```
#  Import libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore", category=FutureWarning)

print("Hello Colab!")
```

```
Hello Colab!
```

```
# For better visuals
sns.set_style("whitegrid")
plt.rcParams['figure.figsize'] = (10, 6)
```

```
# Pandas display settings to show all columns in one line
pd.set_option('display.max_columns', None)   # Show all columns
pd.set_option('display.width', None)         # Don't wrap to next line
pd.set_option('display.max_colwidth', None)  # Show full content in each cell
```

```
#  Load dataset
df = pd.read_csv("penguins.csv")
```

```
# Basic exploration
print("\n--- First 5 Rows ---")
print(df.head())

print("\n--- Shape of Dataset ---")
print(df.shape)

print("\n--- Data Info ---")
print(df.info())

print("\n--- Statistical Summary ---")
print(df.describe())
```

```
--- First 5 Rows ---
   id species     island  bill_length_mm  bill_depth_mm  flipper_length_mm  \
0   0  Adelie  Torgersen            39.1           18.7              181.0
1   1  Adelie  Torgersen            39.5           17.4              186.0
2   2  Adelie  Torgersen            40.3           18.0              195.0
3   3  Adelie  Torgersen             NaN            NaN                NaN
4   4  Adelie  Torgersen            36.7           19.3              193.0

   body_mass_g     sex  year
0       3750.0    male  2007
1       3800.0  female  2007
2       3250.0  female  2007
3          NaN     NaN  2007
4       3450.0  female  2007
```

```
--- Shape of Dataset ---
(344, 9)

--- Data Info ---
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 344 entries, 0 to 343
Data columns (total 9 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   id                 344 non-null    int64  
 1   species            344 non-null    object 
 2   island             344 non-null    object 
 3   bill_length_mm     342 non-null    float64
 4   bill_depth_mm      342 non-null    float64
 5   flipper_length_mm  342 non-null    float64
 6   body_mass_g        342 non-null    float64
 7   sex                333 non-null    object 
 8   year               344 non-null    int64  
dtypes: float64(4), int64(2), object(3)
memory usage: 24.3+ KB
None

--- Statistical Summary ---
               id  bill_length_mm  bill_depth_mm  flipper_length_mm  \
count  344.000000      342.000000     342.000000         342.000000   
mean   171.500000       43.921930      17.151170         200.915205   
std     99.448479        5.459584       1.974793          14.061714   
min      0.000000       32.100000      13.100000         172.000000   
25%     85.750000       39.225000      15.600000         190.000000   
50%    171.500000       44.450000      17.300000         197.000000   
75%    257.250000       48.500000      18.700000         213.000000   
max    343.000000       59.600000      21.500000         231.000000   

       body_mass_g         year  
count   342.000000   344.000000  
mean   4201.754386  2008.029070  
std     801.954536     0.818356  
min    2700.000000  2007.000000  
25%    3550.000000  2007.000000  
50%    4050.000000  2008.000000  
75%    4750.000000  2009.000000  
max    6300.000000  2009.000000  
```

```
# Missing values
print("\n--- Missing Values ---")
print(df.isnull().sum())
```

```
--- Missing Values ---
id                    0
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
```

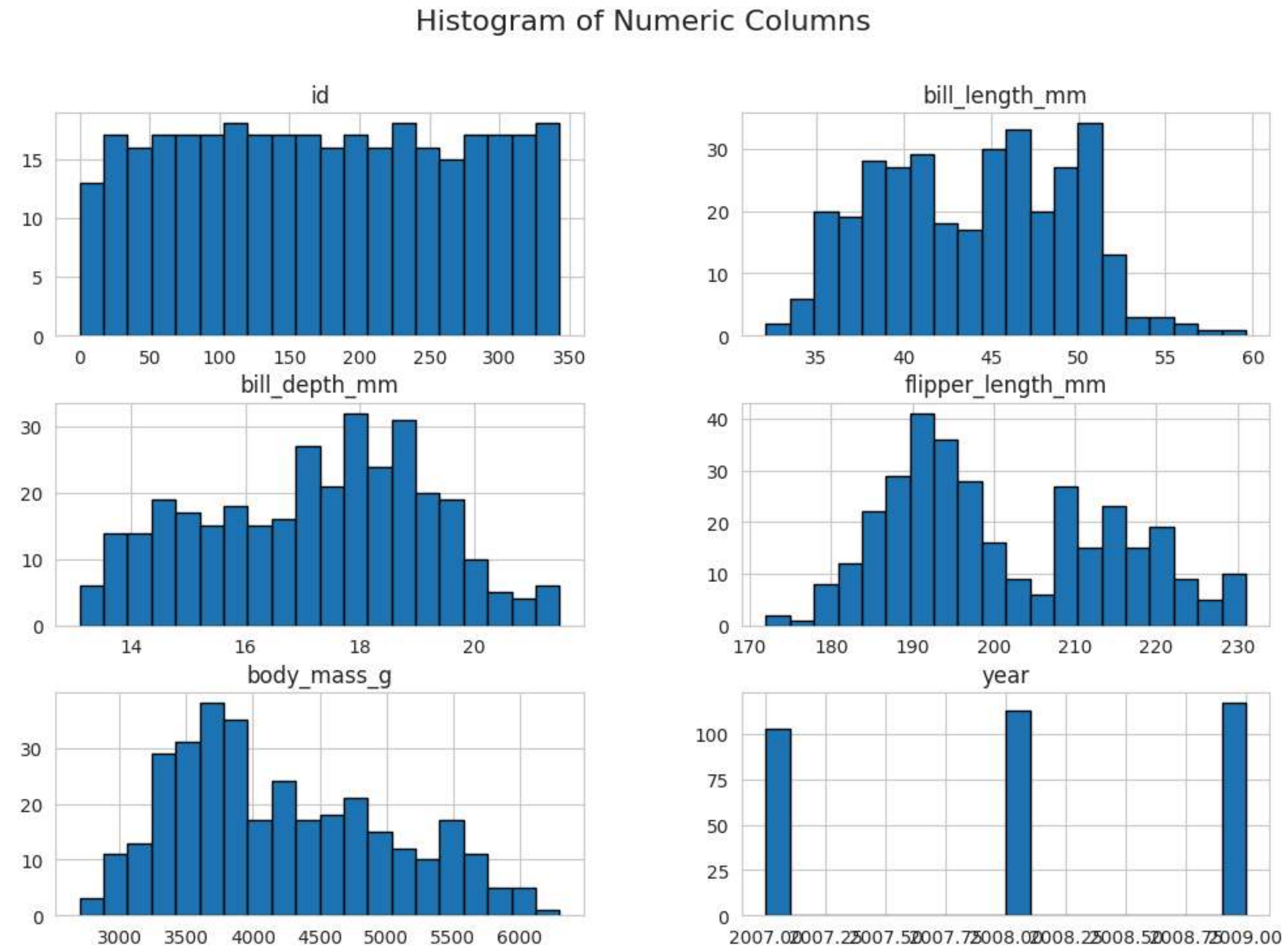
```
#  Unique values in categorical columns
print("\n--- Unique Categorical Values ---")
for col in df.select_dtypes(include='object'):
    print(f"{col}: {df[col].unique()}")
```

