 6))
sns.histplot(data=df, x='body_mass_g', bins=20, kde=True)
plt.title('Body Mass Distribution')
plt.show()
```



In [16]:
```
# Sex Distribution
plt.figure(figsize=(8, 6))
sns.countplot(data=df, x='sex')
plt.title('Sex Distribution')
plt.show()
```

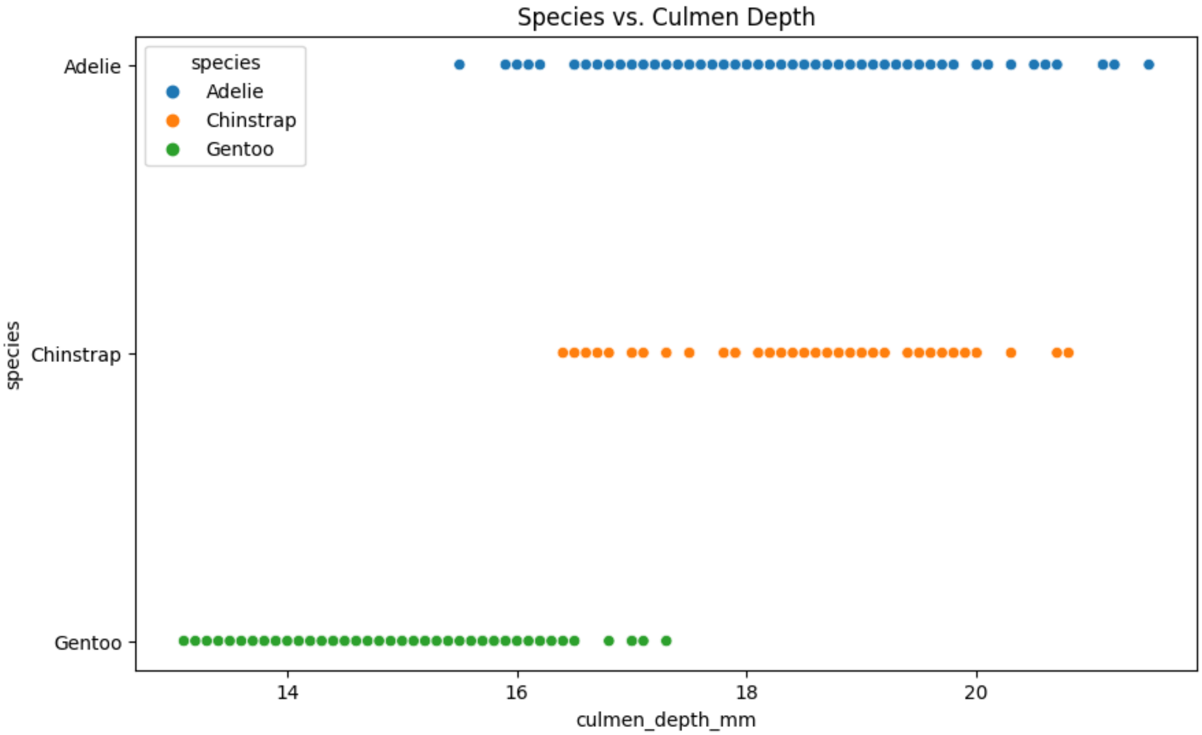


In [17]:
```
# Scatterplot of Species vs. Culmen Length
plt.figure(figsize=(10, 6))
sns.scatterplot(data=df, x='culmen_length_mm', y='species', hue='species')
plt.title('Species vs. Culmen Length')
