Name : Ch Mubashir

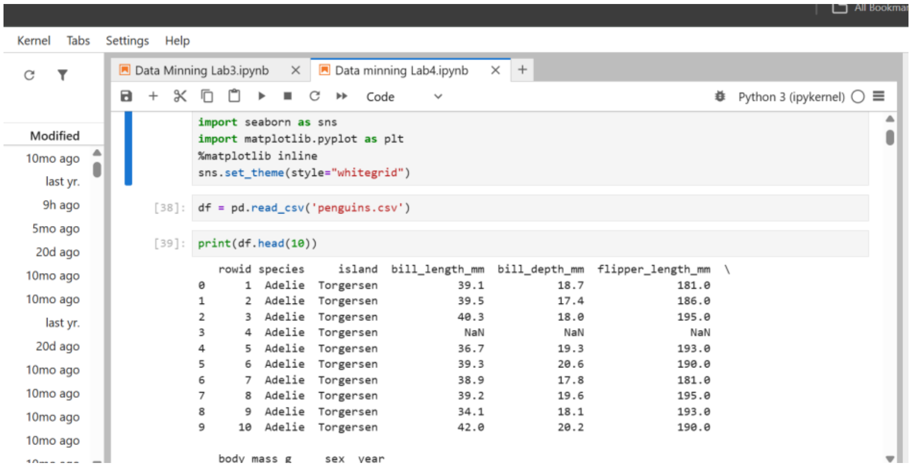
SAP : 56892.

Course : Data Minning .

Lab Task 04 :

Tasks

1. Load the dataset and show first 10 rows. Identify datatypes and count of missing values per column.

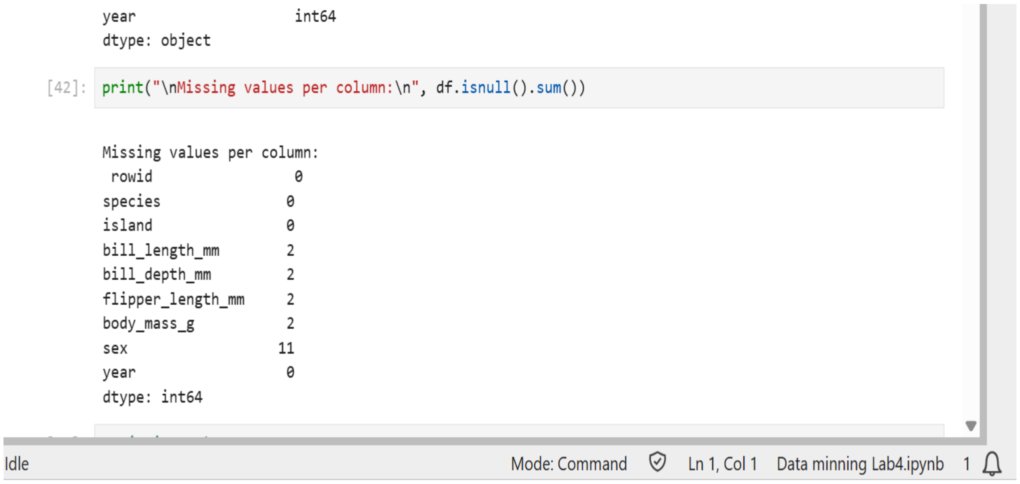


```
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
sns.set_theme(style="whitegrid")
```

```
[38]: df = pd.read_csv('penguins.csv')
```

```
[39]: print(df.head(10))
```

```
   rowid species     island  bill_length_mm  bill_depth_mm  flipper_length_mm  \
0      1  Adelie  Torgersen            39.1           18.7              181.0
1      2  Adelie  Torgersen            39.5           17.4              186.0
2      3  Adelie  Torgersen            40.3           18.0              195.0
3      4  Adelie  Torgersen             NaN            NaN                NaN
4      5  Adelie  Torgersen            36.7           19.3              193.0
5      6  Adelie  Torgersen            39.3           20.6              190.0
6      7  Adelie  Torgersen            38.9           17.8              181.0
7      8  Adelie  Torgersen            39.2           19.6              195.0
8      9  Adelie  Torgersen            34.1           18.1              193.0
9     10  Adelie  Torgersen            42.0           20.2              190.0

   body_mass_g     sex  year
```