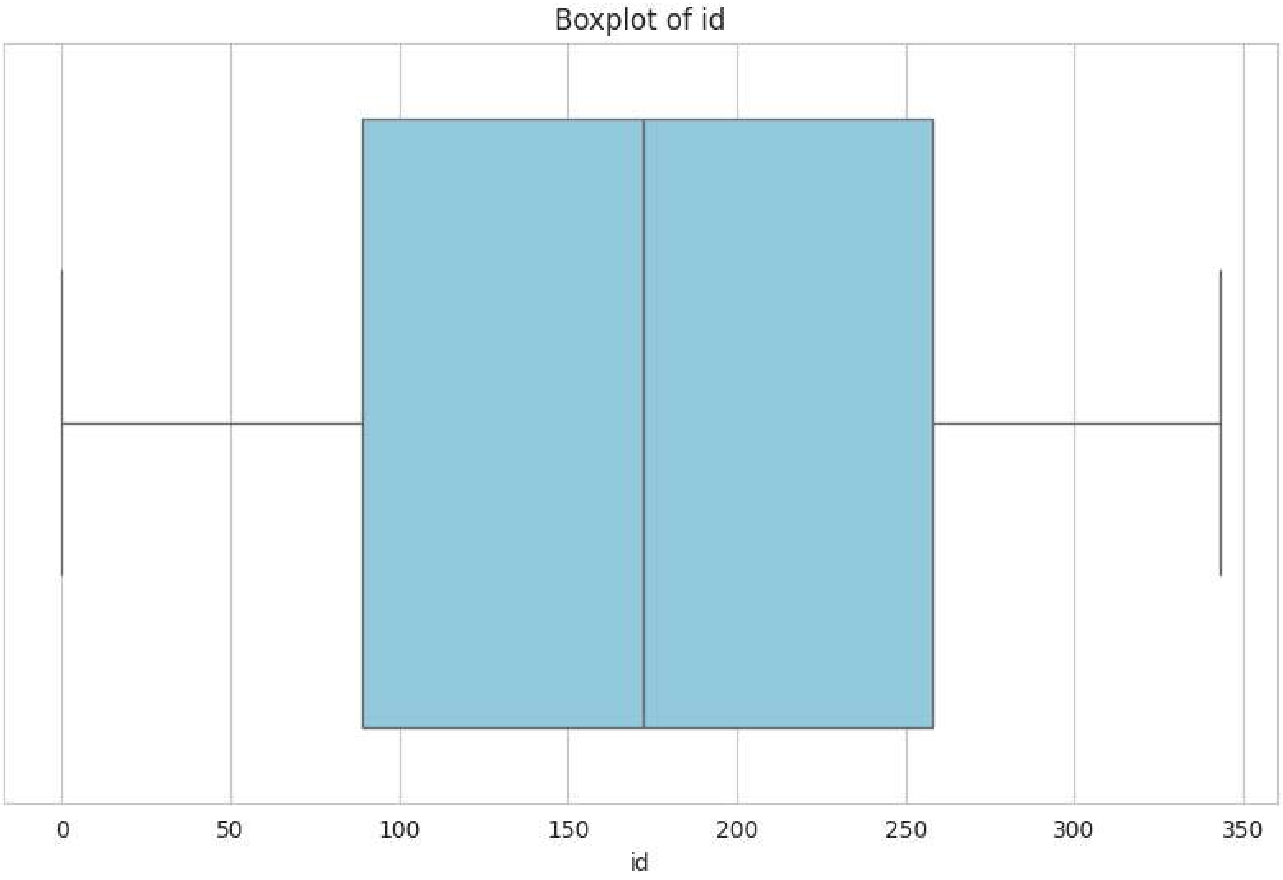
```

--- Unique Categorical Values ---
species: ['Adelie' 'Gentoo' 'Chinstrap']
island: ['Torgersen' 'Biscoe' 'Dream']
sex: ['male' 'female' nan]
```

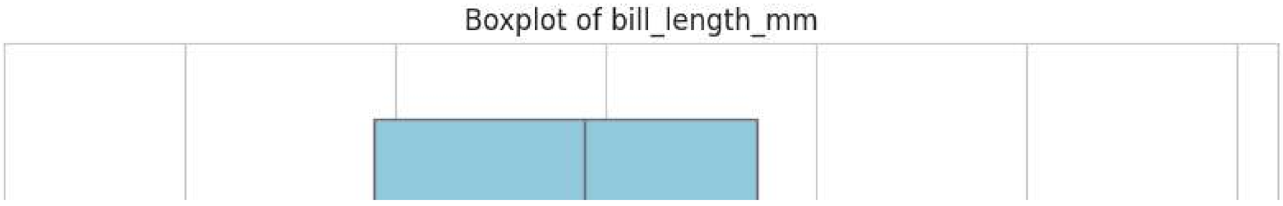
```
#  Data Cleaning
# Drop rows with missing values
df.dropna(inplace=True)


# Check duplicates
duplicates = df.duplicated().sum()
print(f"\n--- Duplicate Rows: {duplicates} ---")
```