plt.show()
```



In [18]:

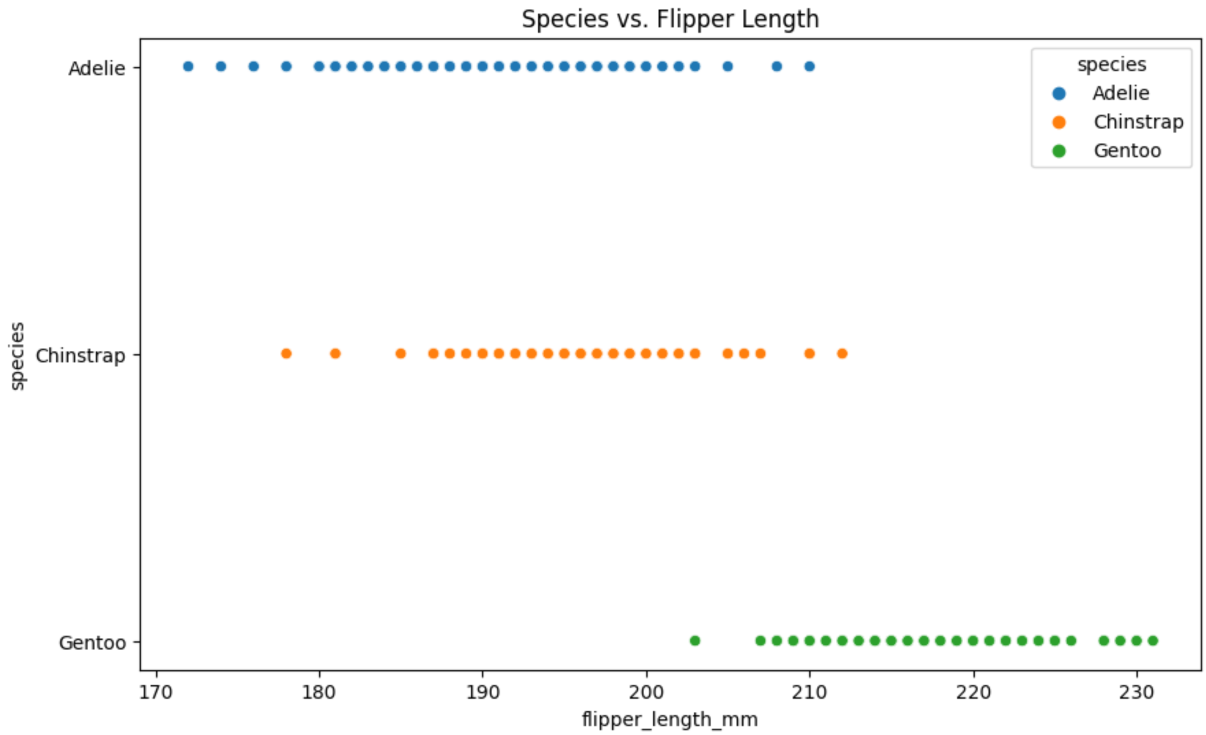
```
# Scatterplot of Species vs. Culmen Depth
plt.figure(figsize=(10, 6))
sns.scatterplot(data=df, x='culmen_depth_mm', y='species', hue='species')
plt.title('Species vs. Culmen Depth')
plt.show()
```



In [19]:

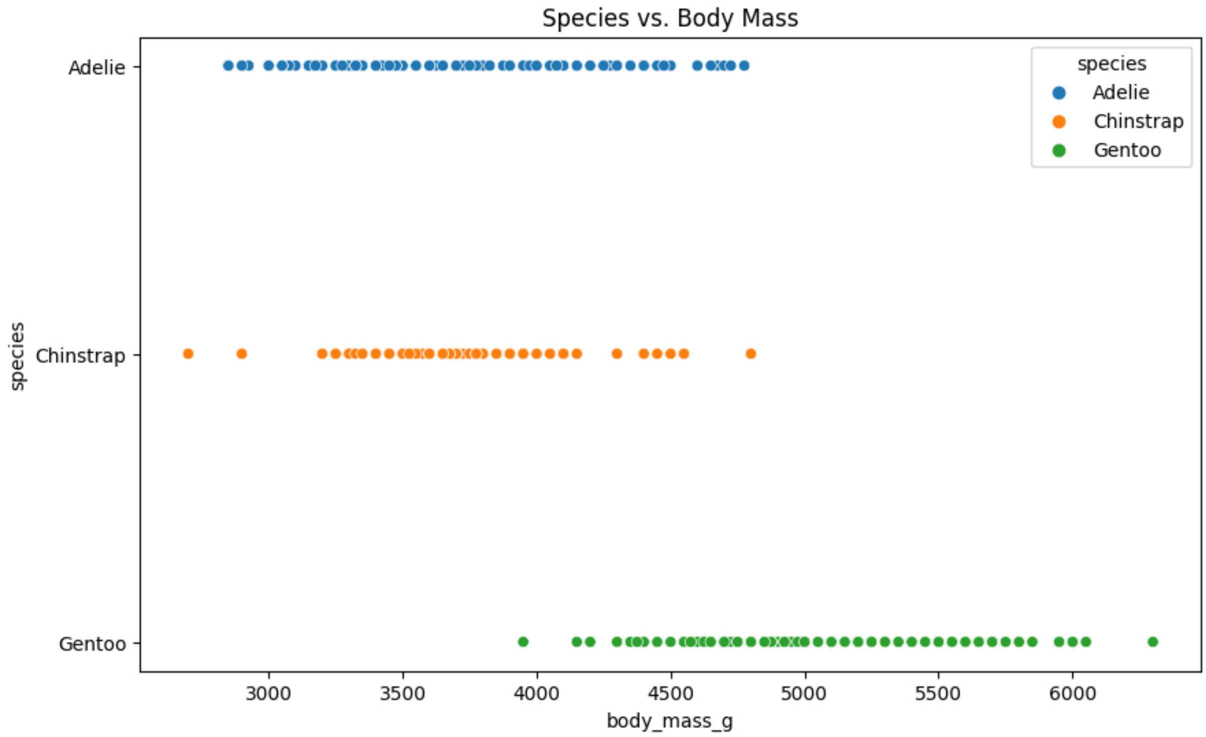
```
# Scatterplot of Species vs. Flipper Length
plt.figure(figsize=(10, 6))
sns.scatterplot(data=df, x='flipper_length_mm', y='species', hue='species')
```