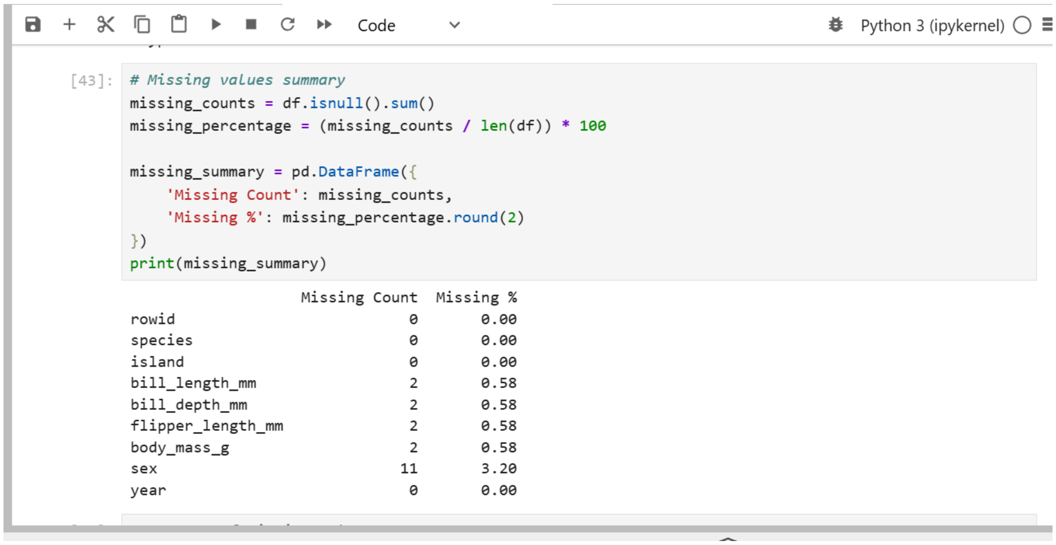


```
year                  int64
dtype: object
```

```
[42]: print("\nMissing values per column:\n", df.isnull().sum())
```

```
Missing values per column:
 rowid                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
```

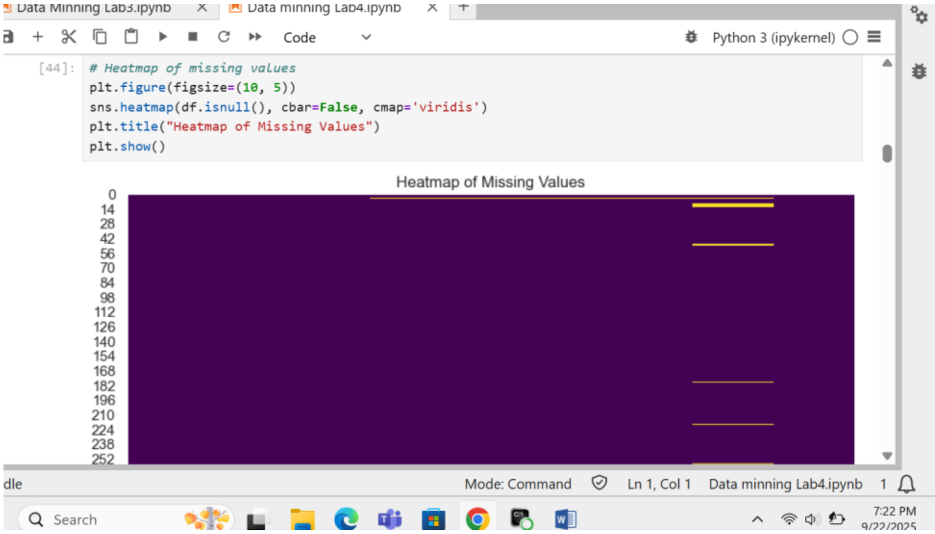
2. Visualize missingness (heatmap or bar chart) and decide which columns (if any) to drop because of excessive missingness — explain your decision in one sentence.