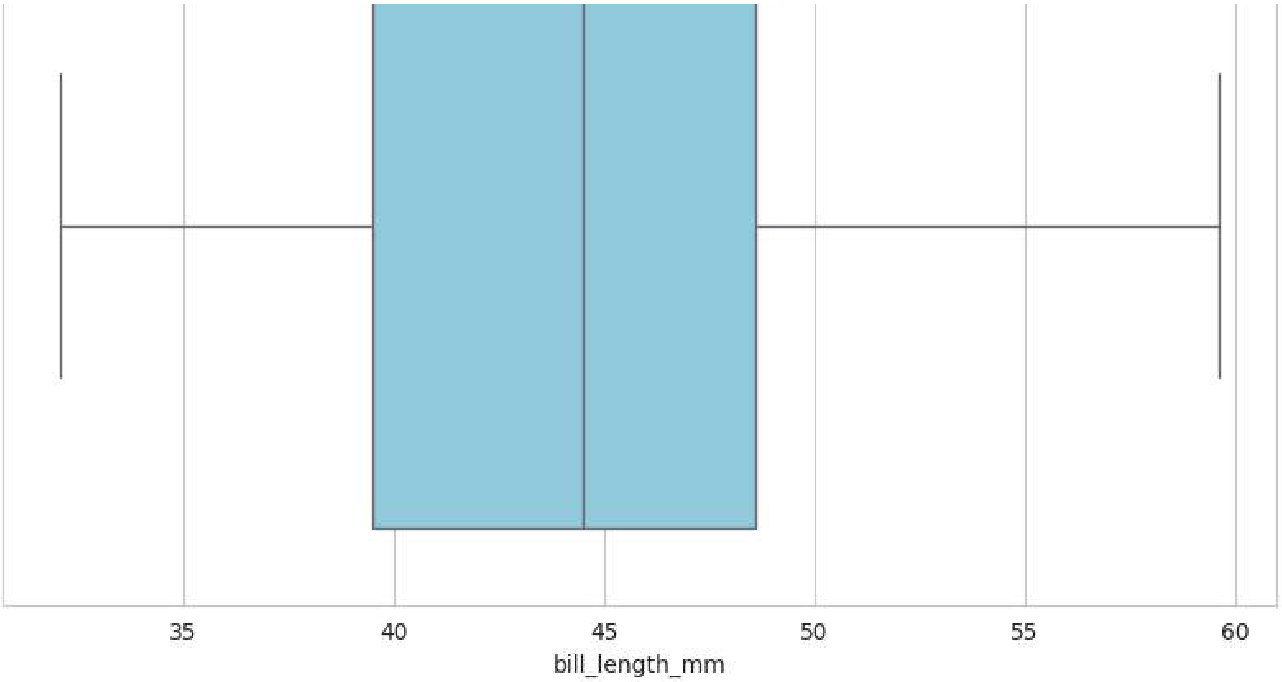
```

--- Duplicate Rows: 0 ---
```

```
#  Univariate Analysis
# Histograms for numeric columns
df.hist(figsize=(12, 8), bins=20, edgecolor='black')
plt.suptitle("Histogram of Numeric Columns", fontsize=16)
plt.show()

# Boxplots for numeric columns
for col in df.select_dtypes(include=np.number):
    sns.boxplot(x=df[col], hue=None, color='skyblue')
    plt.title(f"Boxplot of {col}")
    plt.show()

# Countplots for categorical columns
for col in df.select_dtypes(include='object'):
    sns.countplot(x=col, hue=None, data=df, palette="pastel")
    plt.title(f"Countplot of {col}")
    plt.show()
```

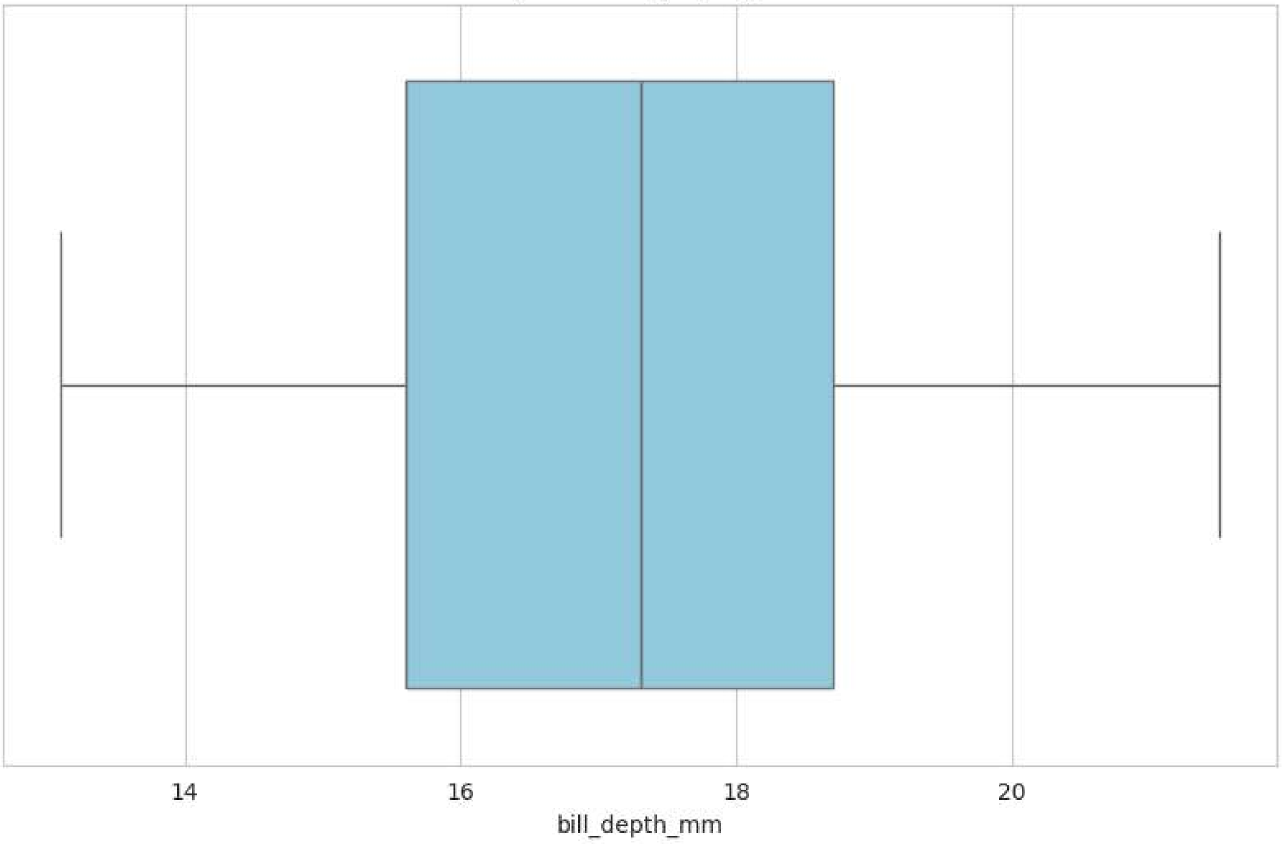

Histogram of Numeric Columns
id
0
5
10
15
0
50
100
150
200
250
300
350
bill_length_mm
0
10
20
30
35
40
45
50
55
60
bill_depth_mm
0
10
20
30
14
16
18
20
flipper_length_mm
0
10
20
30
40
170
180
190
200
210
220
230
body_mass_g
0
10
20
30
3000
3500
4000
4500
5000
5500
6000
year
0
25
50
75
100
2007.00
2007.25
2007.50
2007.75
2008.00
2008.25
2008.50
2008.75
2009.00