```
plt.title('Species vs. Flipper Length')
plt.show()
```

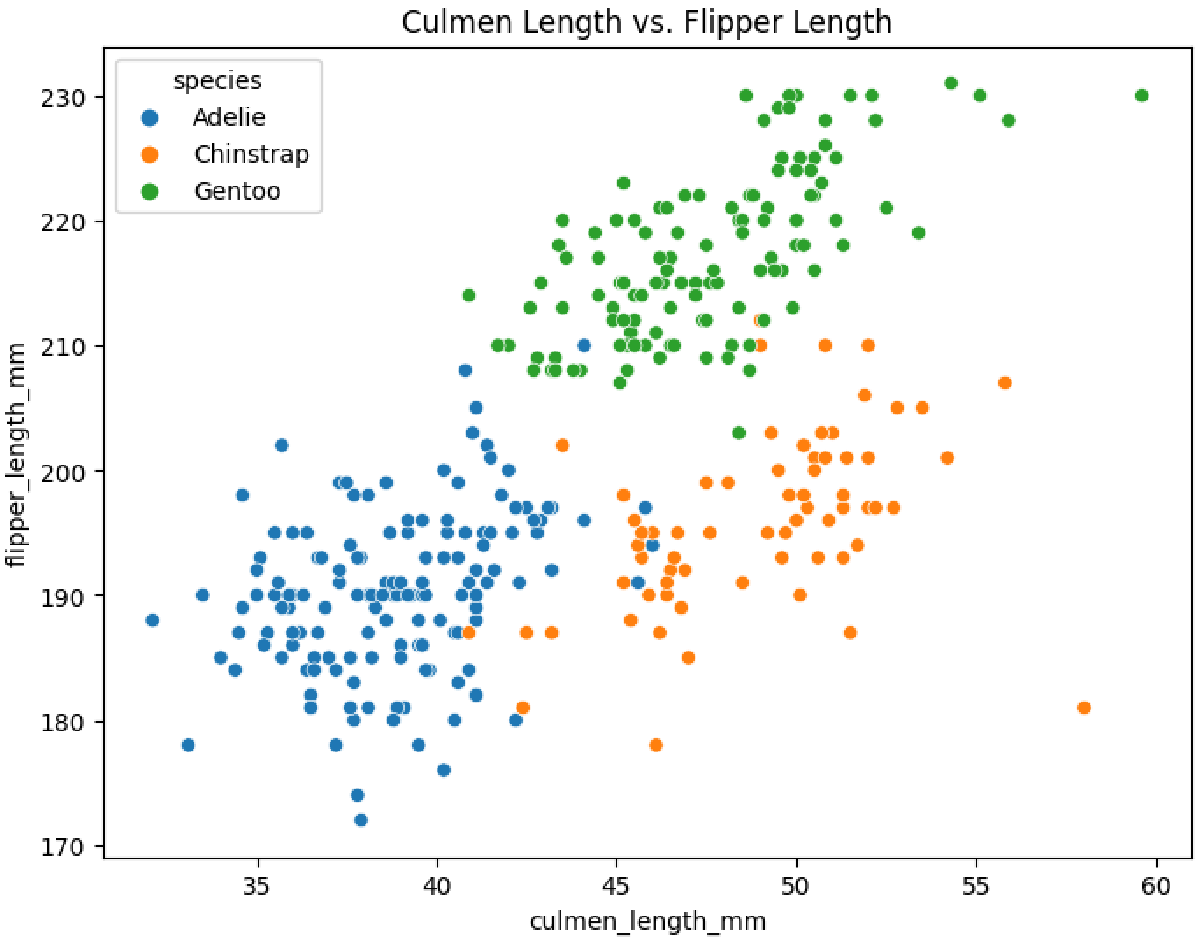


In [20]:

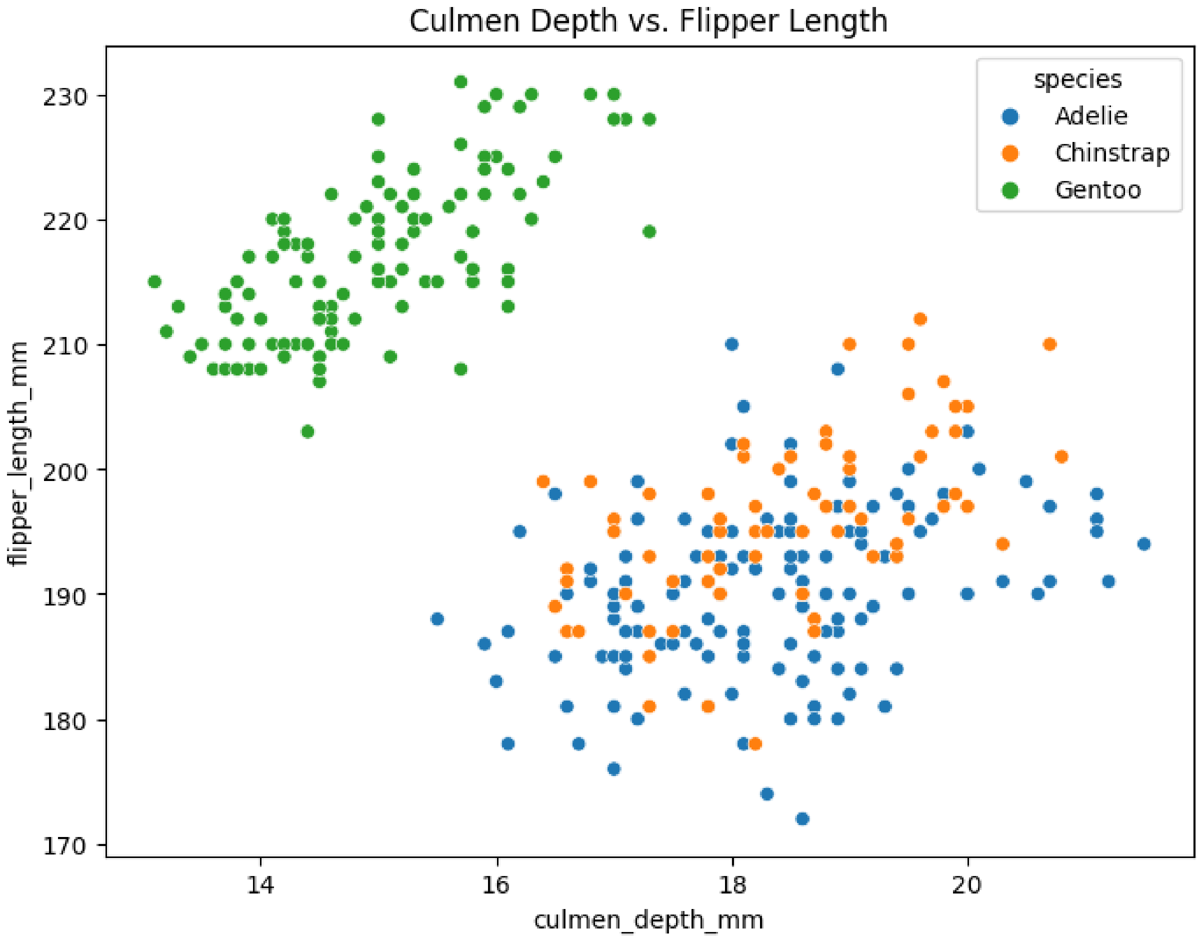
```
# Scatterplot of Species vs. Body Mass
plt.figure(figsize=(10, 6))