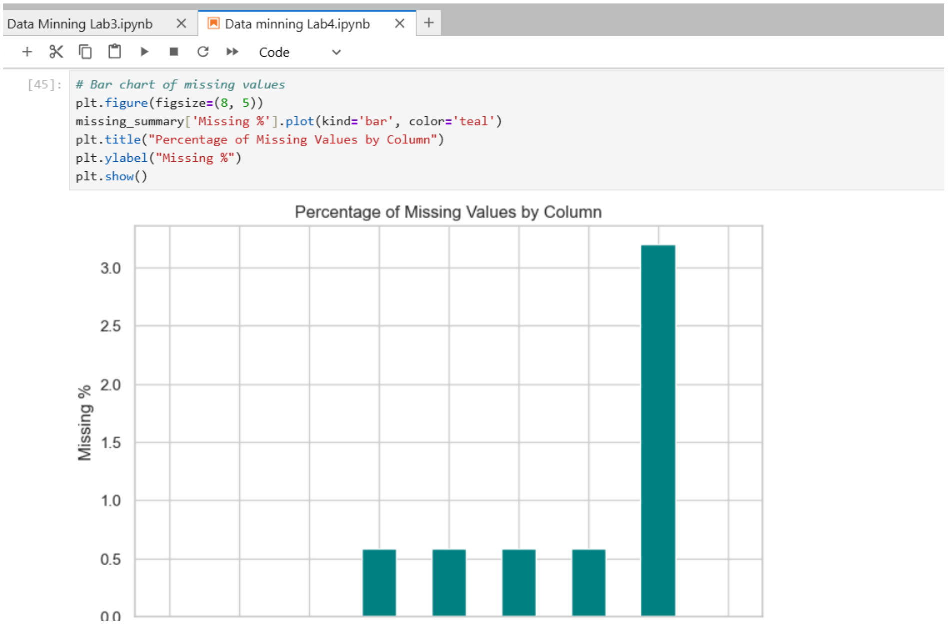
```
[43]: # Missing values summary
      missing_counts = df.isnull().sum()
      missing_percentage = (missing_counts / len(df)) * 100

      missing_summary = pd.DataFrame({
          'Missing Count': missing_counts,
          'Missing %': missing_percentage.round(2)
      })
      print(missing_summary)
```

```
                   Missing Count  Missing %
rowid                          0       0.00
species                        0       0.00
island                         0       0.00
bill_length_mm                 2       0.58
bill_depth_mm                  2       0.58
flipper_length_mm              2       0.58
body_mass_g                    2       0.58
sex                           11       3.20
year                           0       0.00
```

No column has **excessive missingness.** So, **no columns need to be dropped** for missingness.

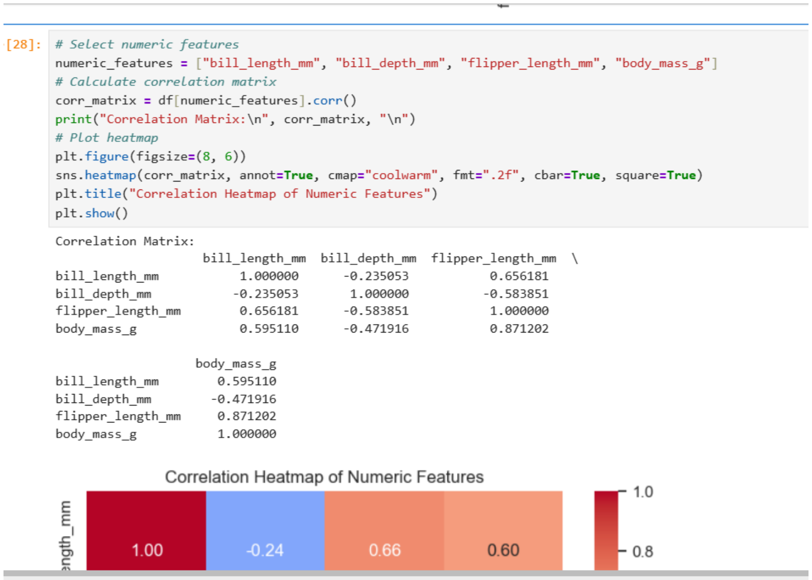


```
[44]: # Heatmap of missing values
      plt.figure(figsize=(10, 5))
      sns.heatmap(df.isnull(), cbar=False, cmap='viridis')
      plt.title("Heatmap of Missing Values")
      plt.show()
```



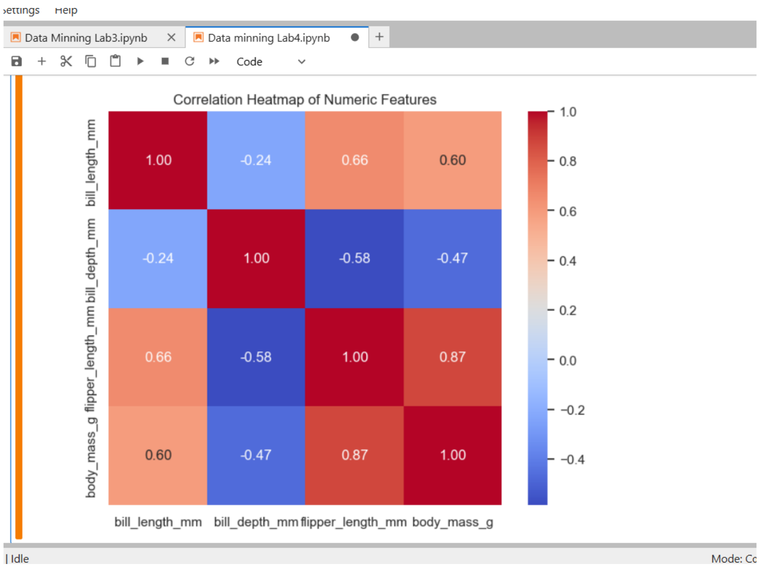
```
[45]: # Bar chart of missing values
      plt.figure(figsize=(8, 5))
      missing_summary['Missing %'].plot(kind='bar', color='teal')
      plt.title("Percentage of Missing Values by Column")
      plt.ylabel("Missing %")
      plt.show()
```