Boxplot of id
0
50
100
150
200
250
300
350
id

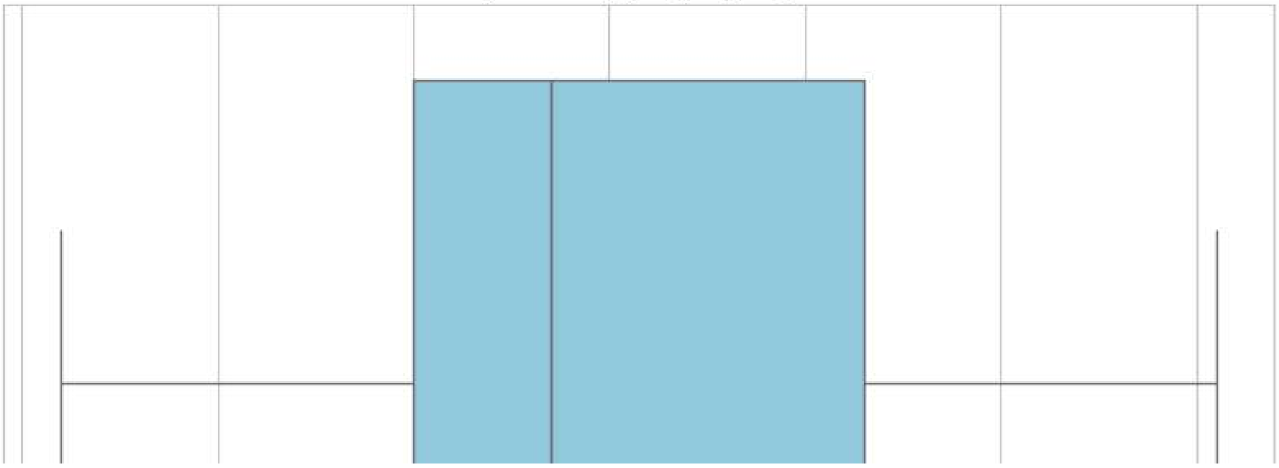

Boxplot of bill_length_mm

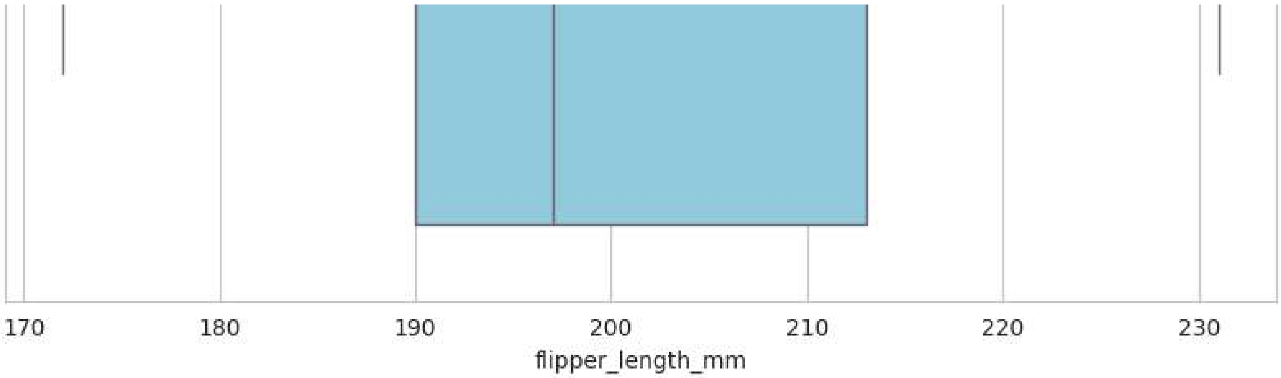


Boxplot of bill_depth_mm

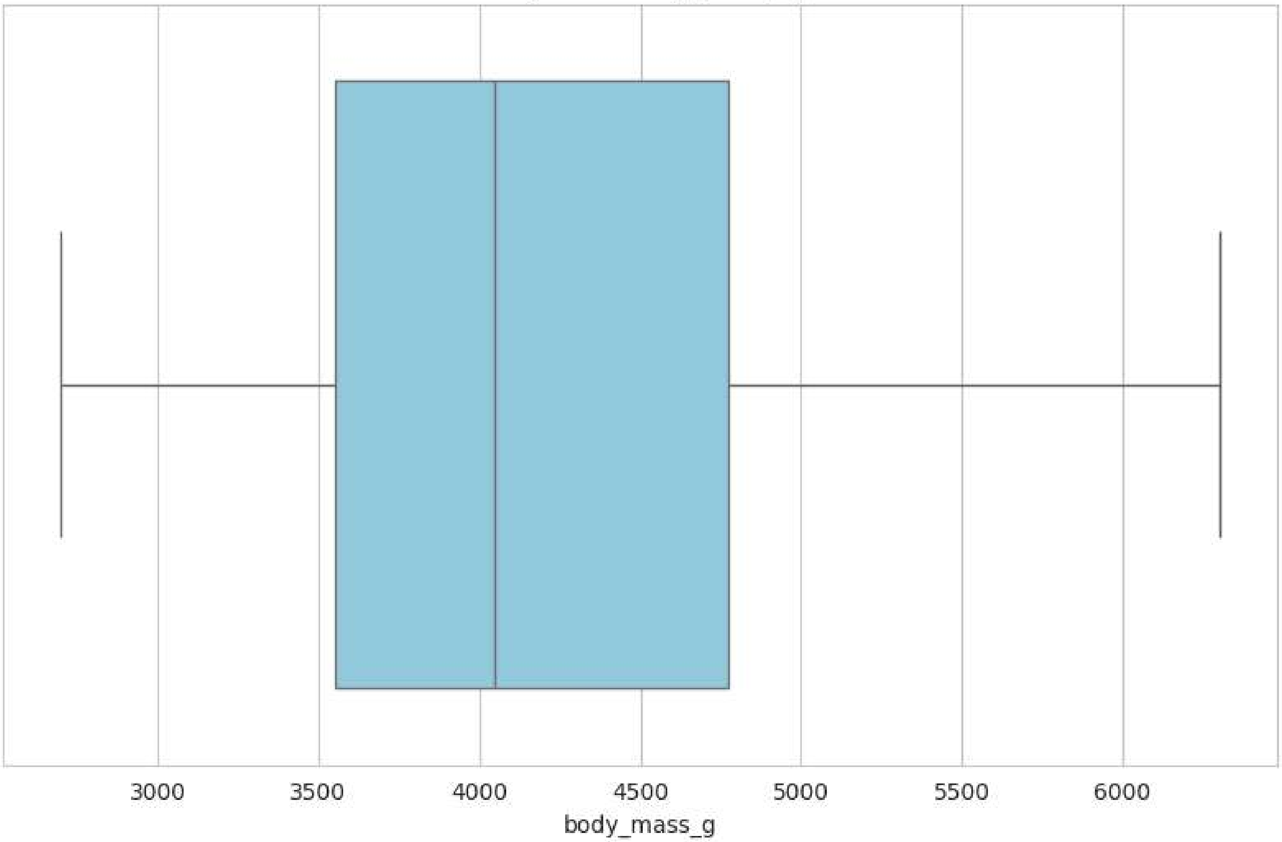


Boxplot of flipper_length_mm

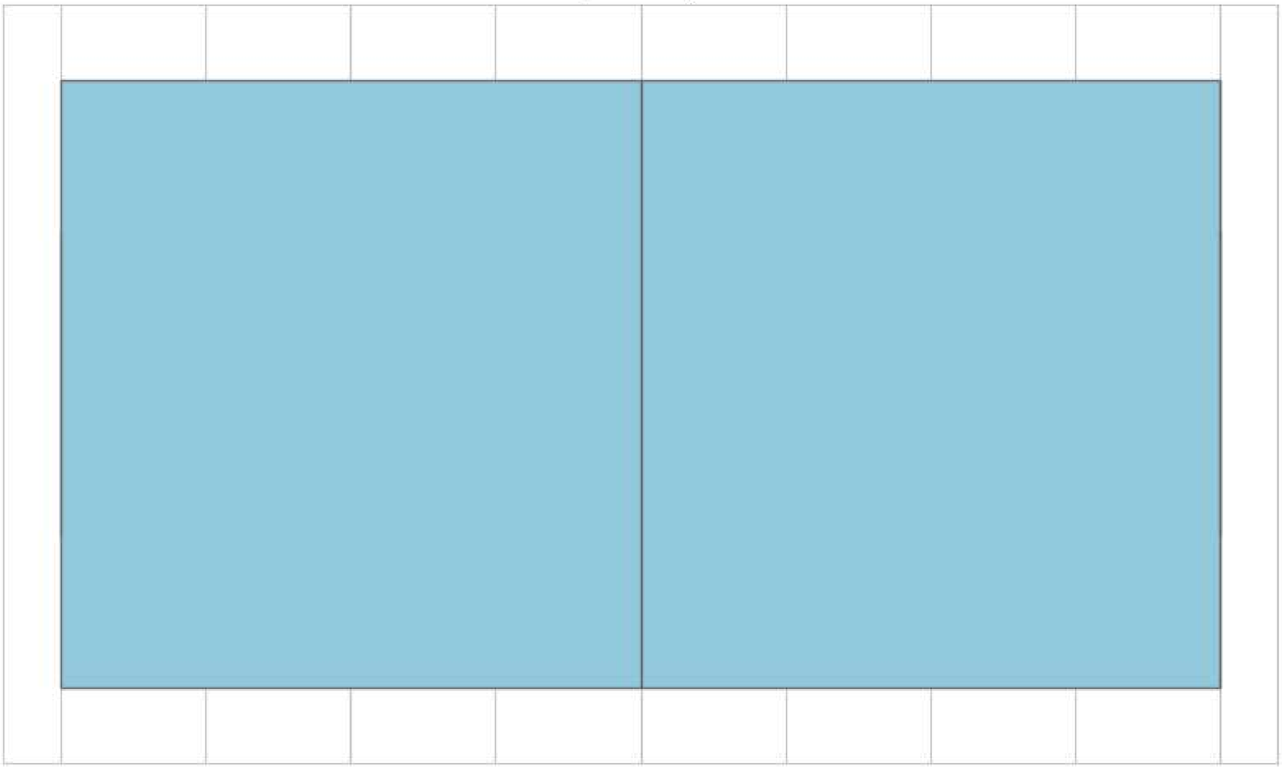


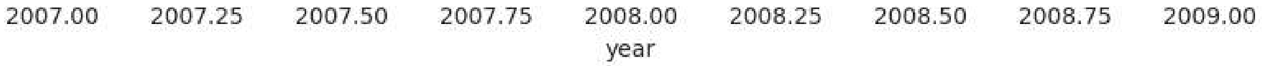


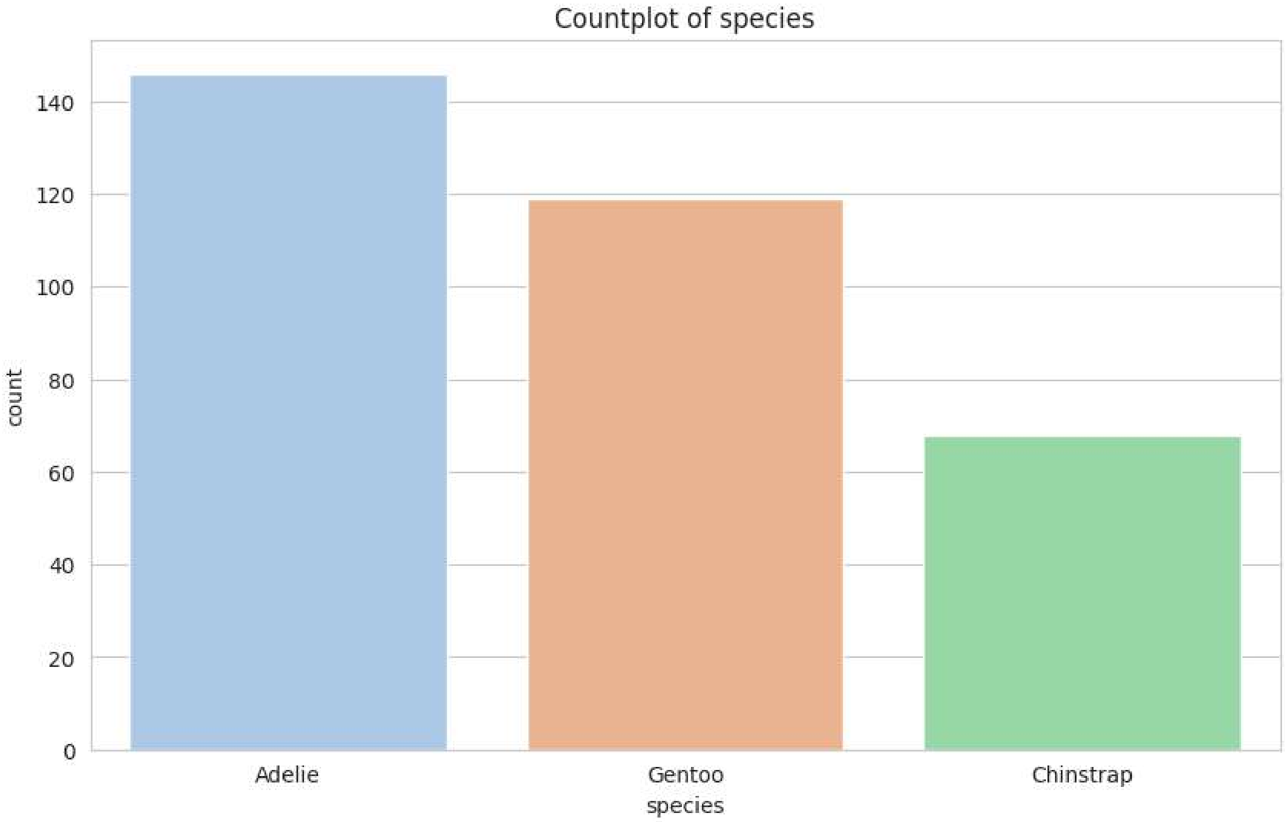
Boxplot of body_mass_g

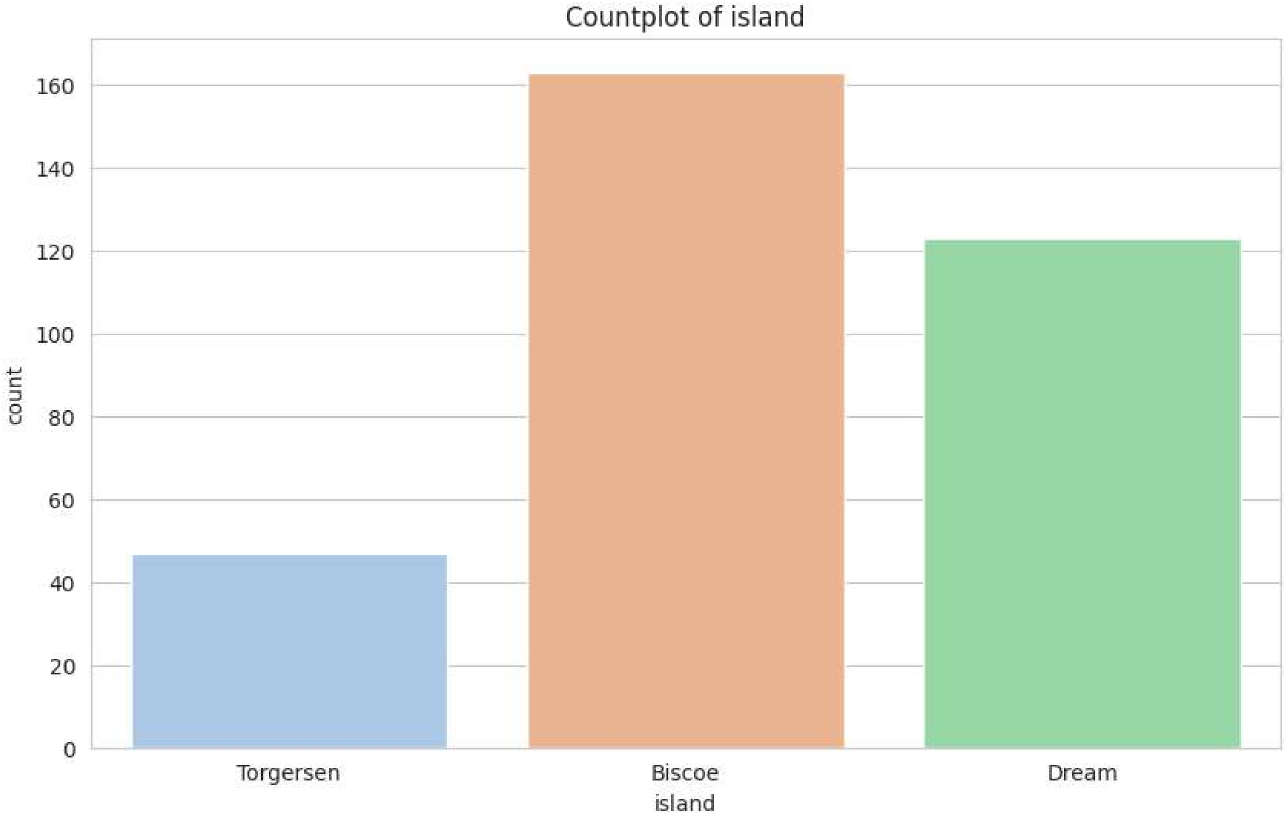


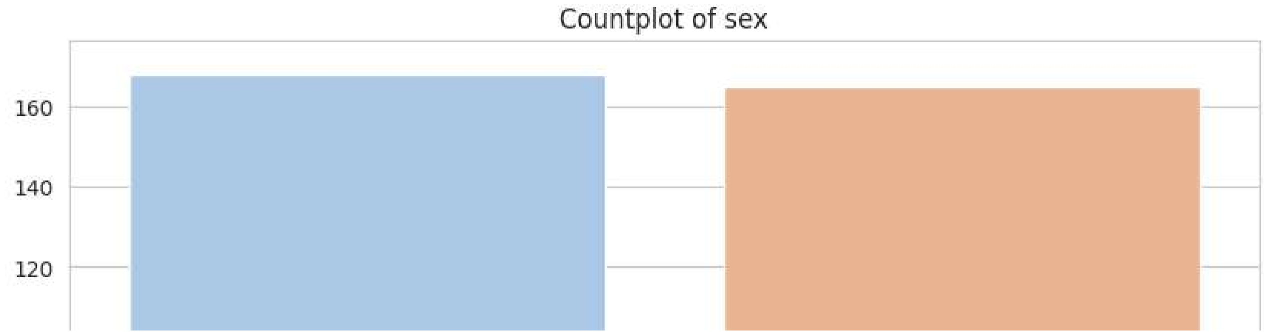
Boxplot of year

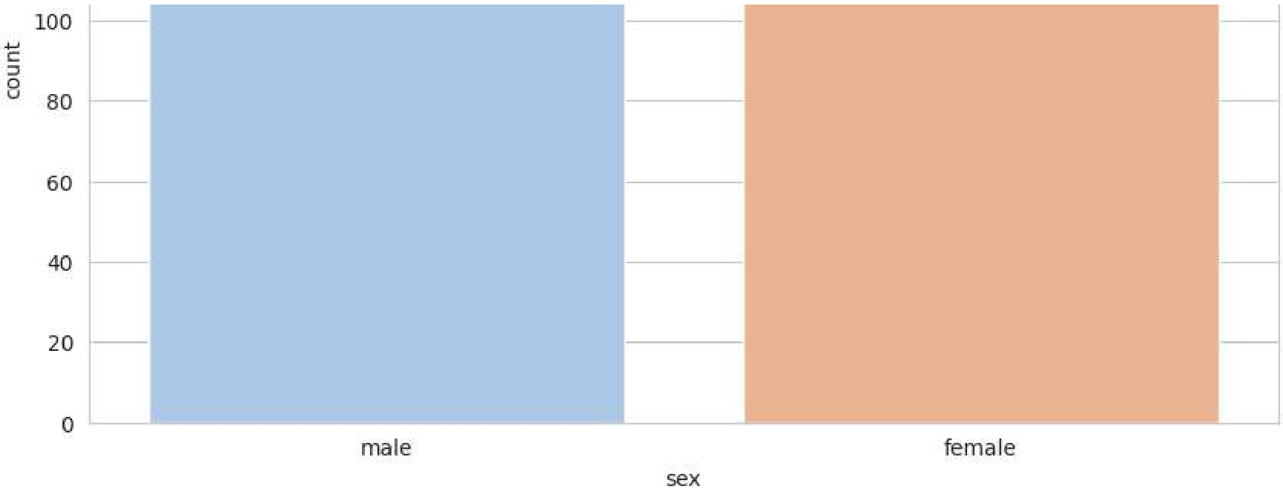



2007.00
2007.25
2007.50
2007.75
2008.00
2008.25
2008.50
2008.75
2009.00
year

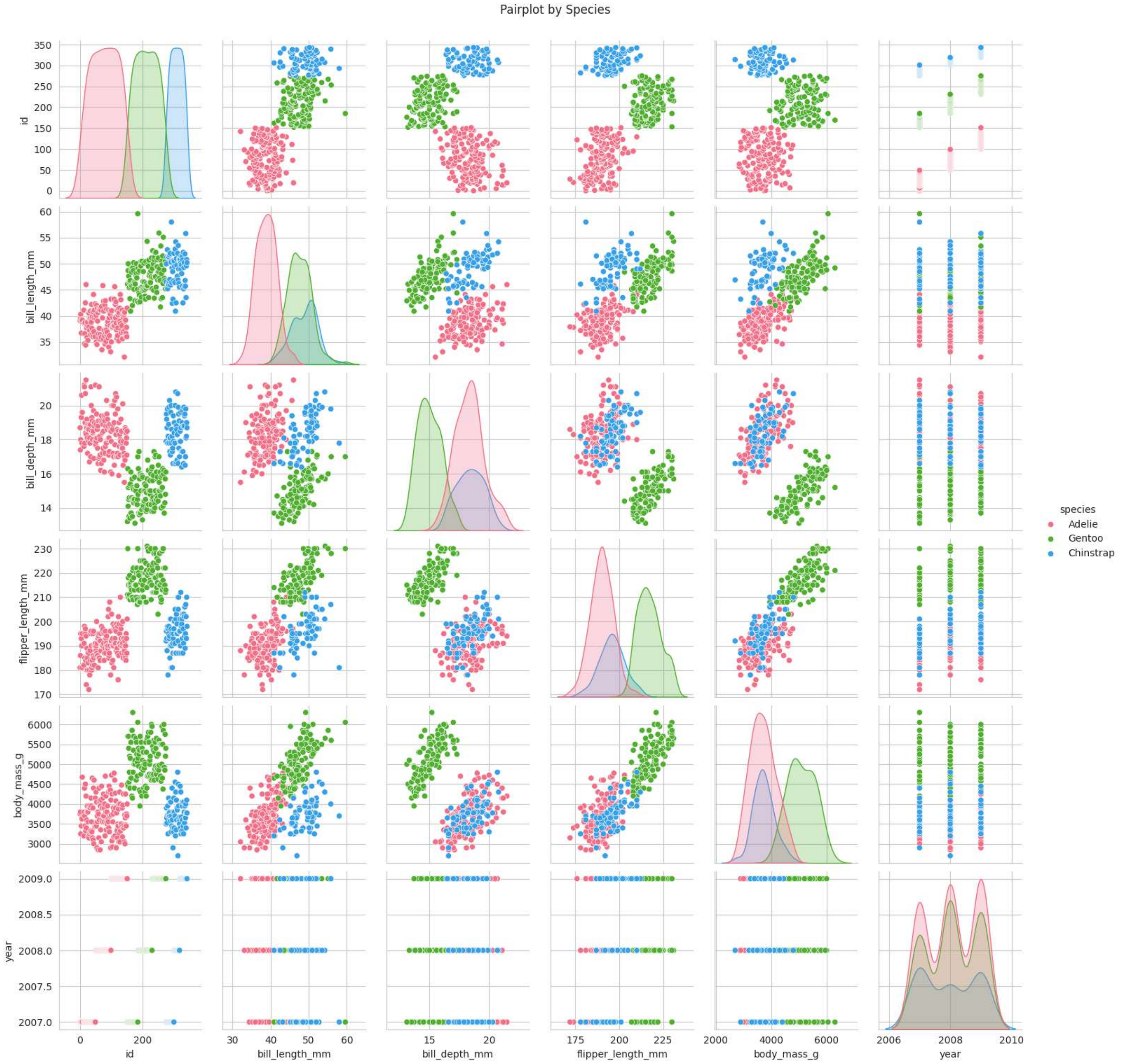

Countplot of species
count
0
20
40
60
80
100
120
140
Adelie
Gentoo
Chinstrap
species

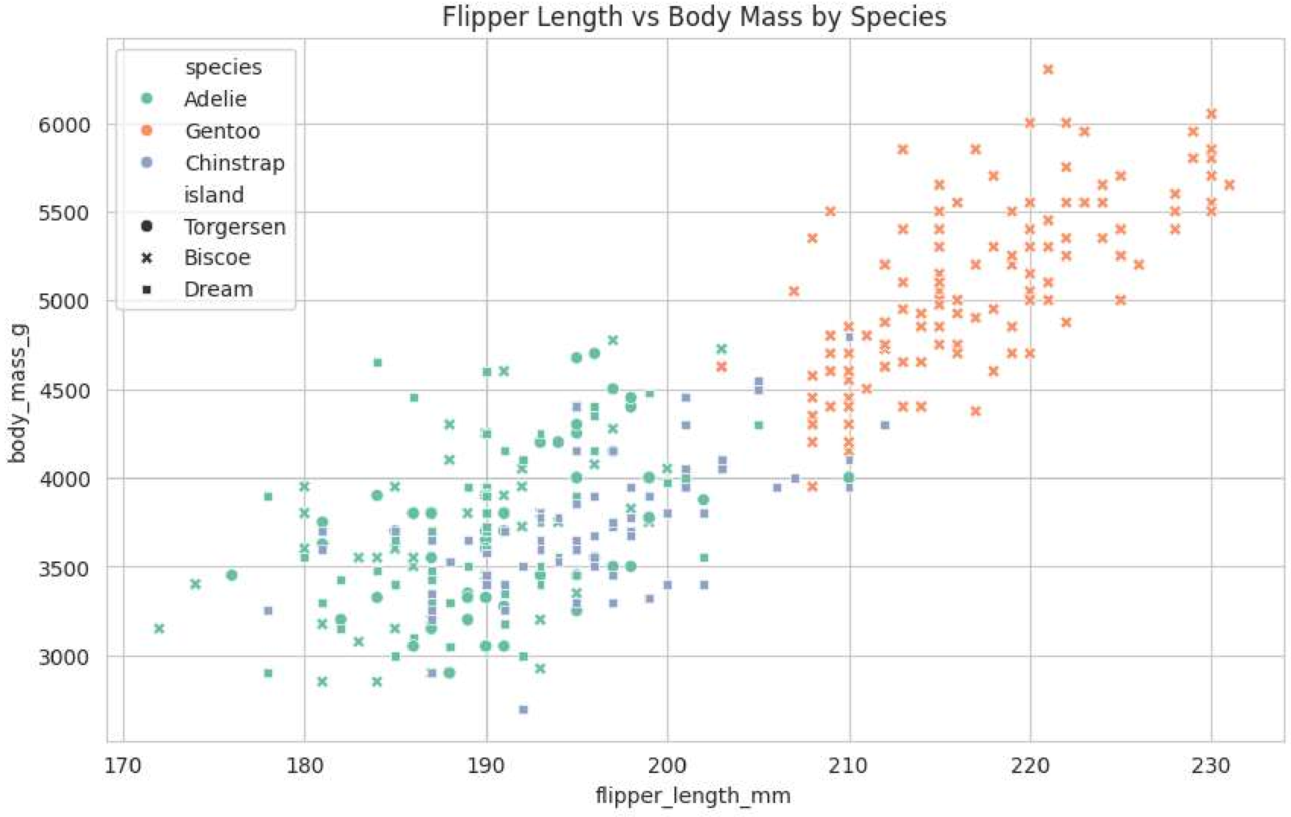

Countplot of island
count
0
20
40
60
80
100
120
140
160
Torgersen
Biscoe
Dream
island


Countplot of sex
120
140
160


100
80
60
40
20
0
count
male
female
sex

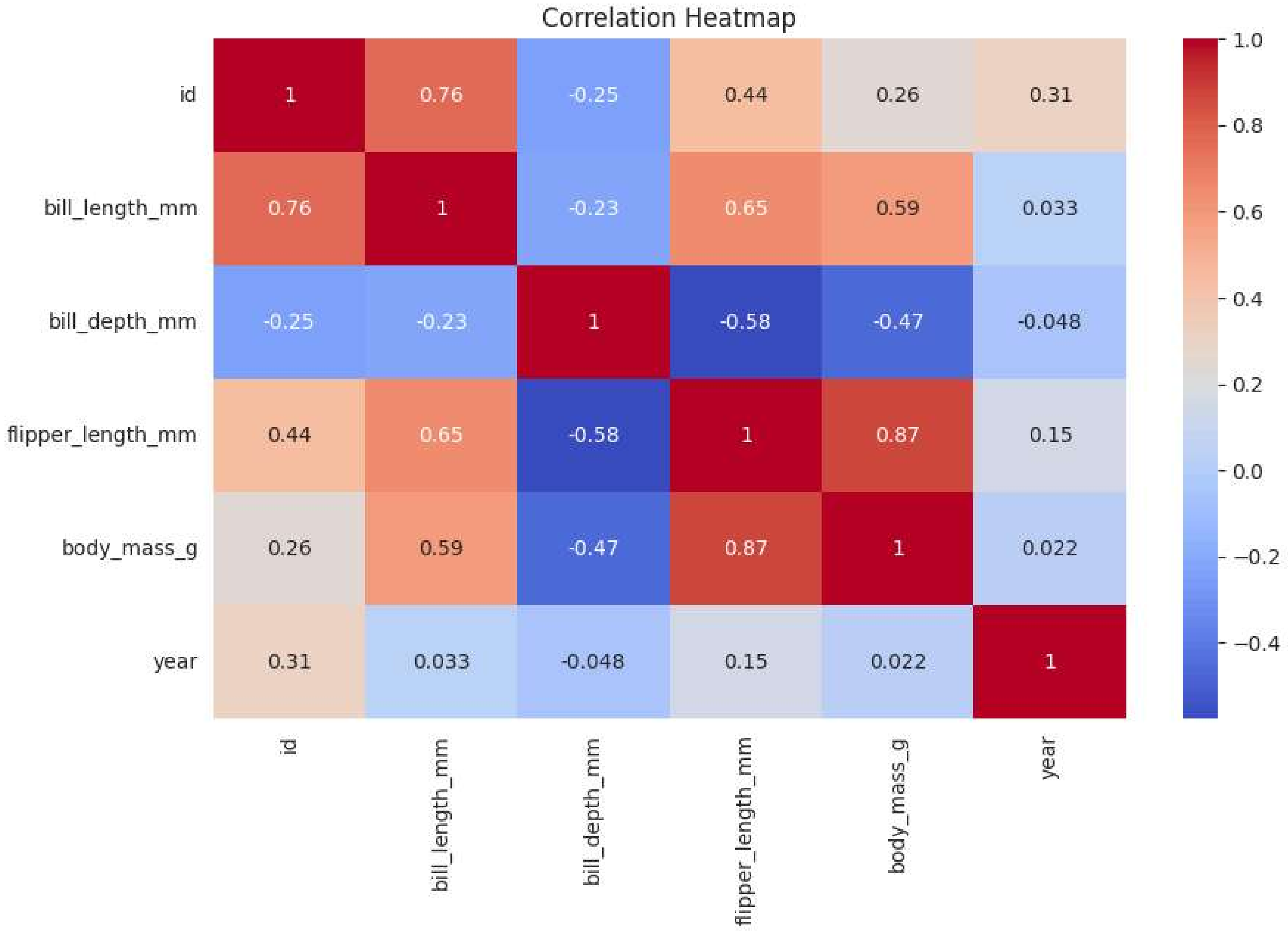
```
# Bivariate Analysis
# Pairplot
sns.pairplot(df, hue="species", palette="husl")
plt.suptitle("Pairplot by Species", y=1.02)
plt.show()

# Scatterplot: flipper length vs body mass
sns.scatterplot(
    x="flipper_length_mm",
    y="body_mass_g",
    hue="species",
    style="island",
    data=df,
    palette="Set2"
)
plt.title("Flipper Length vs Body Mass by Species")
plt.show()

# Group statistics by species
print("\n--- Mean values grouped by Species ---")
print(df.groupby("species").mean(numeric_only=True))
```


Pairplot by Species
id
350
300
250
200
150
100
50
0
bill_length_mm
60
55
50
45
40
35
bill_depth_mm
20
18
16
14
flipper_length_mm
230
220
210
200
190
180
170
body_mass_g
6000
5500
5000
4500
4000
3500
3000
year
2009.0
2008.5
2008.0
2007.5
2007.0
0
200
id
30
40
50
60
bill_length_mm
15
20
bill_depth_mm
175
200
225
flipper_length_mm
2000
4000
6000
body_mass_g
2006
2008
2010
year
species
Adelie
Gentoo
Chinstrap


Flipper Length vs Body Mass by Species
species
Adelie
Gentoo
Chinstrap
island
Torgersen
Biscoe
Dream
6000
5500
5000
4500
4000
3500
3000
body_mass_g
170
180
190
200
210
220
230
flipper_length_mm

```
--- Mean values grouped by Species ---
                   id  bill_length_mm  bill_depth_mm  flipper_length_mm  \
species
Adelie      78.000000       38.823973      18.347260         190.102740
Chinstrap  309.500000       48.833824      18.420588         195.823529
Gentoo     212.462185       47.568067      14.996639         217.235294

           body_mass_g         year
species
Adelie     3706.164384  2008.054795
Chinstrap  3733.088235  2007.970588
Gentoo     5092.436975  2008.067227
```

```
#  Correlation Analysis
corr = df.corr(numeric_only=True)
sns.heatmap(corr, annot=True, cmap="coolwarm")
plt.title("Correlation Heatmap")
plt.show()
```


Correlation Heatmap
id
bill_length_mm
bill_depth_mm
flipper_length_mm
body_mass_g
year