sns.scatterplot(data=df, x='body_mass_g', y='species', hue='species')
plt.title('Species vs. Body Mass')
plt.show()
```



In [21]:
```
# Scatterplot of Culmen Length vs. Flipper Length
plt.figure(figsize=(8, 6))
sns.scatterplot(data=df, x='culmen_length_mm', y='flipper_length_mm', hue='species'
plt.title('Culmen Length vs. Flipper Length')
plt.show()
```

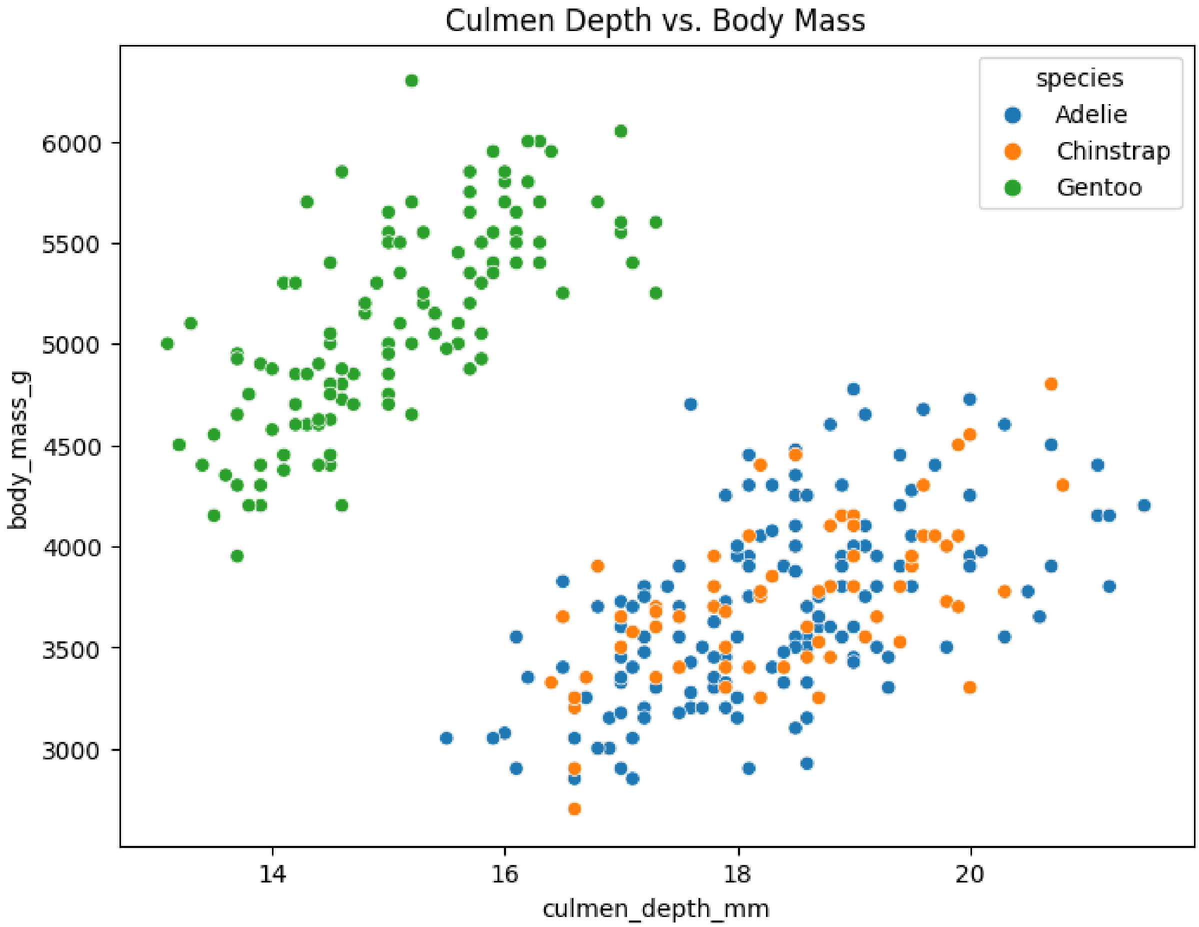


In [22]:
```
# Scatterplot of Culmen Depth vs. Flipper Length
plt.figure(figsize=(8, 6))