3. **Calculate and display the correlation matrix for numeric features (bill_length_mm, bill_depth_mm, flipper_length_mm, body_mass_g) and plot it as a heatmap.**



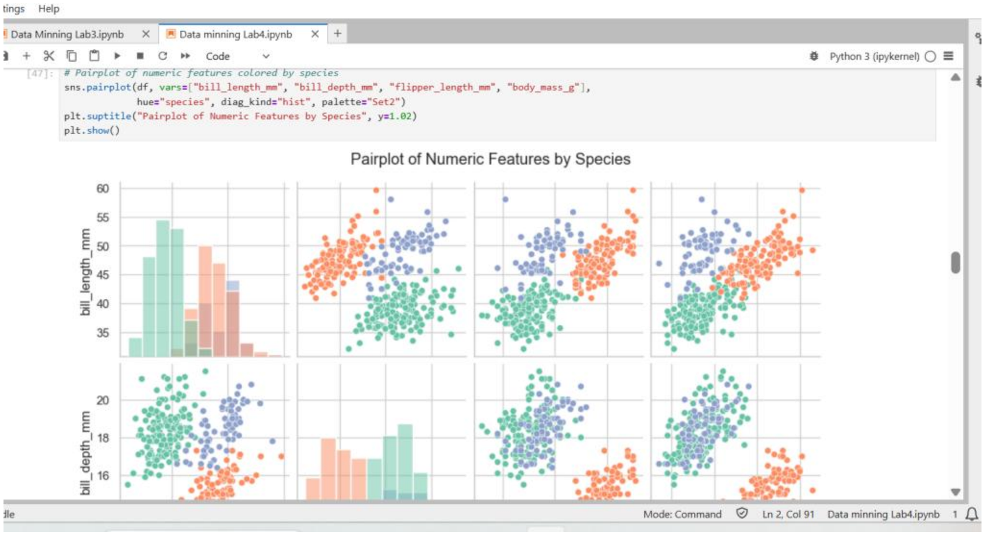
```
[28]: # Select numeric features
      numeric_features = ["bill_length_mm", "bill_depth_mm", "flipper_length_mm", "body_mass_g"]
      # Calculate correlation matrix
      corr_matrix = df[numeric_features].corr()
      print("Correlation Matrix:\n", corr_matrix, "\n")
      # Plot heatmap
      plt.figure(figsize=(8, 6))
      sns.heatmap(corr_matrix, annot=True, cmap="coolwarm", fmt=".2f", cbar=True, square=True)
      plt.title("Correlation Heatmap of Numeric Features")
      plt.show()
```

```
Correlation Matrix:
                    bill_length_mm  bill_depth_mm  flipper_length_mm  \
bill_length_mm           1.000000      -0.235053           0.656181
bill_depth_mm           -0.235053       1.000000          -0.583851
flipper_length_mm        0.656181      -0.583851           1.000000
body_mass_g              0.595110      -0.471916           0.871202

                   body_mass_g
bill_length_mm        0.595110
bill_depth_mm        -0.471916
flipper_length_mm     0.871202
body_mass_g           1.000000
```



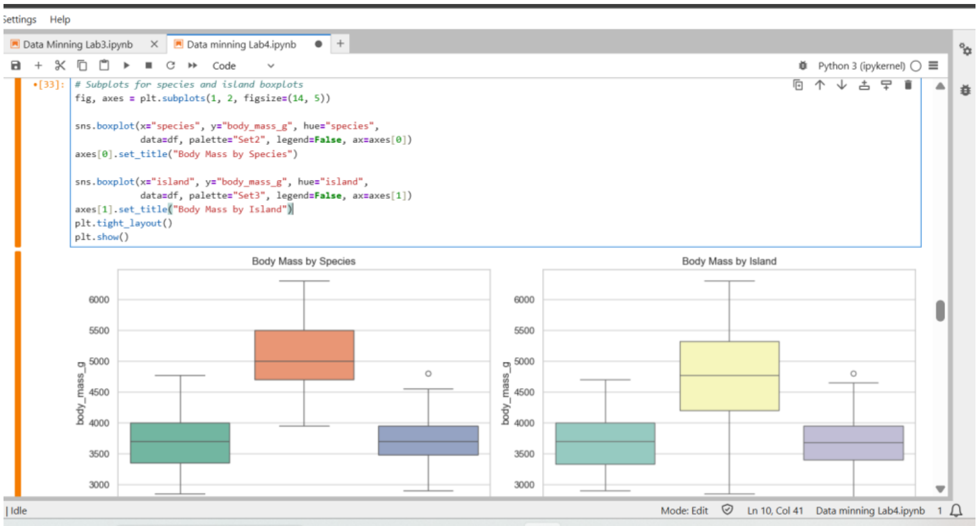
4. **Create a pairplot of the numeric features, colored by species. Comment (one line) on which feature pairs best separate species.**

The pairs **bill_length_mm vs bill_depth_mm** and **flipper_length_mm vs body_mass_g** best separate the penguin species.



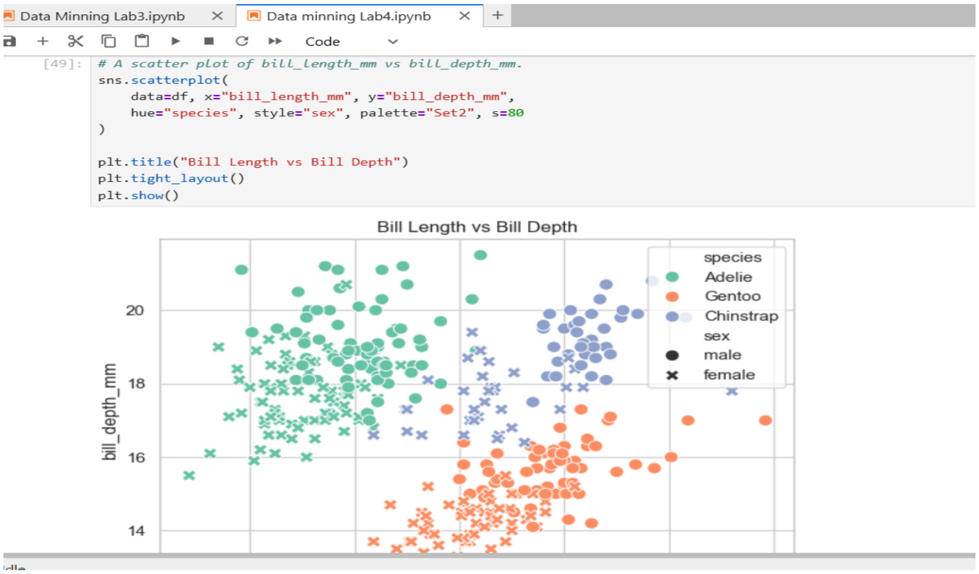
```
[47]: # Pairplot of numeric features colored by species
      sns.pairplot(df, vars=["bill_length_mm", "bill_depth_mm", "flipper_length_mm", "body_mass_g"],
                   hue="species", diag_kind="hist", palette="Set2")
      plt.suptitle("Pairplot of Numeric Features by Species", y=1.02)
      plt.show()
```

5. Draw boxplots of body_mass_g for each species and for each island (two separate plots). Note any islands with systematically heavier/lighter penguins.



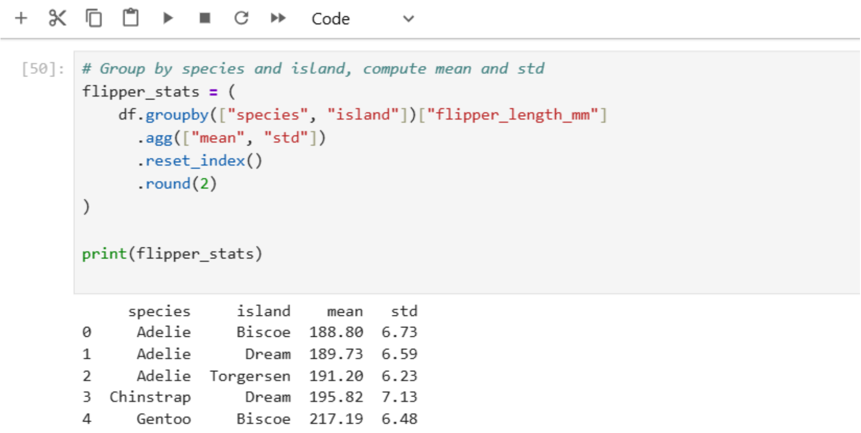
```
# Subplots for species and island boxplots
fig, axes = plt.subplots(1, 2, figsize=(14, 5))

sns.boxplot(x="species", y="body_mass_g", hue="species",
            data=df, palette="Set2", legend=False, ax=axes[0])
axes[0].set_title("Body Mass by Species")

sns.boxplot(x="island", y="body_mass_g", hue="island",
            data=df, palette="Set3", legend=False, ax=axes[1])
axes[1].set_title("Body Mass by Island")
plt.tight_layout()
plt.show()
```

6. Make a scatter plot of bill_length_mm vs bill_depth_mm colored by species; add sex as marker shape if available.



```
# A scatter plot of bill_length_mm vs bill_depth_mm.
sns.scatterplot(
    data=df, x="bill_length_mm", y="bill_depth_mm",
    hue="species", style="sex", palette="Set2", s=80
)

plt.title("Bill Length vs Bill Depth")
plt.tight_layout()
plt.show()
```

7. Using groupby, compute the mean and standard deviation of flipper_length_mm for each combination of species and island. Show results as a tidy table.

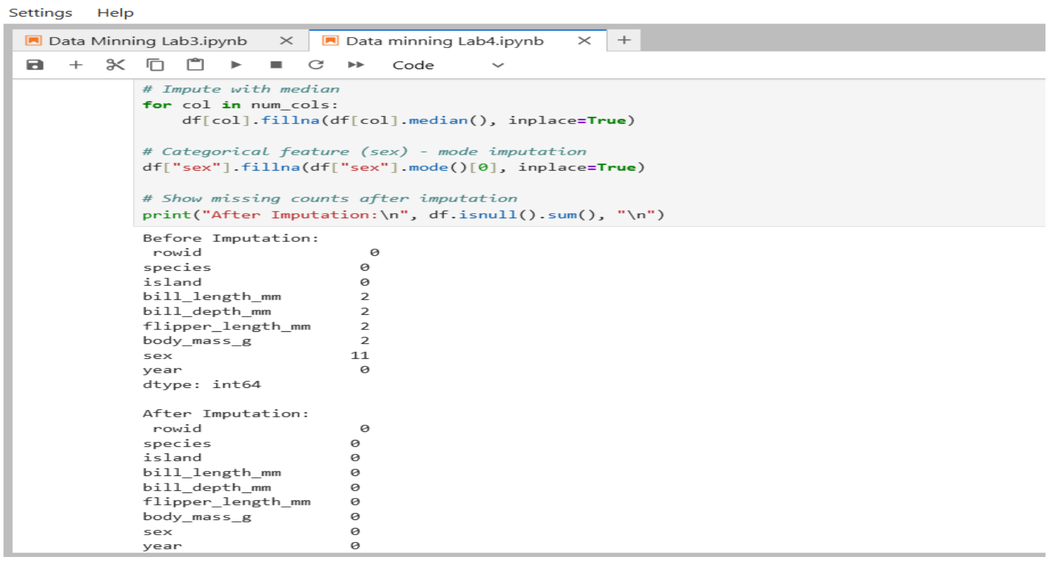


```
[50]: # Group by species and island, compute mean and std
      flipper_stats = (
          df.groupby(["species", "island"])["flipper_length_mm"]
            .agg(["mean", "std"])
            .reset_index()
            .round(2)
      )

      print(flipper_stats)
```

```
     species     island    mean   std
0     Adelie     Biscoe  188.80  6.73
1     Adelie      Dream  189.73  6.59
2     Adelie  Torgersen  191.20  6.23
3  Chinstrap      Dream  195.82  7.13
4     Gentoo     Biscoe  217.19  6.48
```

8. **Handle missing values: choose one reasonable imputation strategy for numeric columns (explain why) and apply it; then show before/after missing counts.**

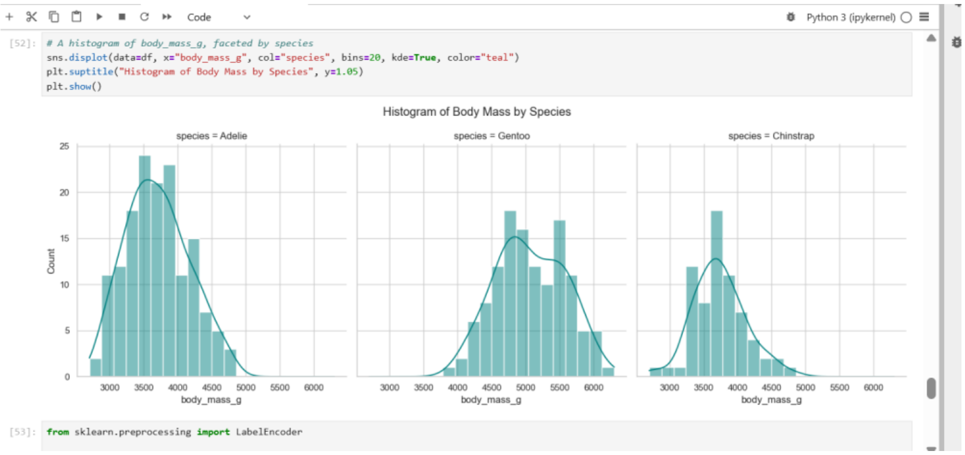
For **numeric columns**, we'll use **median imputation** (because it is robust to outliers and keeps the central tendency of the data).



```
# Impute with median
for col in num_cols:
    df[col].fillna(df[col].median(), inplace=True)

# Categorical feature (sex) - mode imputation
df["sex"].fillna(df["sex"].mode()[0], inplace=True)

# Show missing counts after imputation
print("After Imputation:\n", df.isnull().sum(), "\n")
```

```
Before Imputation:
 rowid                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

After Imputation:
 rowid              0
species            0
island             0
bill_length_mm     0
bill_depth_mm      0
flipper_length_mm  0
body_mass_g        0
sex                0
year               0
```

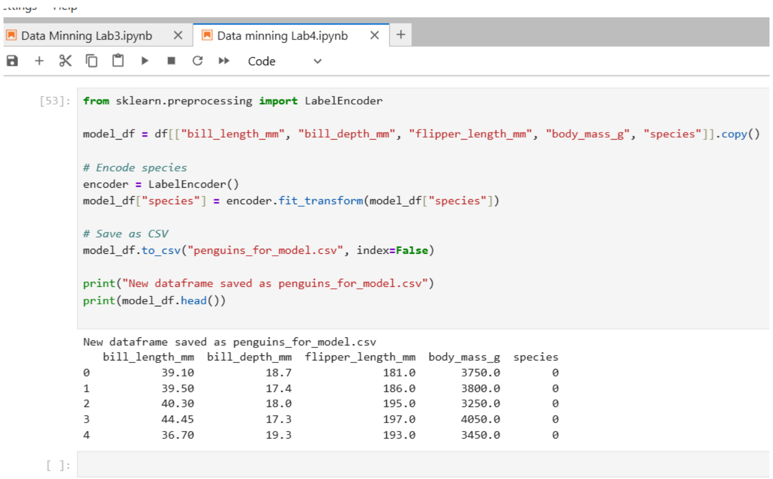
9. **Create a histogram of body_mass_g, faceted by species (use seaborn FacetGrid or displot with col=species).**



```
[52]: # A histogram of body_mass_g, faceted by species
      sns.displot(data=df, x="body_mass_g", col="species", bins=20, kde=True, color="teal")
      plt.suptitle("Histogram of Body Mass by Species", y=1.05)
      plt.show()
```

```
[53]: from sklearn.preprocessing import LabelEncoder
```

10. Short modelling preparation: create a new dataframe with only numeric features and encode species to numeric labels — save it as penguins_for_model.csv.



```
[53]: from sklearn.preprocessing import LabelEncoder

      model_df = df[["bill_length_mm", "bill_depth_mm", "flipper_length_mm", "body_mass_g", "species"]].copy()

      # Encode species
      encoder = LabelEncoder()
      model_df["species"] = encoder.fit_transform(model_df["species"])

      # Save as CSV
      model_df.to_csv("penguins_for_model.csv", index=False)

      print("New dataframe saved as penguins_for_model.csv")
      print(model_df.head())
```

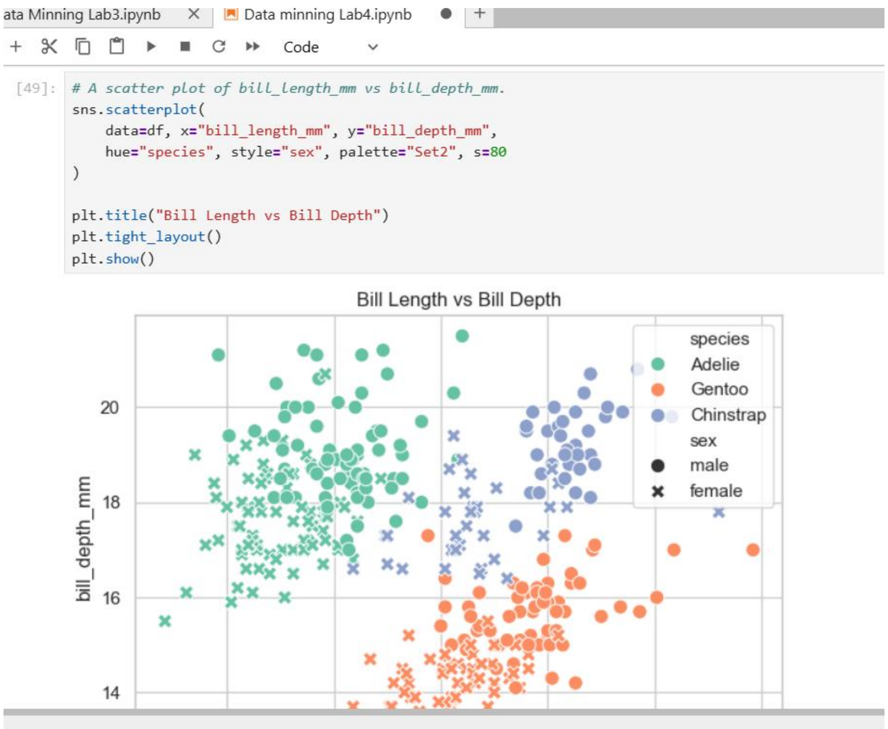
```
New dataframe saved as penguins_for_model.csv
   bill_length_mm  bill_depth_mm  flipper_length_mm  body_mass_g  species
0           39.10           18.7              181.0       3750.0        0
1           39.50           17.4              186.0       3800.0        0
2           40.30           18.0              195.0       3250.0        0
3           44.45           17.3              197.0       4050.0        0
4           36.70           19.3              193.0       3450.0        0
```

```
[ ]:
```

Exercise:

A. Which two numerical features seem to be most predictive of species? Justify with visuals/tables.

Correlation & pairplots that **bill_length_mm** and **bill_depth_mm** are the strongest at separating species.

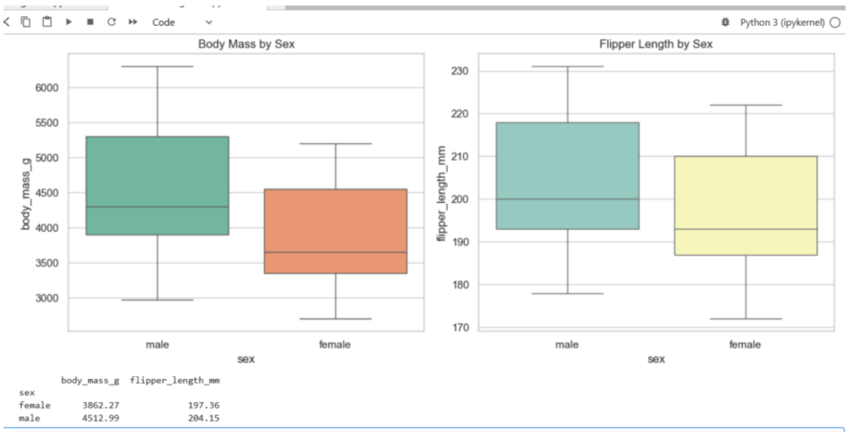


```
# A scatter plot of bill_length_mm vs bill_depth_mm.
sns.scatterplot(
    data=df, x="bill_length_mm", y="bill_depth_mm",
    hue="species", style="sex", palette="Set2", s=80
)

plt.title("Bill Length vs Bill Depth")
plt.tight_layout()
plt.show()
```

B. Is there significant sexual dimorphism (difference between sexes) in body_mass_g or flipper_length_mm? Show supporting plot(s).

Males generally have **higher body mass** and **longer flippers** than females.

Boxplots and group means confirm sexual dimorphism.



```
        body_mass_g  flipper_length_mm
sex
female      3862.27             197.36
male        4512.99             204.15
```

Conclusion :

1. **Bill length and bill depth** are the two most important features that separate species; species form clear, non-overlapping clusters in this space.
2. **Flipper length and body mass** are also informative but overlap more between species.
3. **Sexual dimorphism** is evident: males are heavier and have longer flippers than females across species.
4. **Islands** indirectly affect size because some species (e.g., Gentoo on Biscoe) are systematically heavier.
5. Overall, **species identity** and **sex** are the two most important biological factors influencing penguin morphology.