sns.scatterplot(data=df, x='culmen_depth_mm', y='flipper_length_mm', hue='species')
plt.title('Culmen Depth vs. Flipper Length')
plt.show()
```



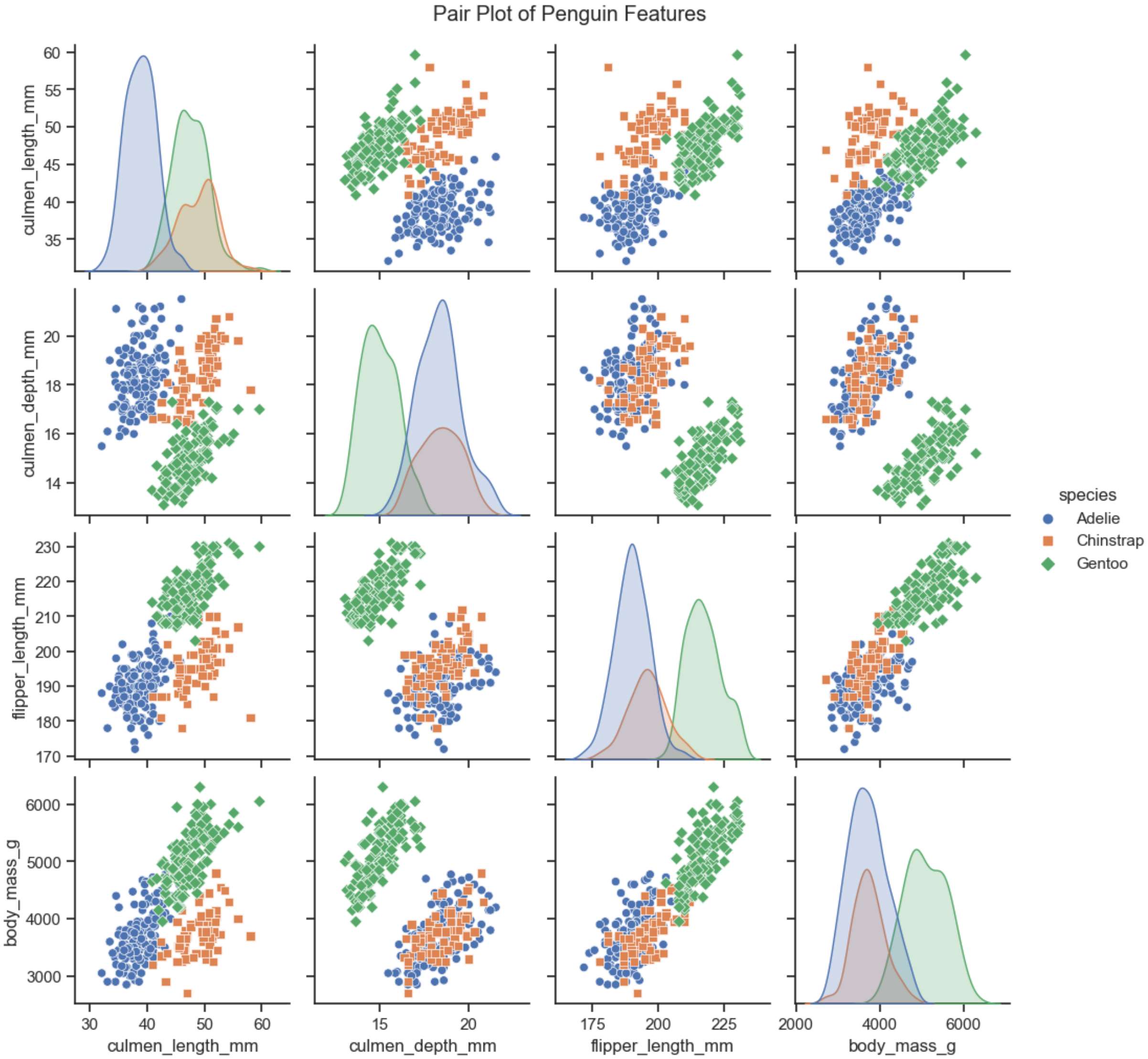
In [23]:

```
# Scatterplot of Culmen Depth vs. Body Mass
plt.figure(figsize=(8, 6))
sns.scatterplot(data=df, x='culmen_depth_mm', y='body_mass_g', hue='species')
plt.title('Culmen Depth vs. Body Mass')
plt.show()
```



In [24]:

```
# Pair plot for multiple variables
sns.set(style="ticks")
sns.pairplot(df, hue="species", markers=["o", "s", "D"])
plt.suptitle("Pair Plot of Penguin Features", y=1.02)
plt.show()
```



In [28]:

```
# Basic Descriptive Statistics
description = df.describe()

# Additional Descriptive Statistics
species_counts = df['species'].value_counts()
island_counts = df['island'].value_counts()
sex_counts = df['sex'].value_counts()

# Display the results
print("Basic Descriptive Statistics:")
print(description)

print("\nCounts of Penguins by Species:")
print(species_counts)

print("\nCounts of Penguins by Island:")
print(island_counts)

print("\nCounts of Penguins by Sex:")
print(sex_counts)
```

```
Basic Descriptive Statistics:
       culmen_length_mm  culmen_depth_mm  flipper_length_mm  body_mass_g
count        334.000000       334.000000         334.000000   334.000000
mean          43.994311        17.160479         201.014970  4209.056886
std            5.460521         1.967909          14.022175   804.836129
min           32.100000        13.100000         172.000000  2700.000000
25%           39.500000        15.600000         190.000000  3550.000000
50%           44.500000        17.300000         197.000000  4050.000000
75%           48.575000        18.700000         213.000000  4793.750000
max           59.600000        21.500000         231.000000  6300.000000

Counts of Penguins by Species:
species
Adelie       146
Gentoo       120
Chinstrap     68
Name: count, dtype: int64

Counts of Penguins by Island:
island
Biscoe       164
Dream        123
Torgersen     47
Name: count, dtype: int64

Counts of Penguins by Sex:
sex
MALE      168
FEMALE    165
.           1
Name: count, dtype: int64
```

In [29]:

```
df.isnull().sum()
```

Out[29]:

```
species              0
island               0
culmen_length_mm     0
culmen_depth_mm      0
flipper_length_mm    0
body_mass_g          0
sex                  0
dtype: int64
```

In [31]:

```
def detect_outliers(data):
    Q1 = data.quantile(0.25)
    Q3 = data.quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
    outliers = (data < lower_bound) | (data > upper_bound)
    return outliers

numerical_cols = ['culmen_length_mm', 'culmen_depth_mm', 'flipper_length_mm', 'body
outlier_mask = df[numerical_cols].apply(detect_outliers)

num_outliers = outlier_mask.sum()
```

```
print("Number of Outliers in Each Column:")
print(num_outliers)
```

```
Number of Outliers in Each Column:
culmen_length_mm     0
culmen_depth_mm      0
flipper_length_mm    0
body_mass_g          0
dtype: int64
```

In [32]:

```
categorical_cols = df.select_dtypes(include=['object']).columns

# Perform encoding for each categorical column
for col in categorical_cols:
    if df[col].nunique() <= 2:
        # For binary (2-level) categorical variables, use label encoding
        df[col] = df[col].astype('category')
        df[col] = df[col].cat.codes
    else:
        # For nominal categorical variables, use one-hot encoding
        df = pd.get_dummies(df, columns=[col], prefix=[col])

# Display the encoded DataFrame
print(df.head())
```

```
   culmen_length_mm  culmen_depth_mm  flipper_length_mm  body_mass_g  
0              39.1             18.7              181.0       3750.0  \
1              39.5             17.4              186.0       3800.0   
2              40.3             18.0              195.0       3250.0   
4              36.7             19.3              193.0       3450.0   
5              39.3             20.6              190.0       3650.0   

   species_Adelie  species_Chinstrap  species_Gentoo  island_Biscoe  
0            True              False           False          False  \
1            True              False           False          False   
2            True              False           False          False   
4            True              False           False          False   
5            True              False           False          False   

   island_Dream  island_Torgersen  sex_.  sex_FEMALE  sex_MALE  
0         False              True  False       False      True  
1         False              True  False        True     False  
2         False              True  False        True     False  
4         False              True  False        True     False  
5         False              True  False       False      True  
```

In [42]:

```
from sklearn.model_selection import train_test_split

# Split the data into training and testing sets
target = df['species']
X_train, X_test, y_train, y_test = train_test_split(df, target, test_size=0.2, rand

# Display the shapes of the resulting sets
print("X_train shape:", X_train.shape)
print("X_test shape:", X_test.shape)
print("y_train shape:", y_train.shape)
print("y_test shape:", y_test.shape)
```

```
X_train shape: (275, 7)
X_test shape: (69, 7)
y_train shape: (275,)
y_test shape: (69,)
```

In [ ]:

In [ ]: