

+ Code + Text

Connect ▾

```
[ ] from google.colab import drive
    drive.mount('/content/drive')
```

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).

```
[ ] import pandas as pd
    import matplotlib.pyplot as plt
    import seaborn as sns
    path="/content/drive/MyDrive/penguins/penguins.csv"
    df=pd.read_csv(path)
    df.head()
    print('Dataset has', df.shape[0] , 'rows and', df.shape[1], 'columns')
    df.info()
    df.describe()
    df.isnull().sum()
```

Dataset has 344 rows and 7 columns
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 344 entries, 0 to 343
Data columns (total 7 columns):

#	Column	Non-Null Count	Dtype
0	species	344 non-null	object
1	island	344 non-null	object
2	bill_length_mm	344 non-null	float64

+ Code + Text

Connect ▼

```
[ ] 1 island          344 non-null    object
    2 bill_length_mm 342 non-null    float64
    3 bill_depth_mm   342 non-null    float64
    4 flipper_length_mm 342 non-null    float64
    5 body_mass_g      342 non-null    float64
    6 sex              333 non-null    object
```

```
dtypes: float64(4), object(3)
```

```
memory usage: 18.9+ KB
```

```
species      0
```

island 0

bill_length_mm	2
----------------	---

bill_depth_mm	2
---------------	---

```
flipper_length_mm    2
```

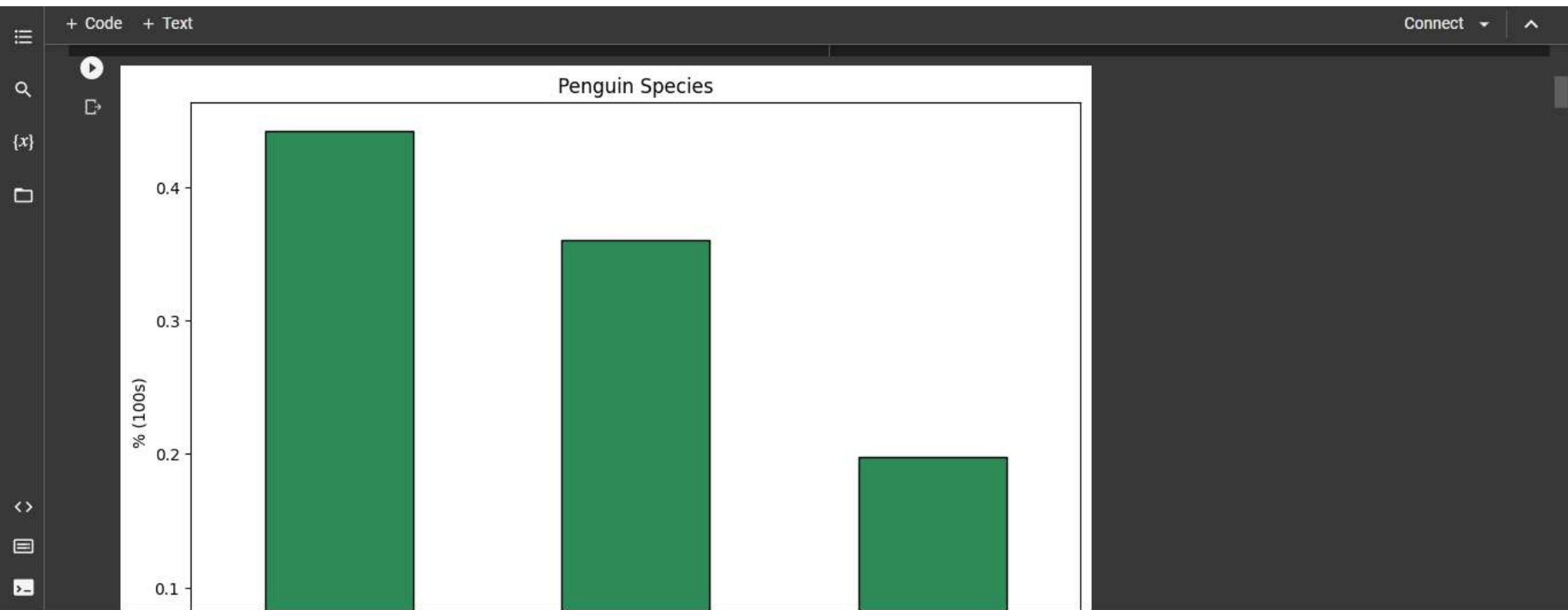
Variable	Unit
body_mass_g	g

```
sex      11
```

```
dtype: int64
```

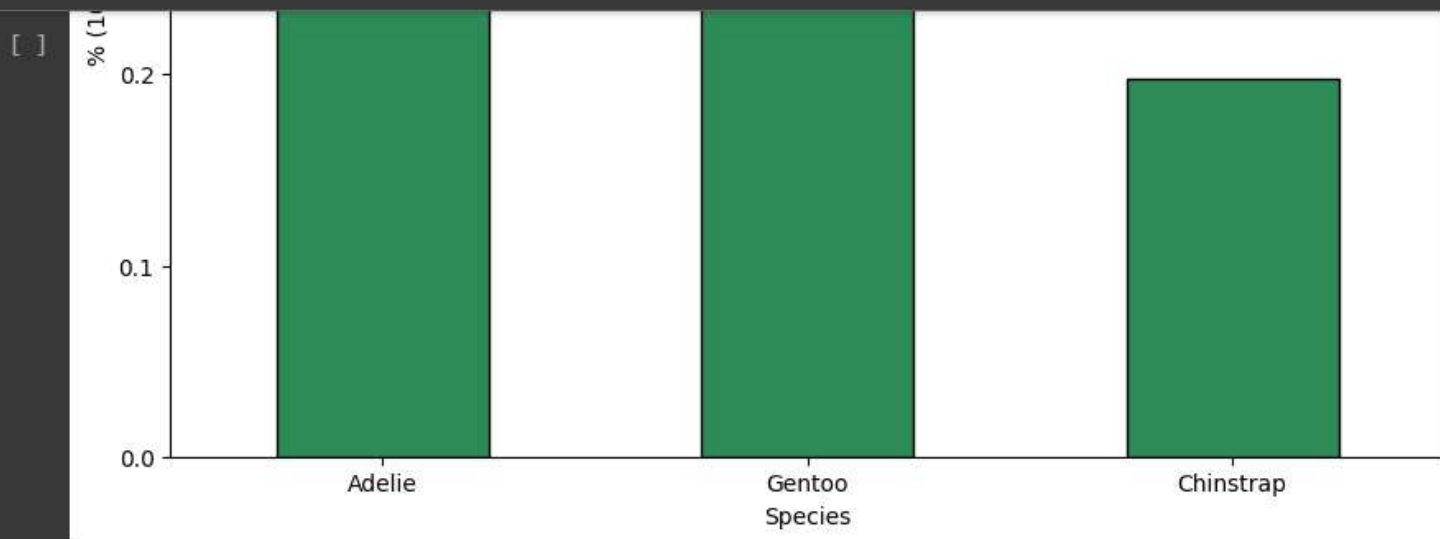
```
plt.rcParams['figure.figsize'] = (10,7)
df['species'].value_counts(normalize = True).plot(kind = 'bar', color = 'seagreen', linewidth = 1, edgecolor = 'k')
plt.title('Penguin Species')
plt.xlabel('Species')
plt.ylabel('% (100s)')
plt.xticks(rotation = 360)
plt.show()
```

Penguin Species

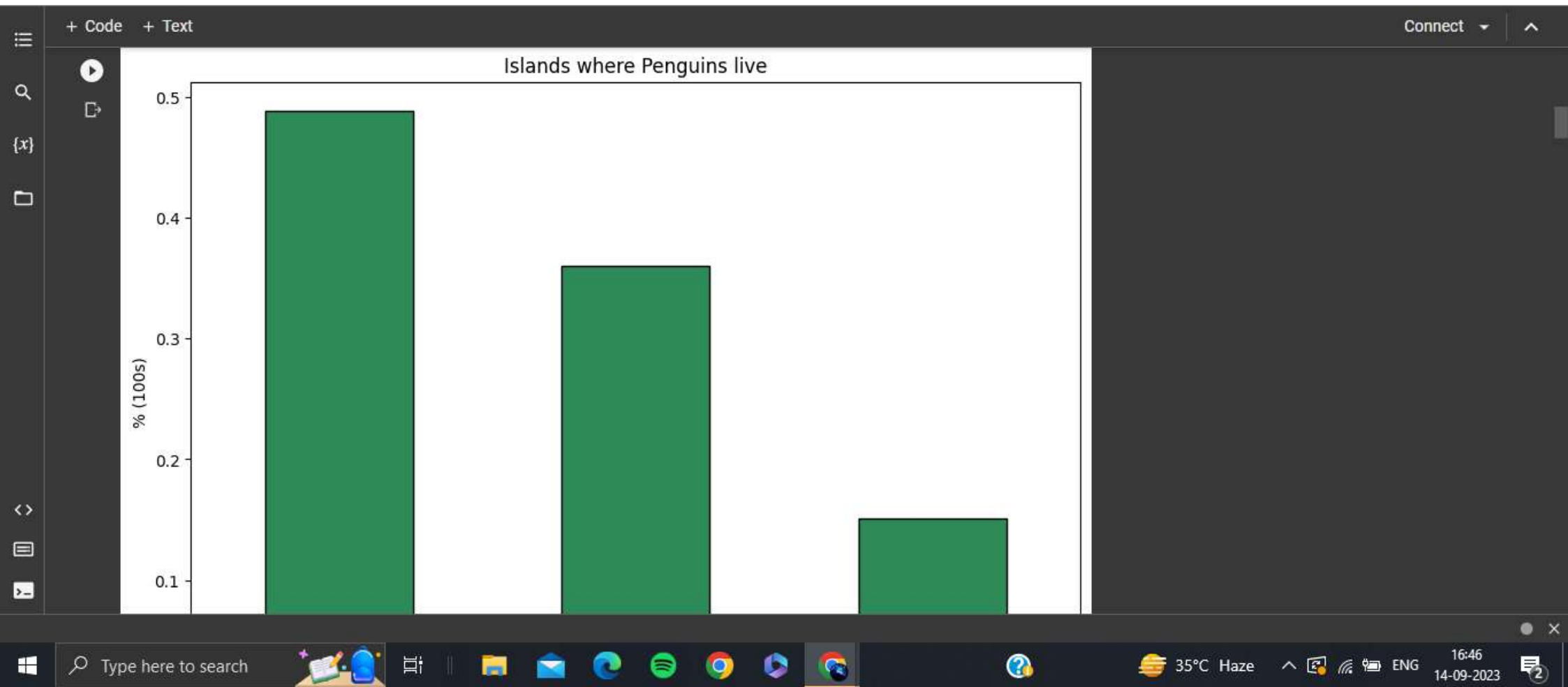


+ Code + Text

Connect



```
df['island'].value_counts(normalize = True).plot(kind = 'bar', color = 'seagreen', linewidth = 1, edgecolor = 'k')  
plt.title('Islands where Penguins live')  
plt.xlabel('Island')  
plt.ylabel('% (100s)')  
plt.xticks(rotation = 360)  
plt.show()
```

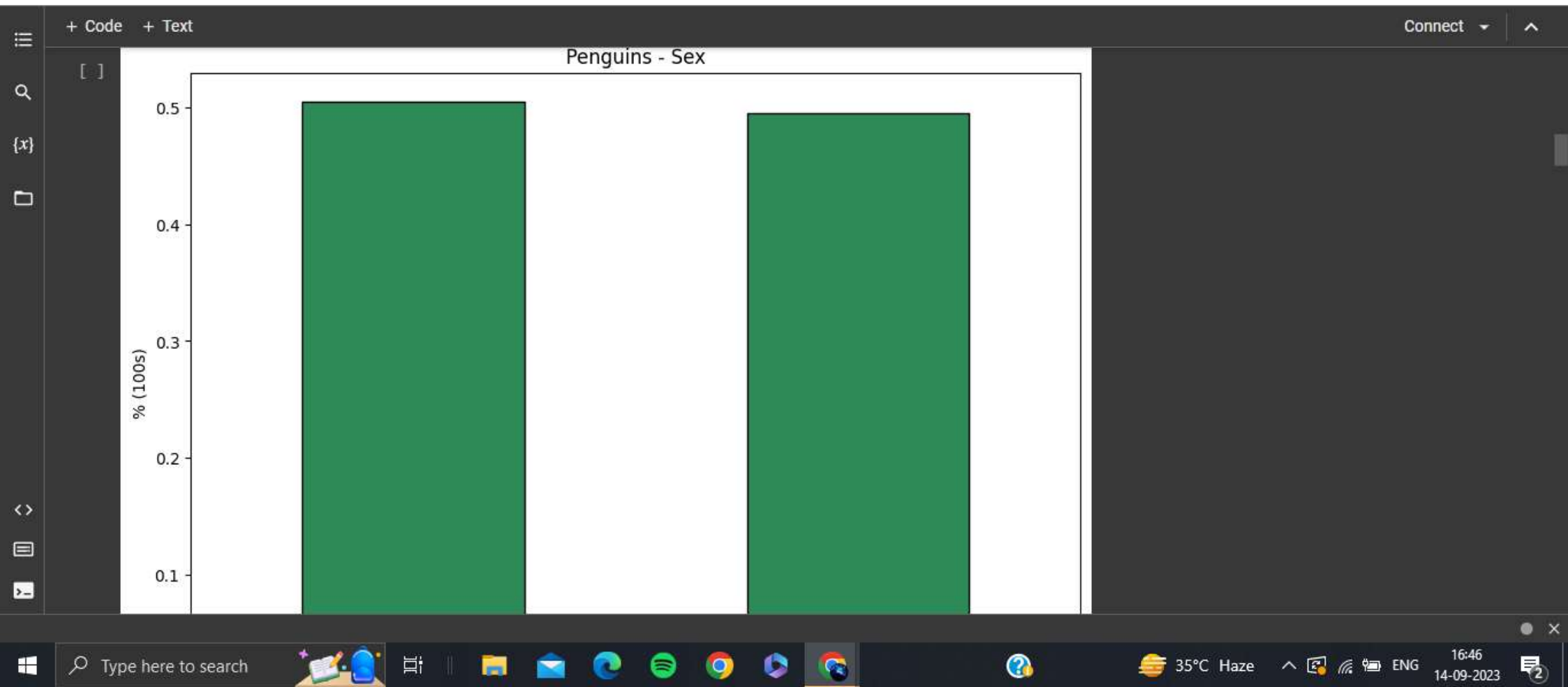


+ Code + Text

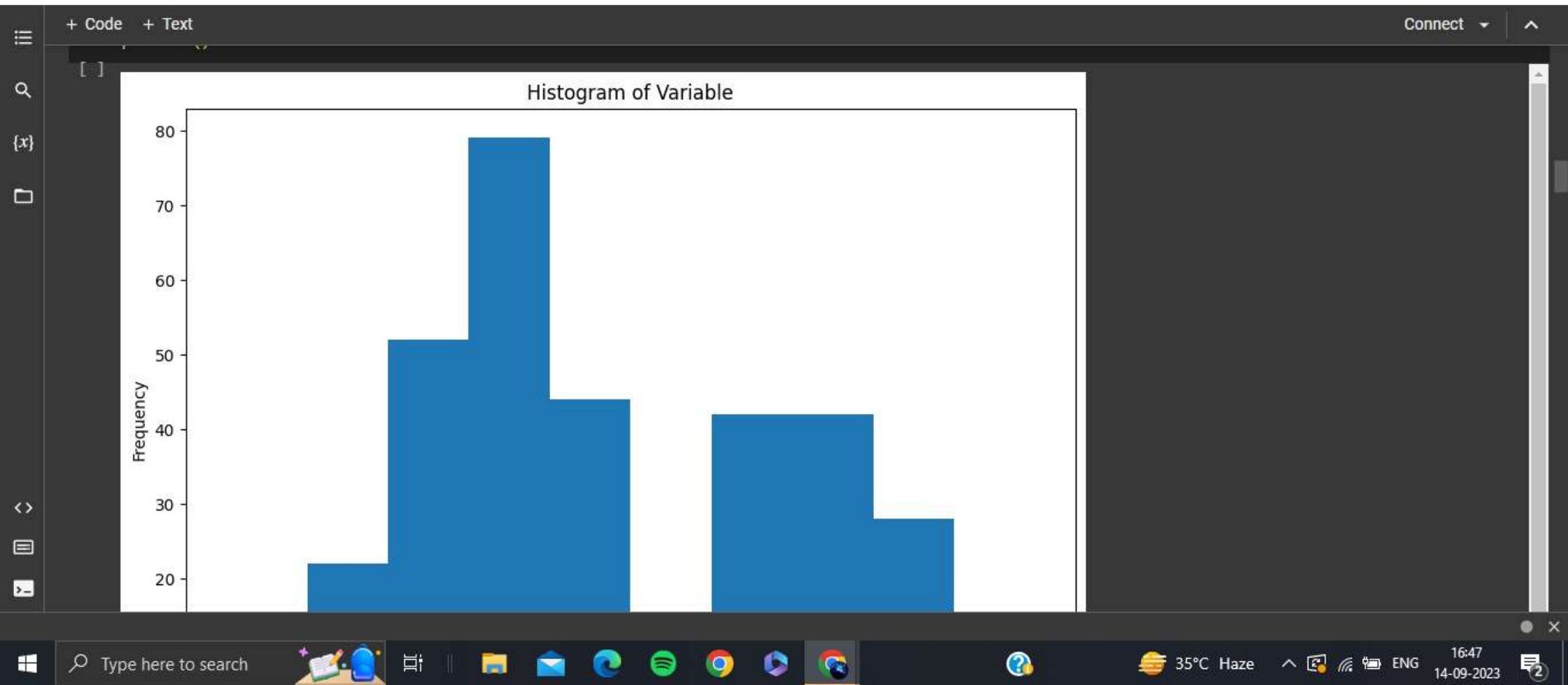
Connect

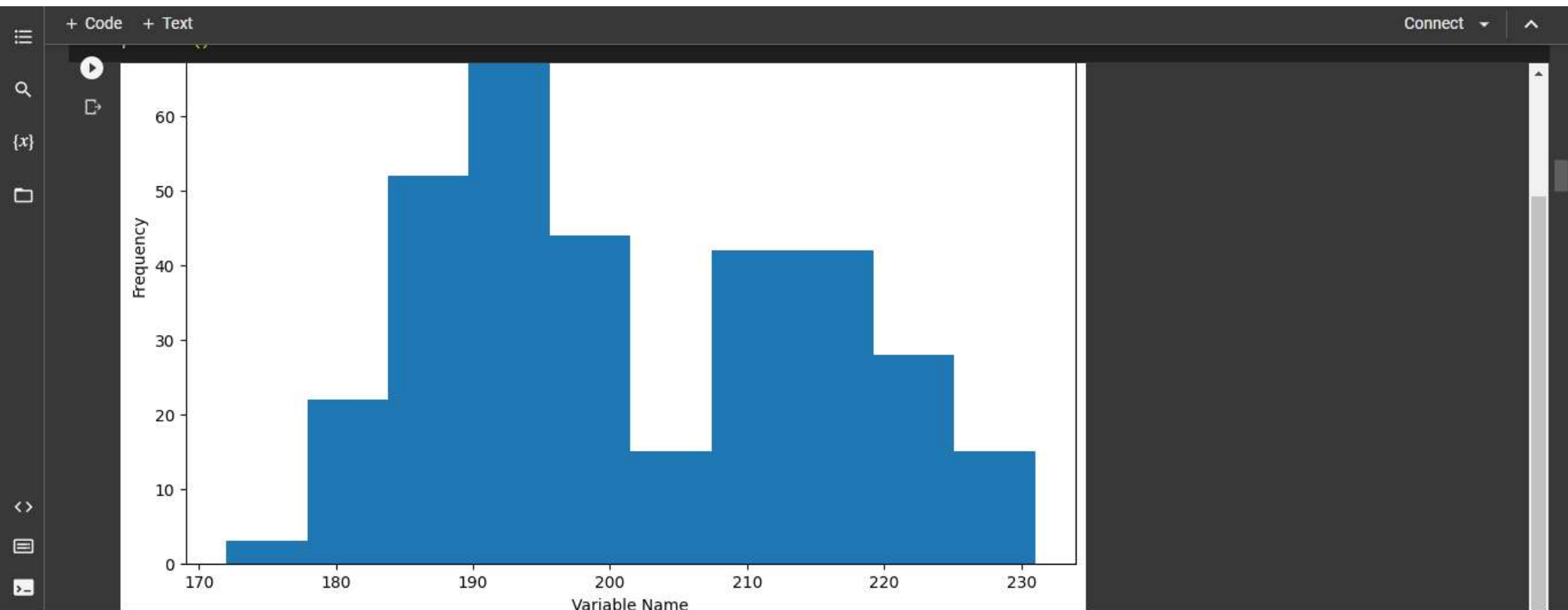


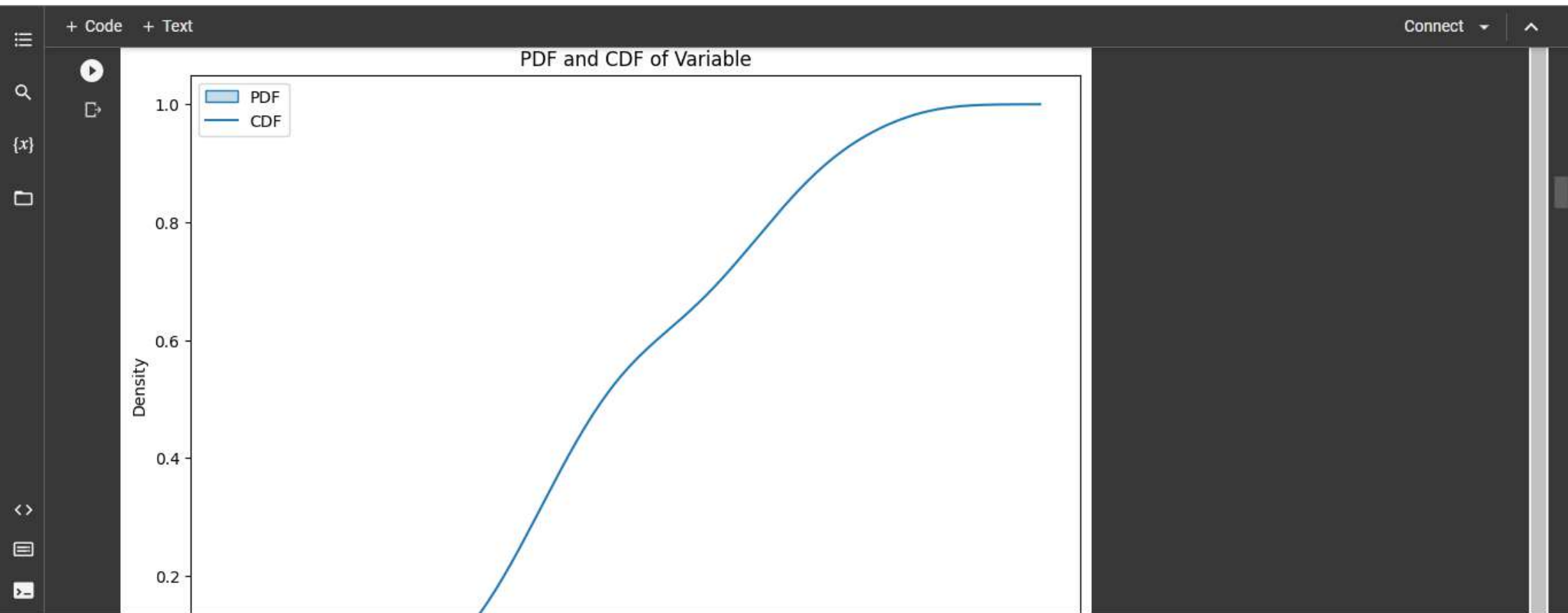
```
df['sex'].value_counts(normalize = True).plot(kind = 'bar', color = 'seagreen', linewidth = 1, edgecolor = 'k')  
plt.title('Penguins - Sex')  
plt.xlabel('Sex')  
plt.ylabel('% (100s)')  
plt.xticks(rotation = 360)  
plt.show()
```







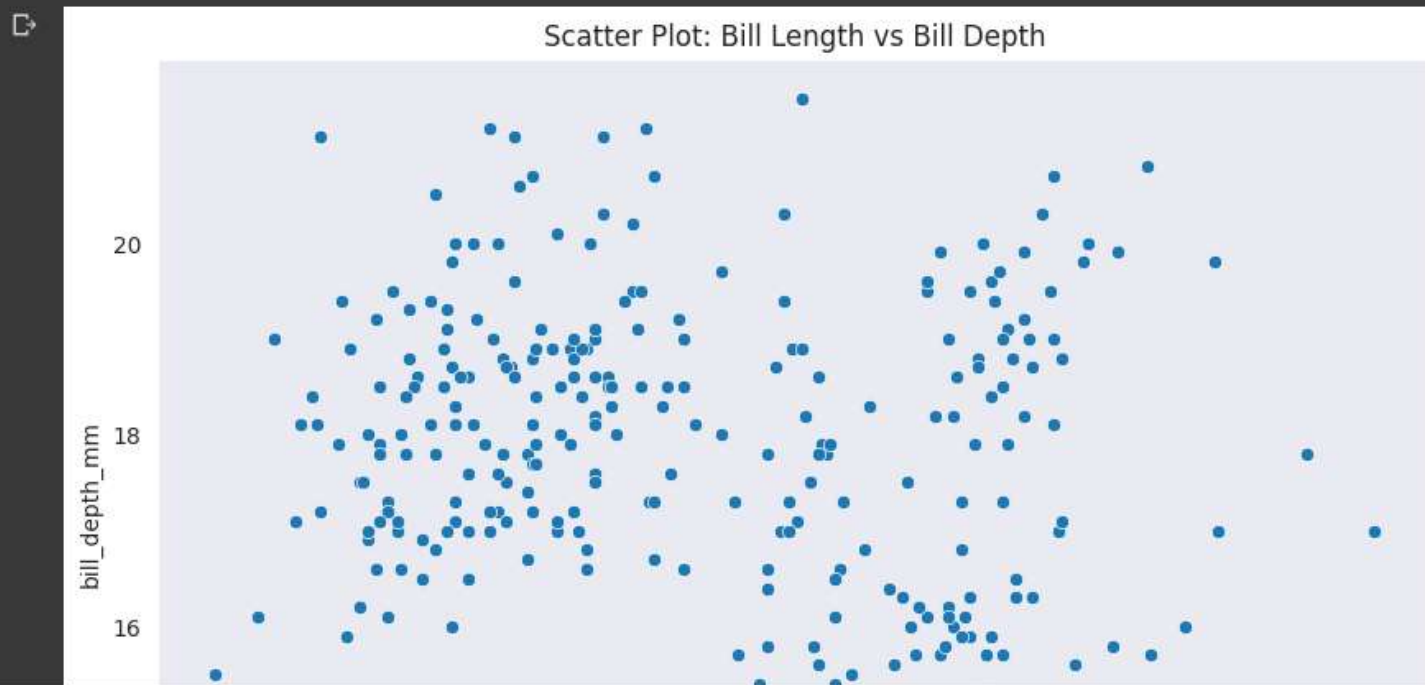


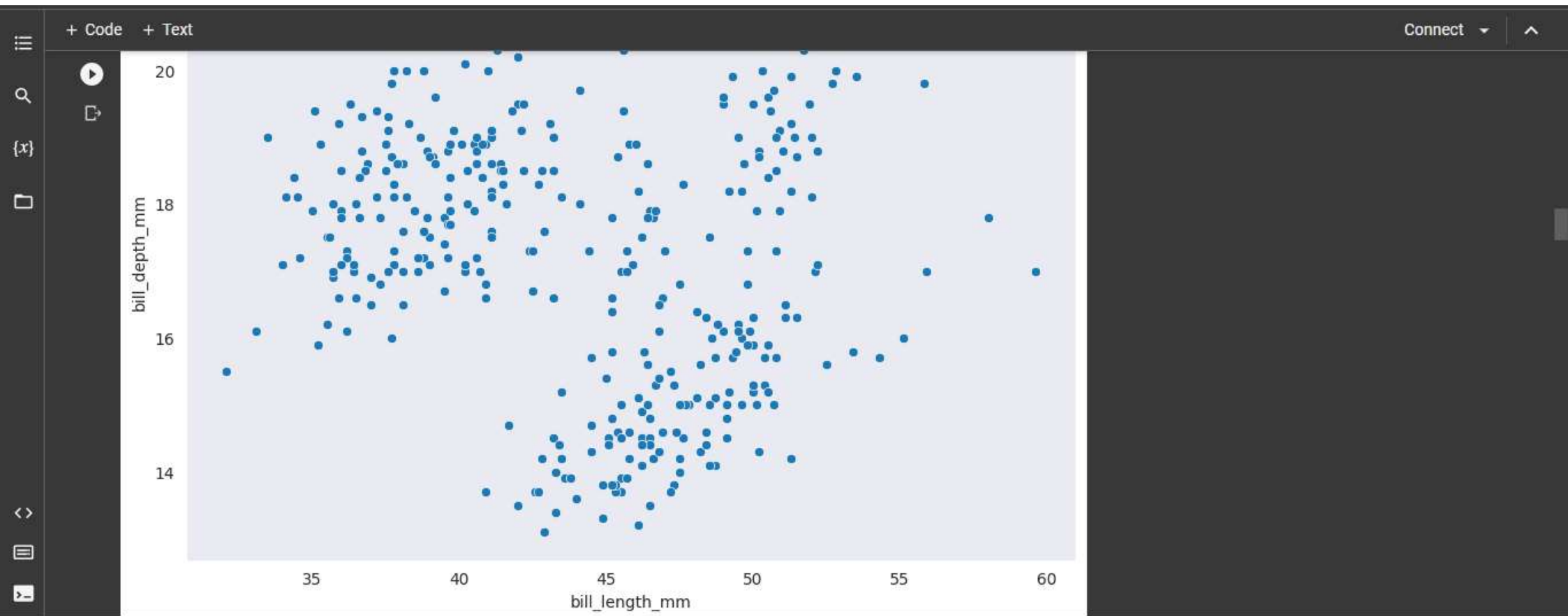


+ Code + Text

Connect

```
sns.scatterplot(data=df, x="bill_length_mm", y="bill_depth_mm")  
plt.title("Scatter Plot: Bill Length vs Bill Depth")  
plt.show()
```





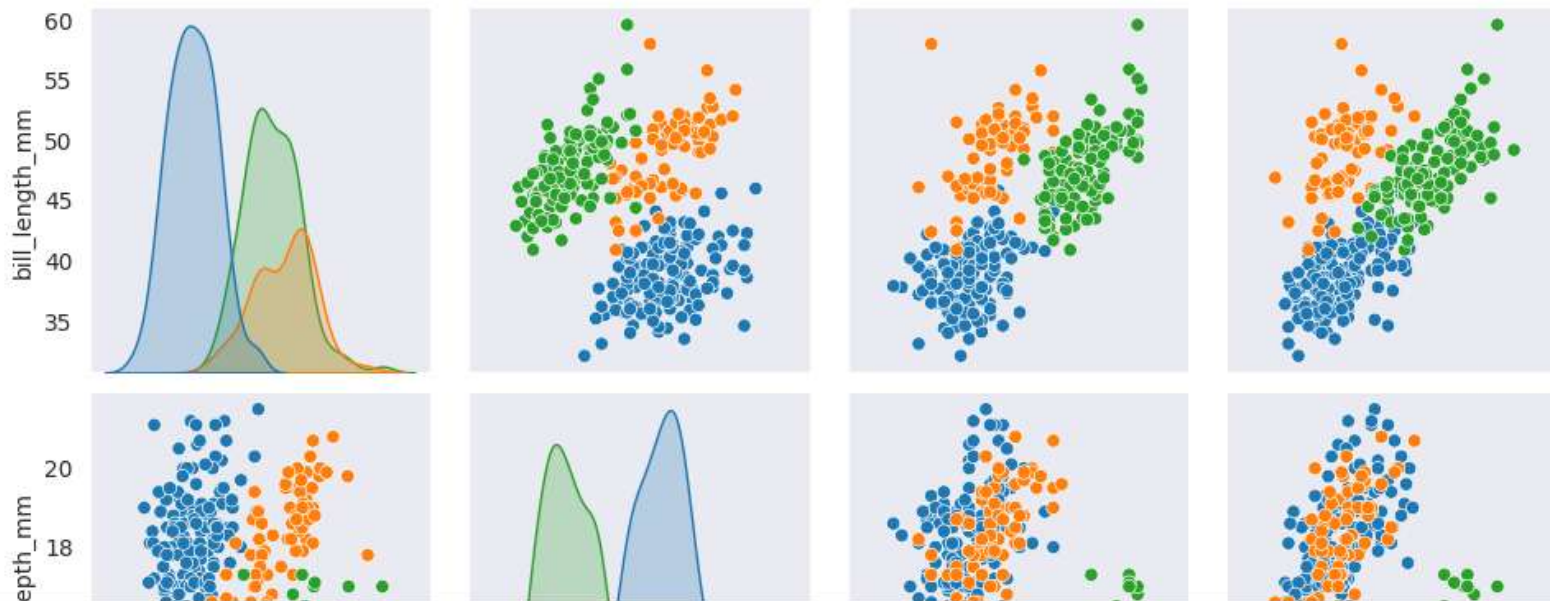
+ Code + Text

Connect

```
sns.pairplot(data=df, hue="species")  
plt.suptitle("Pairplot of Penguin Data", y=1.02)  
plt.show()
```

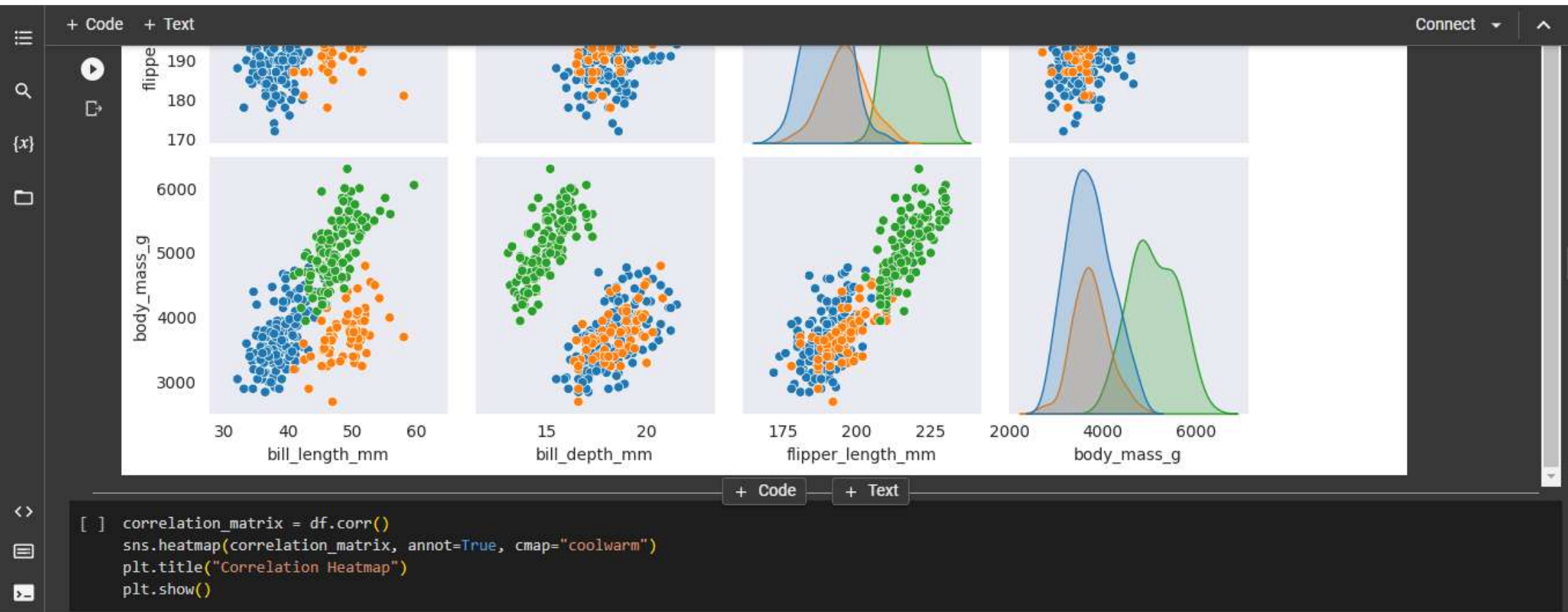


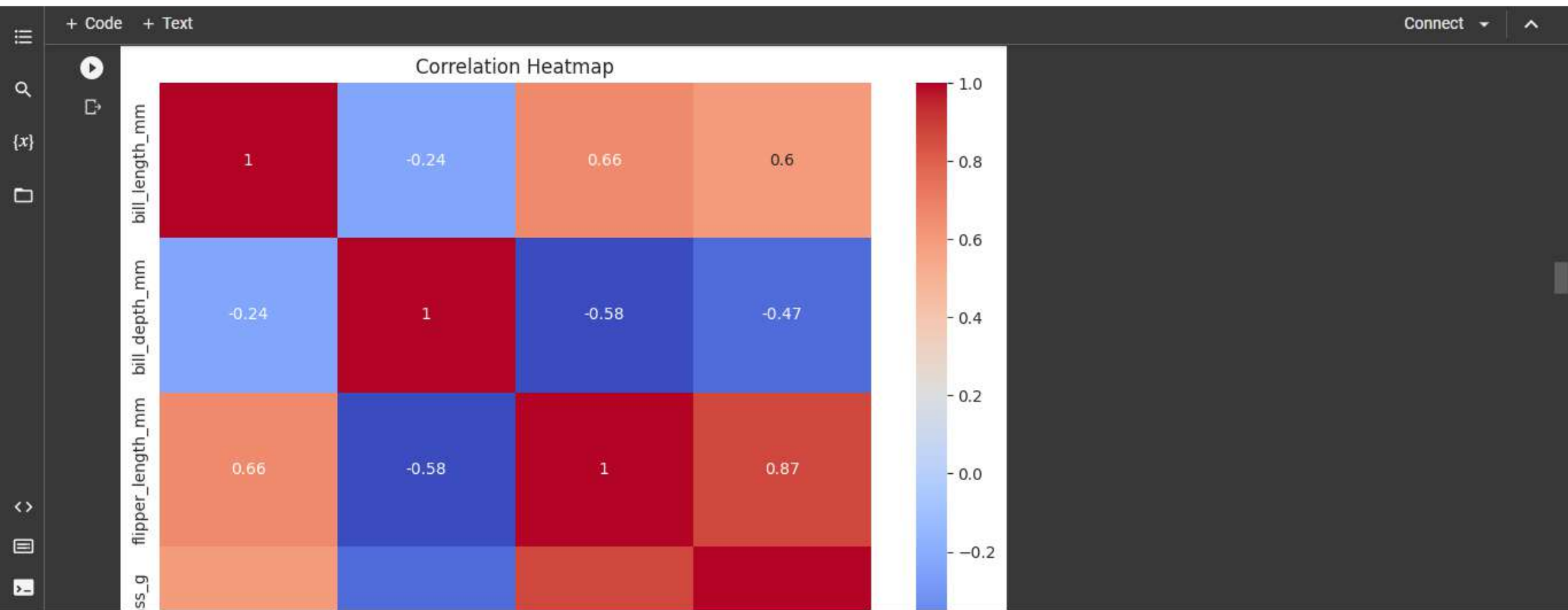
Pairplot of Penguin Data





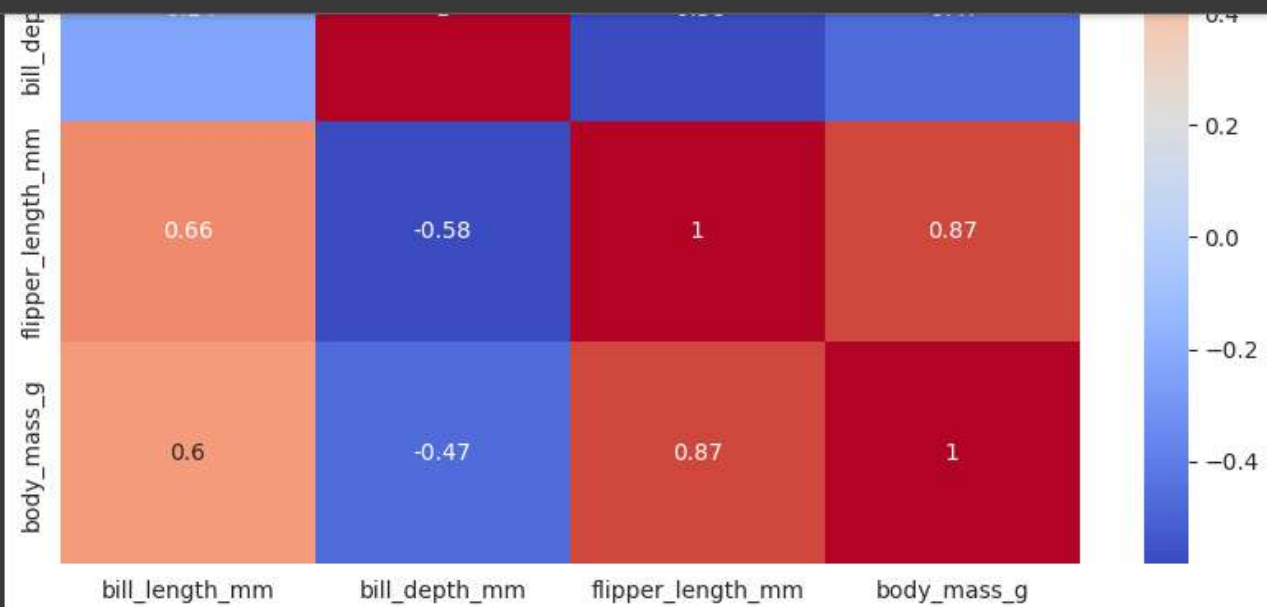




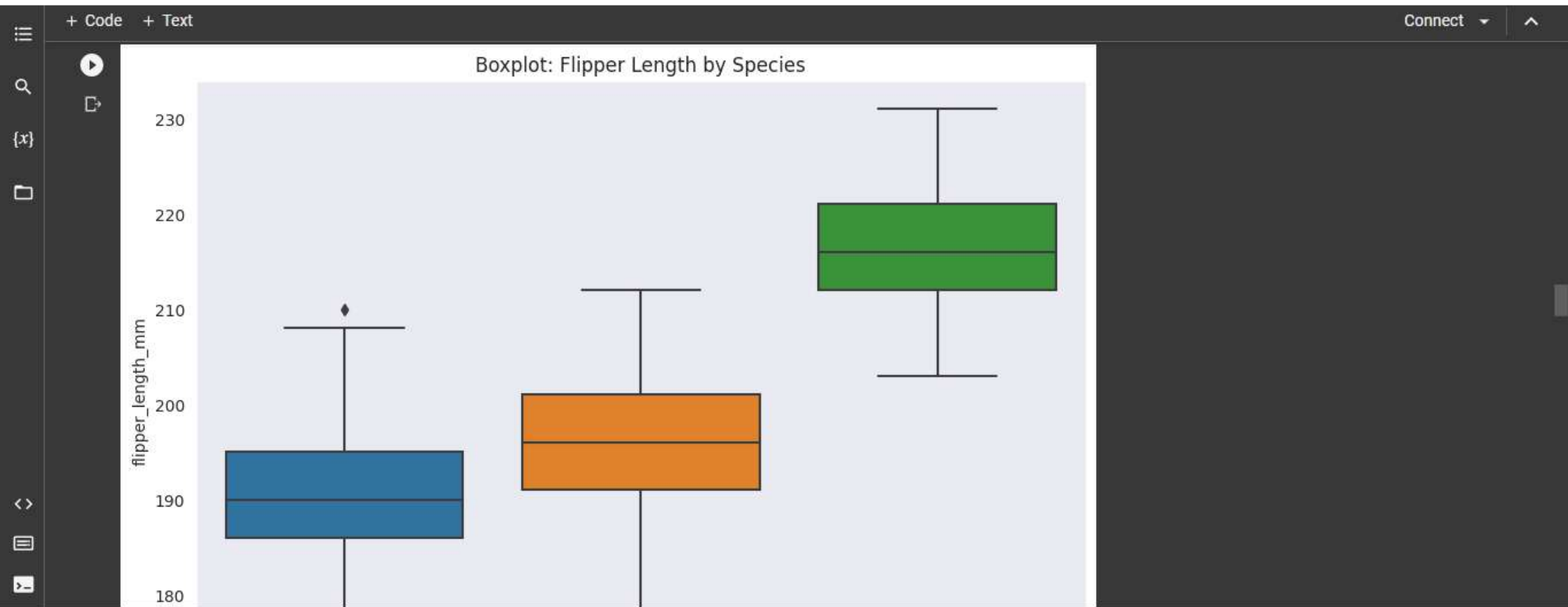


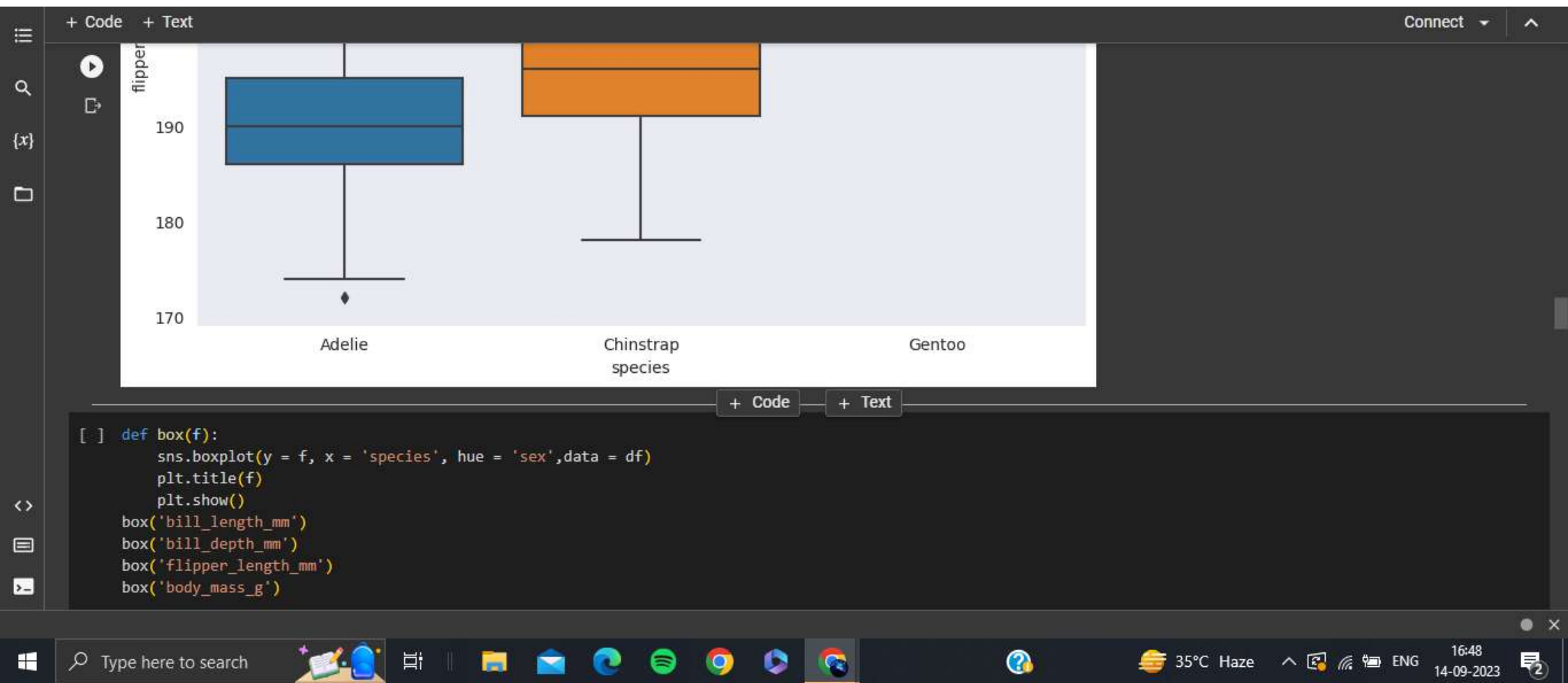
+ Code + Text

Connect



```
[ ] sns.boxplot(data=df, x="species", y="flipper_length_mm")  
plt.title("Boxplot: Flipper Length by Species")  
plt.show()
```

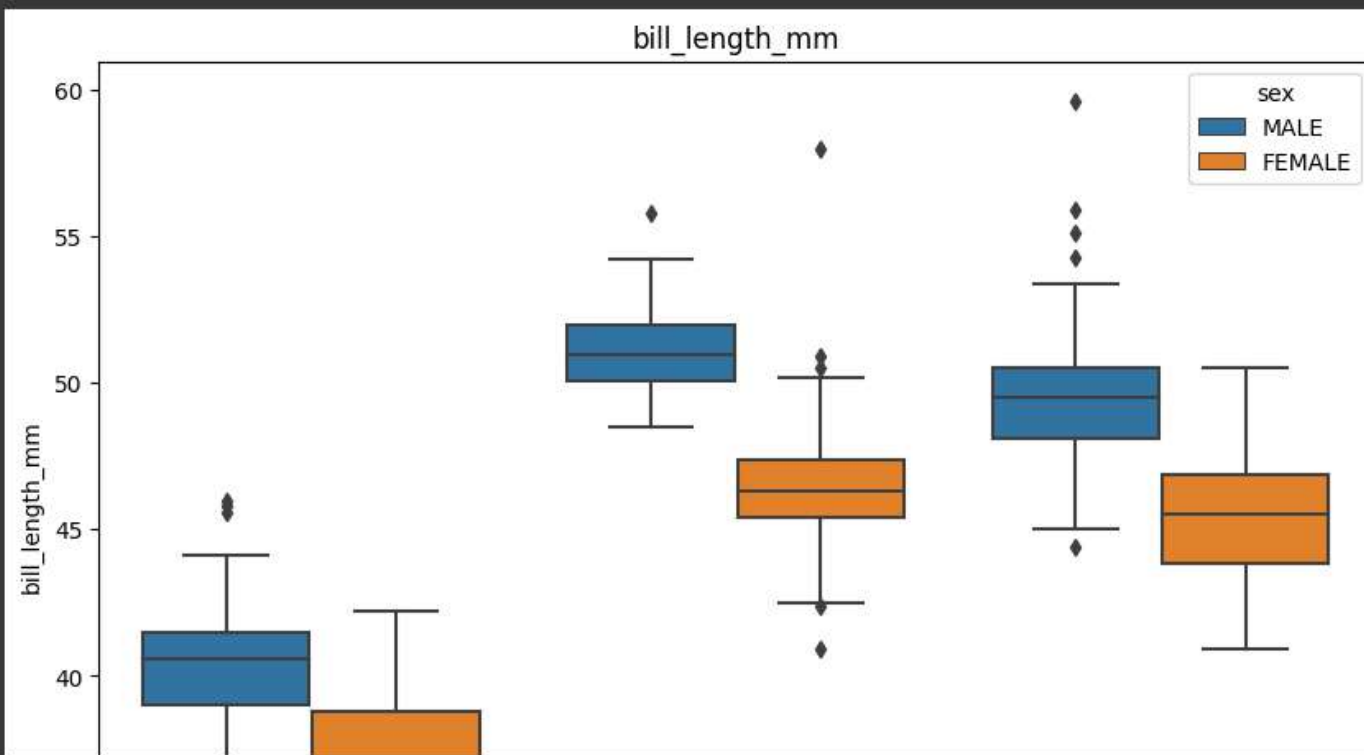




+ Code + Text

Connect

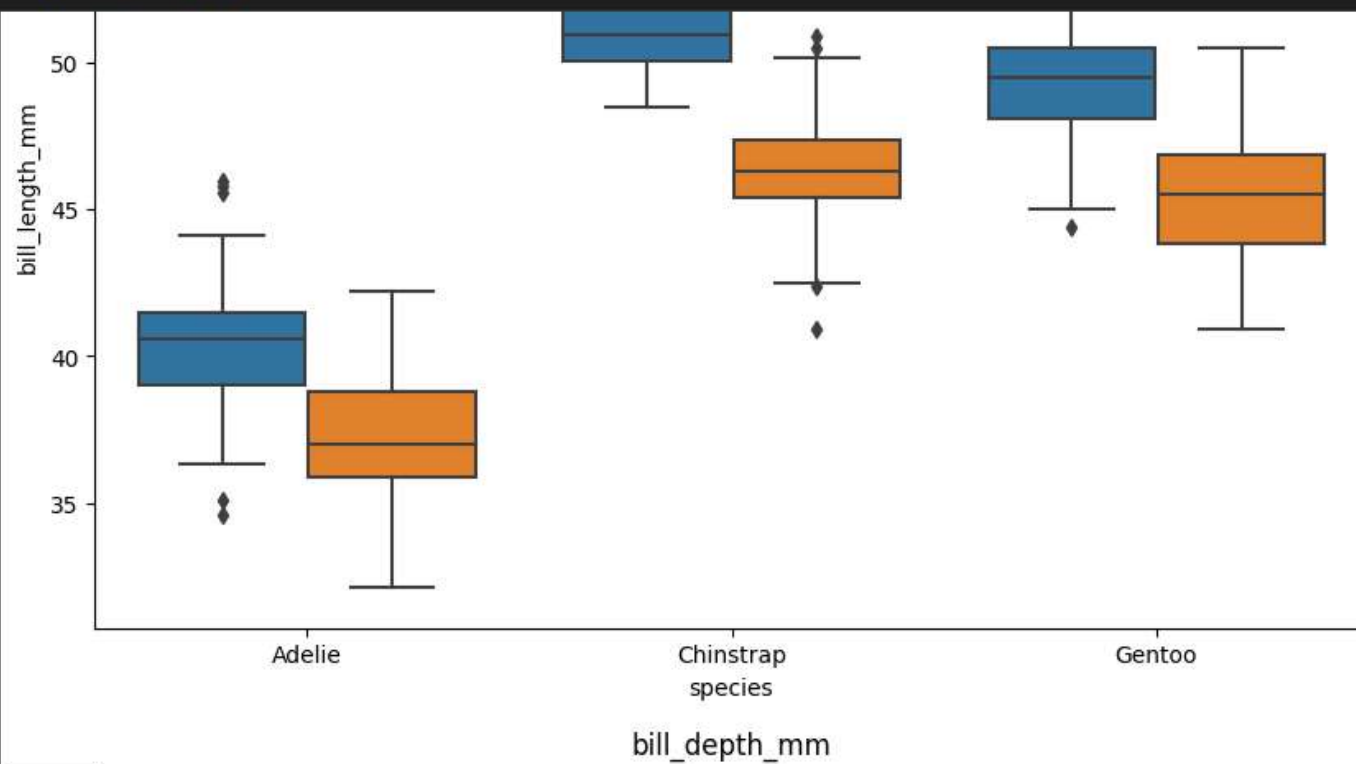
```
box('body_mass_g')
```



+ Code + Text

Connect

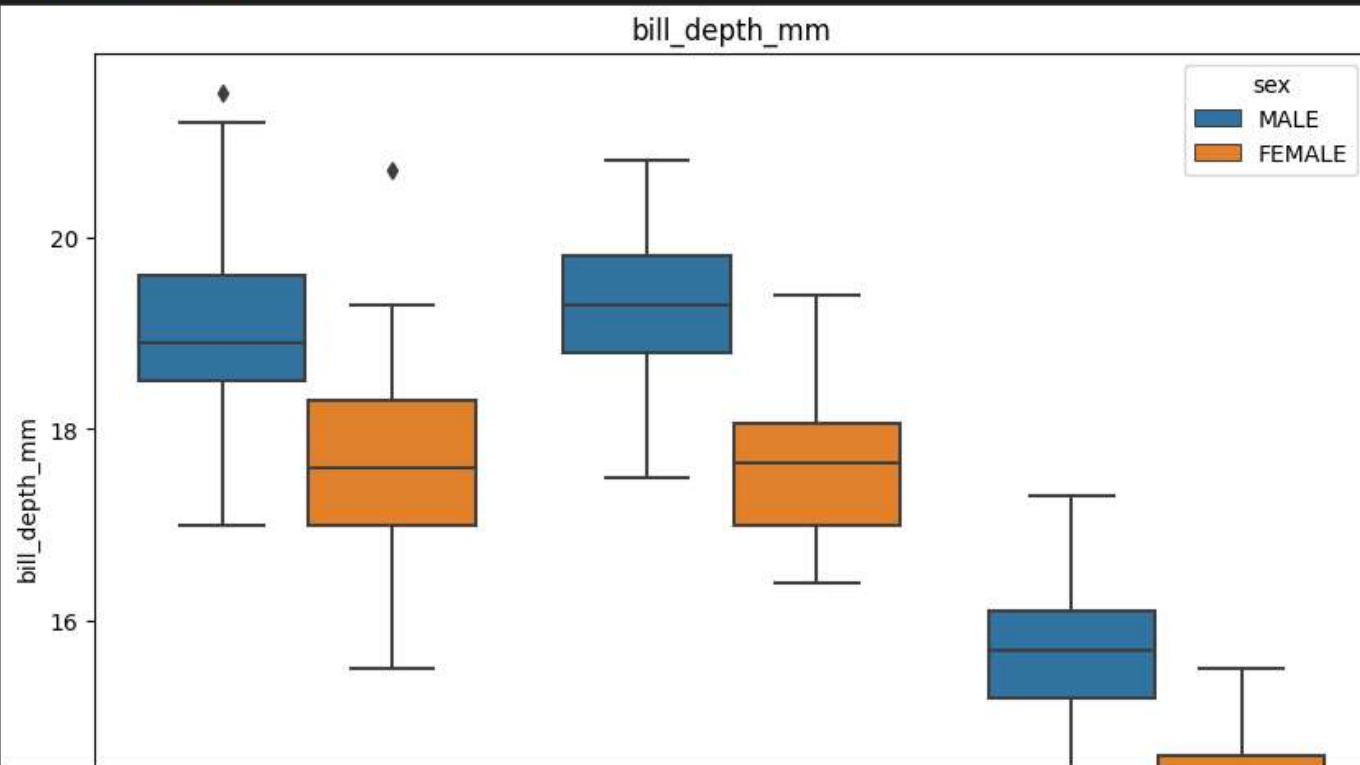
```
box('body_mass_g')
```



+ Code + Text

Connect

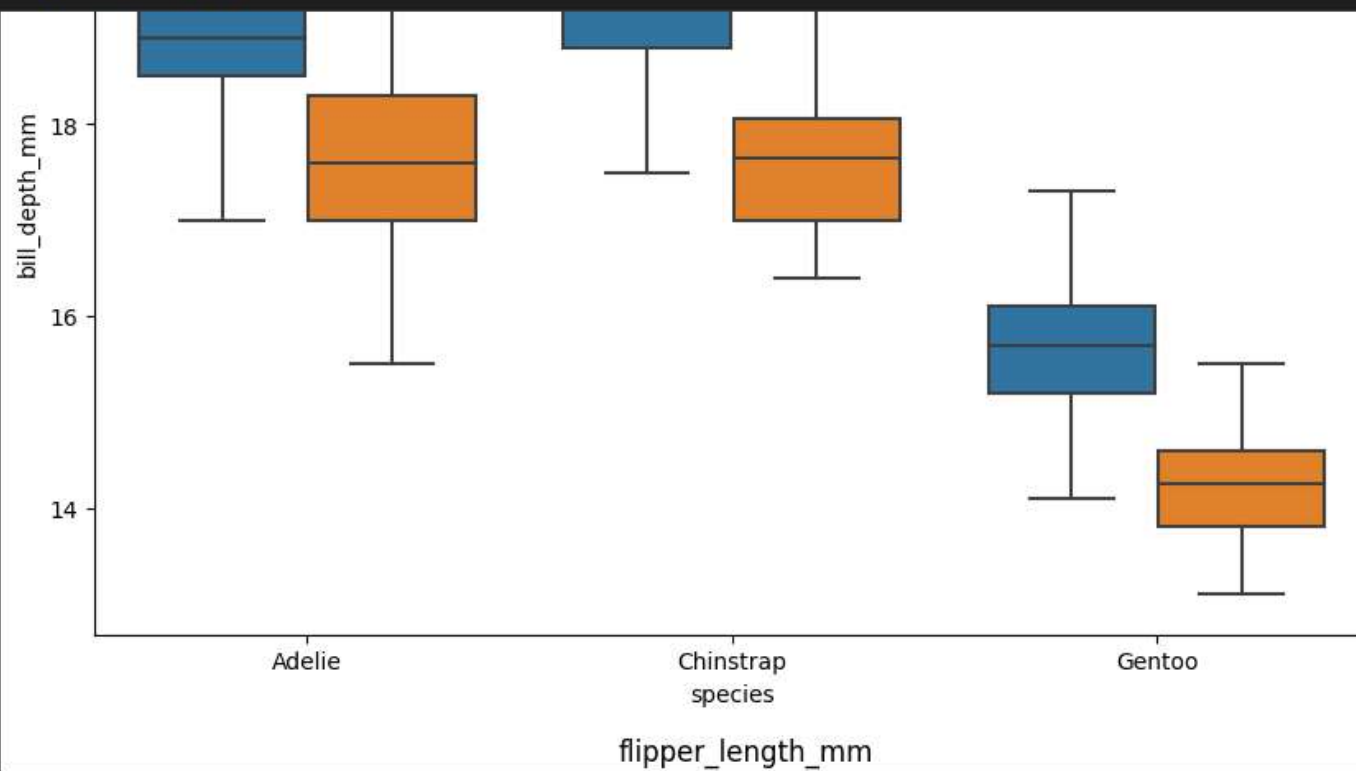
```
box('body_mass_g')
```



+ Code + Text

Connect

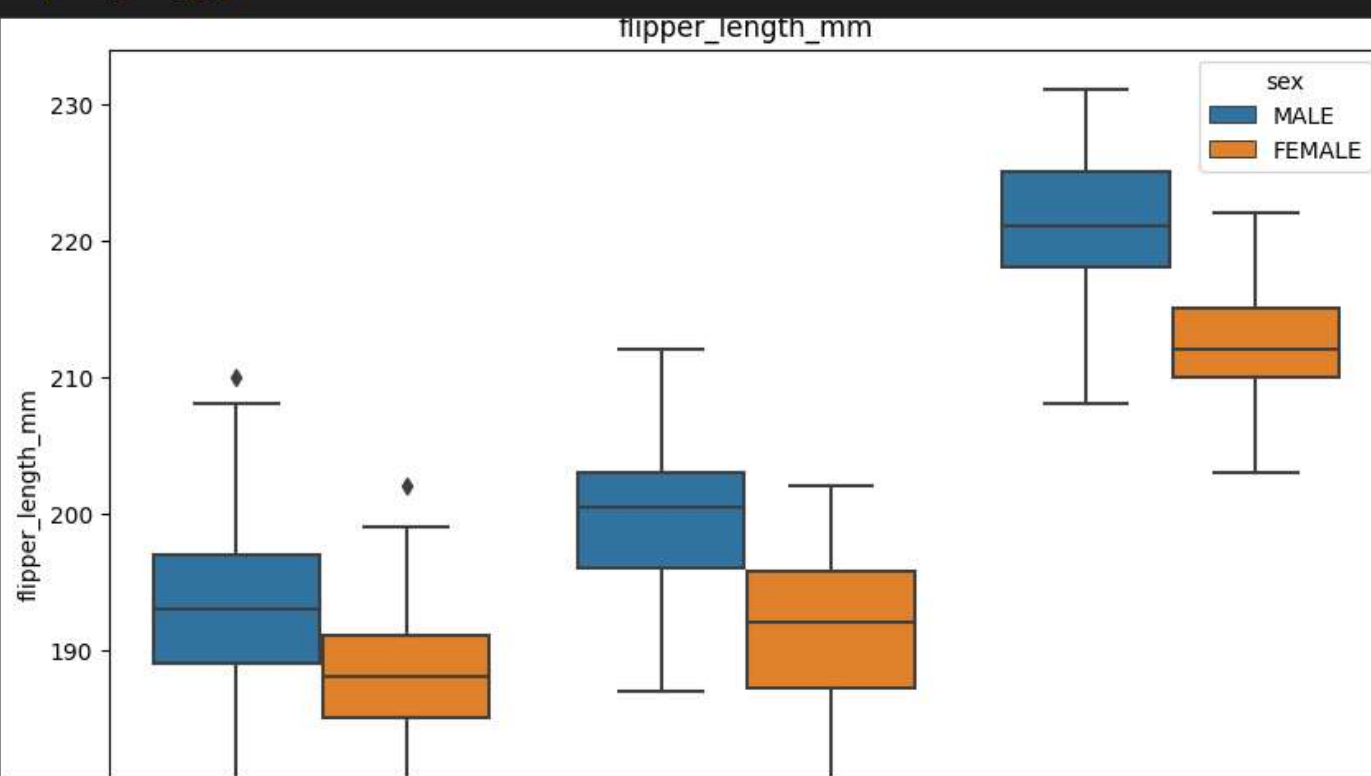
```
box('body_mass_g')
```



+ Code + Text

Connect

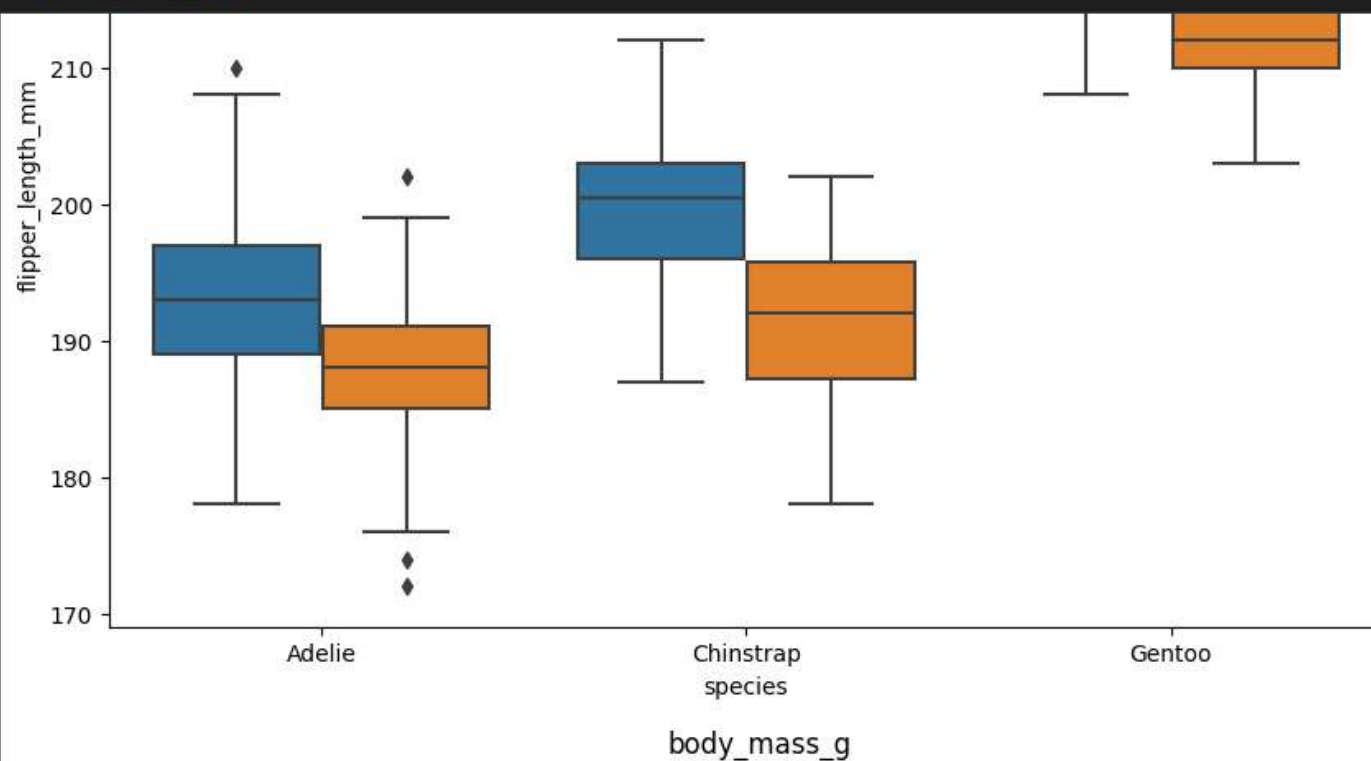
```
box('body_mass_g')
```

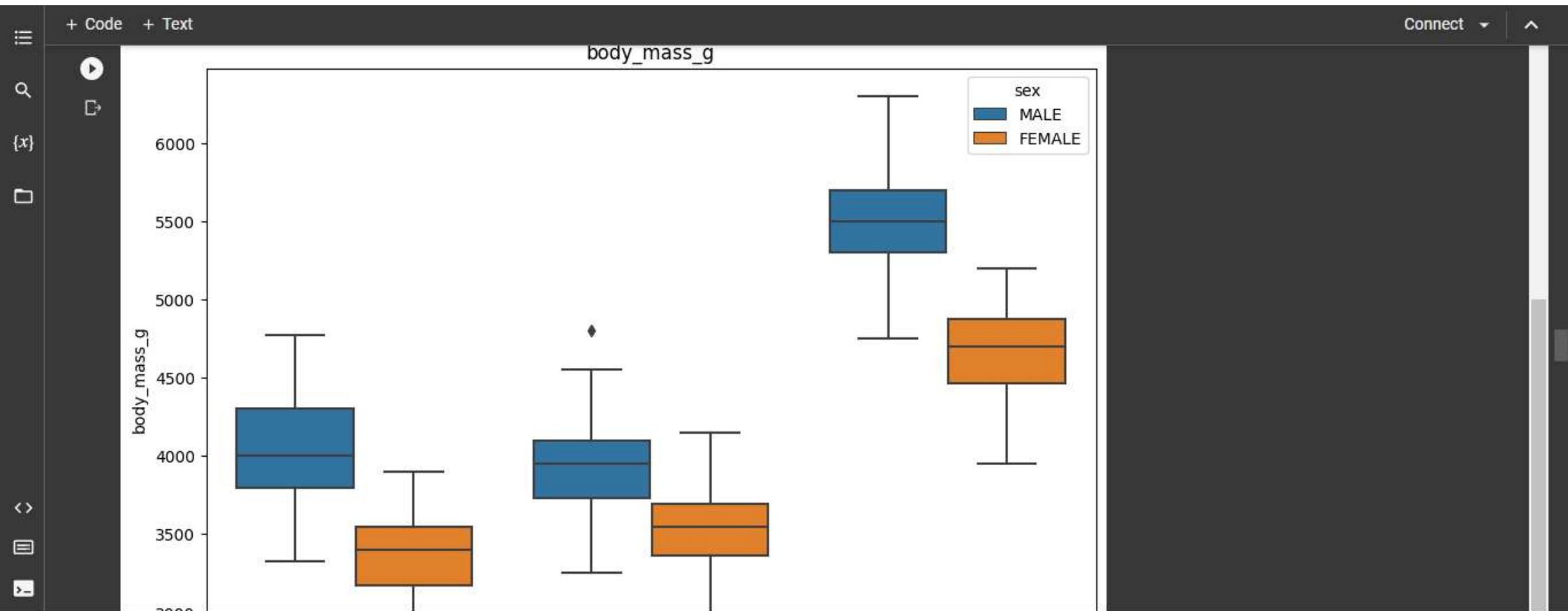


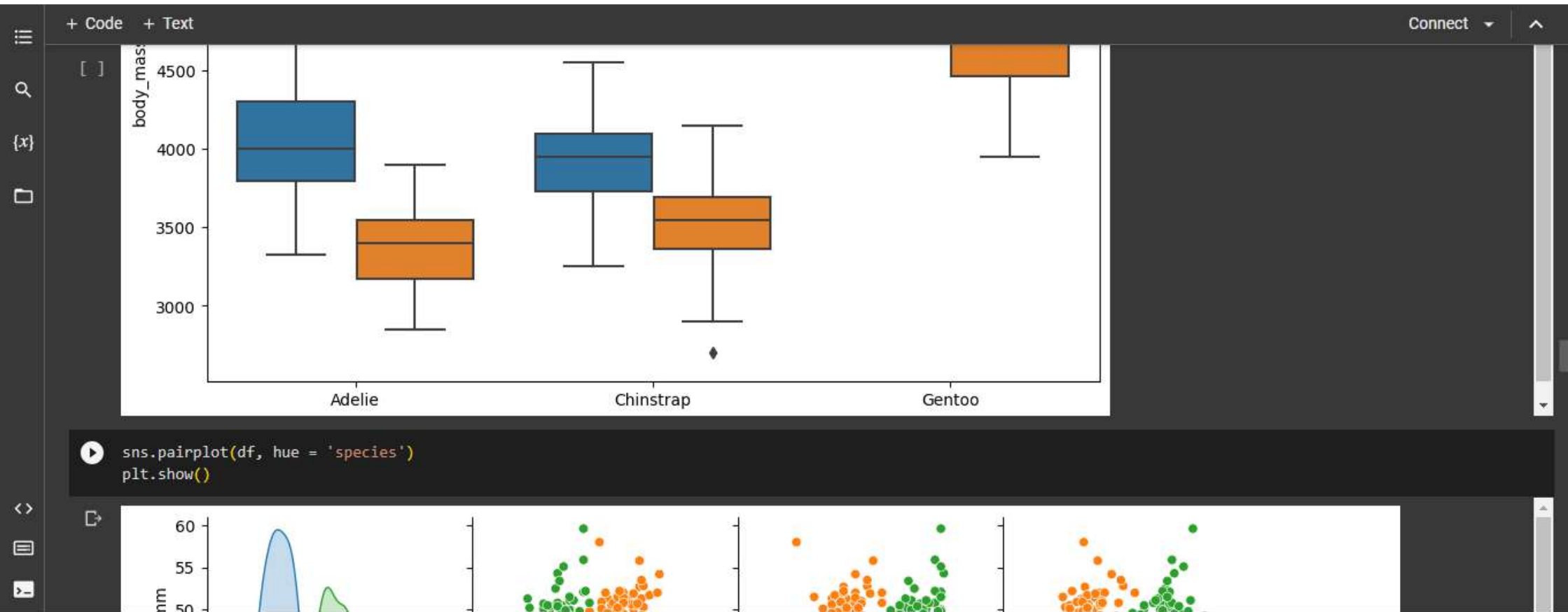
+ Code + Text

Connect

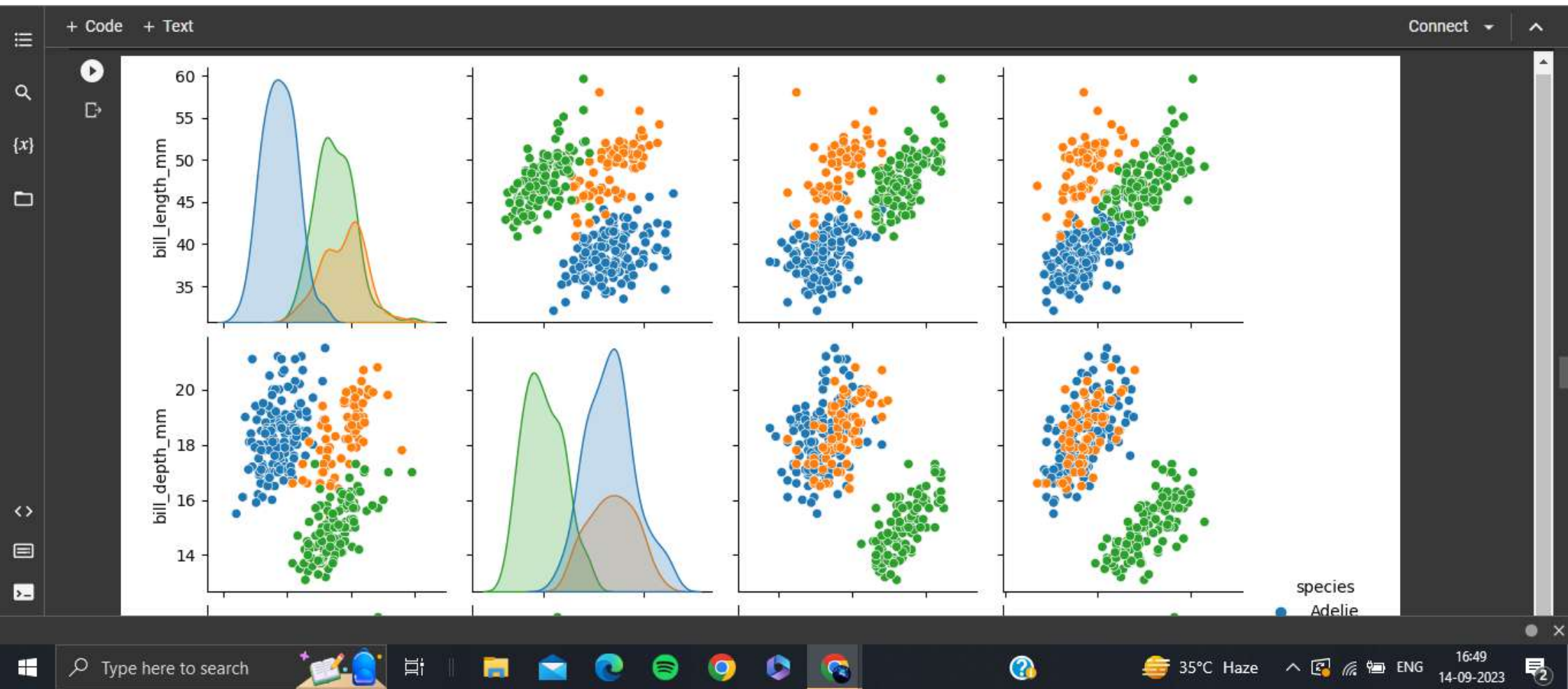
```
box('body_mass_g')
```

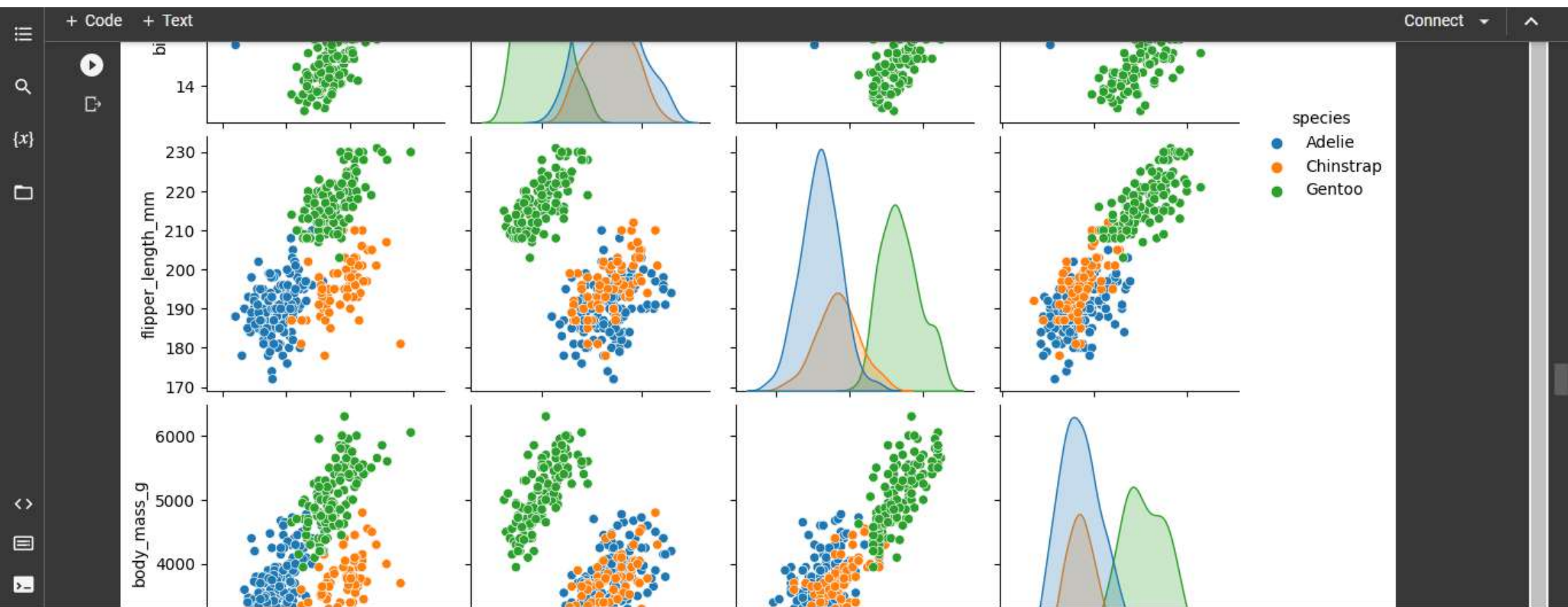


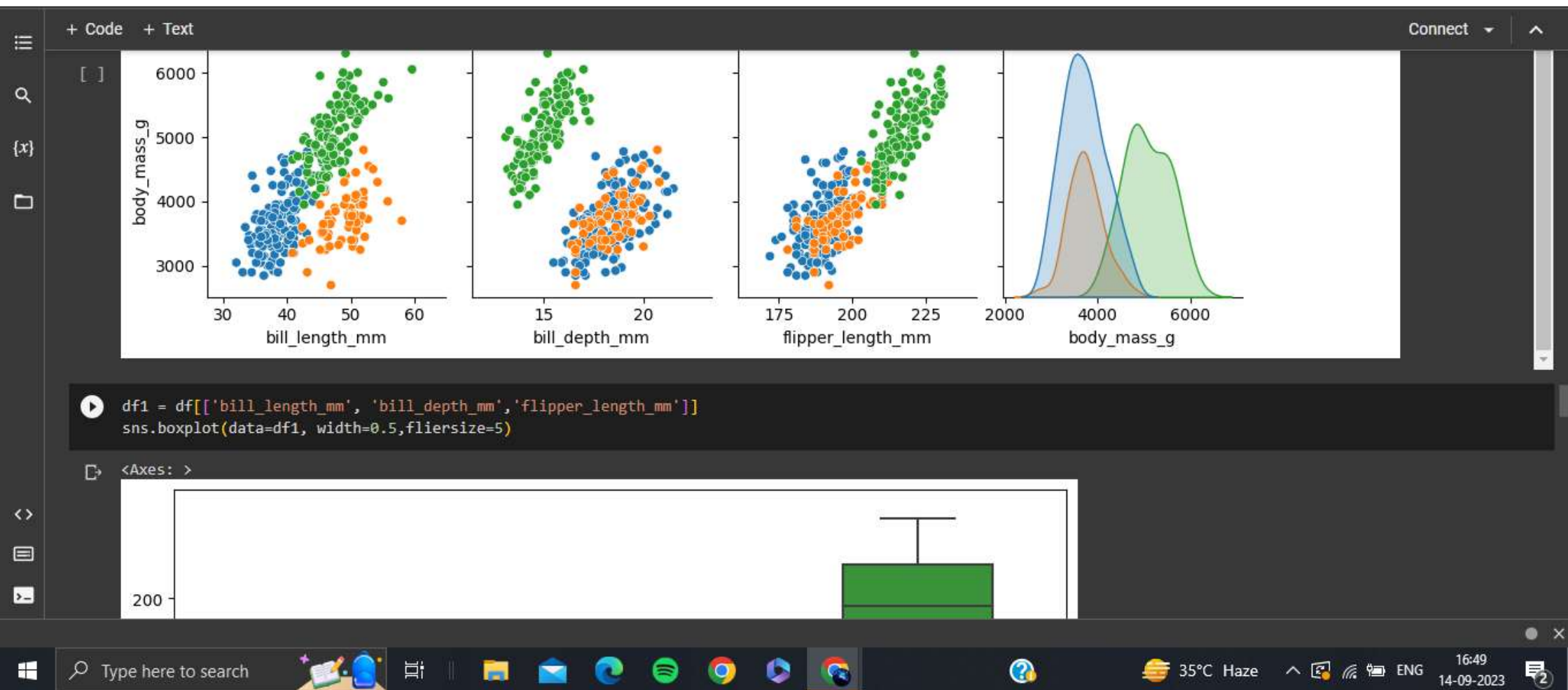


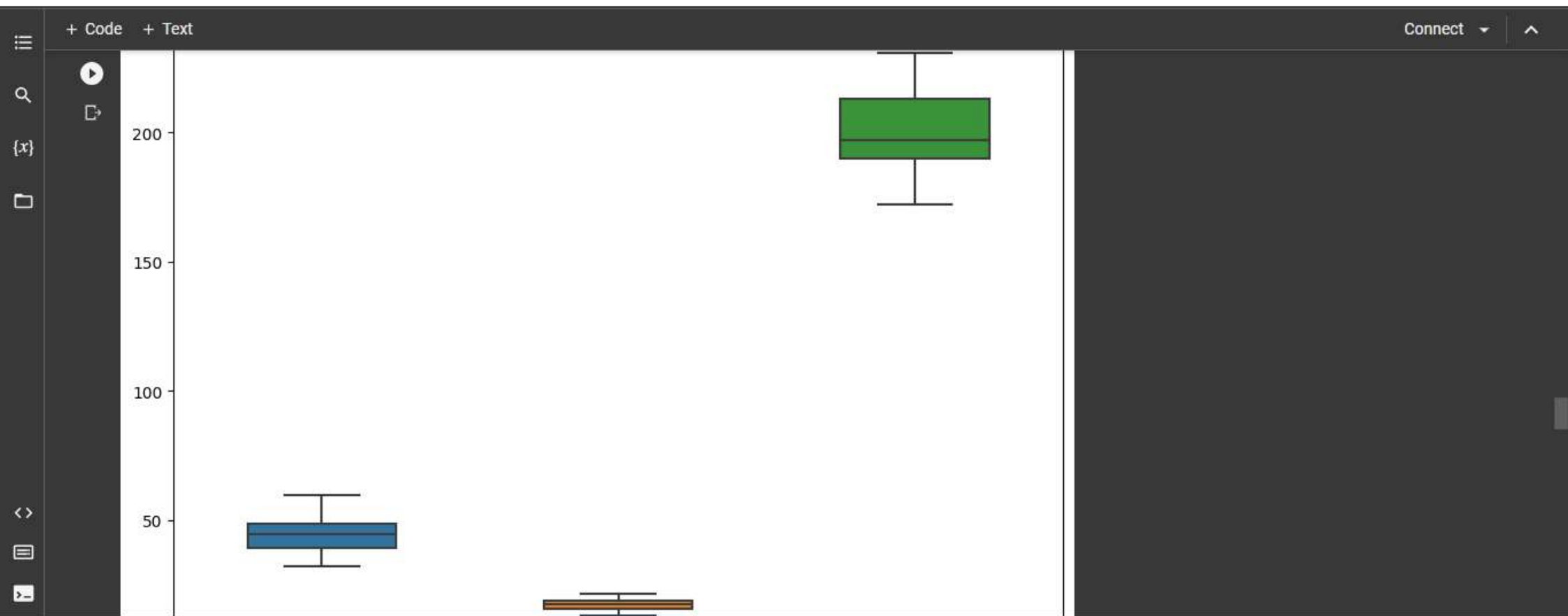


```
sns.pairplot(df, hue = 'species')  
plt.show()
```





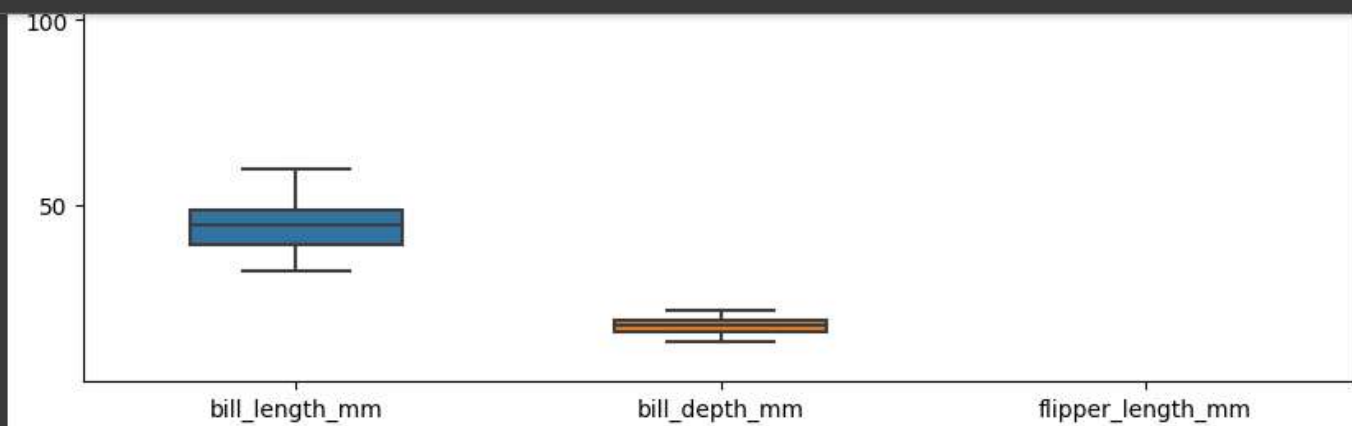




+ Code + Text

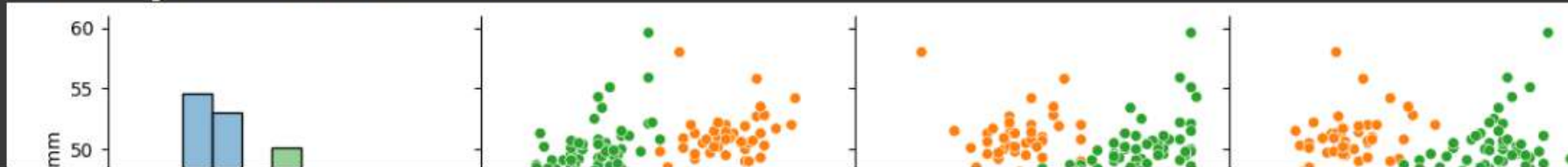
Connect

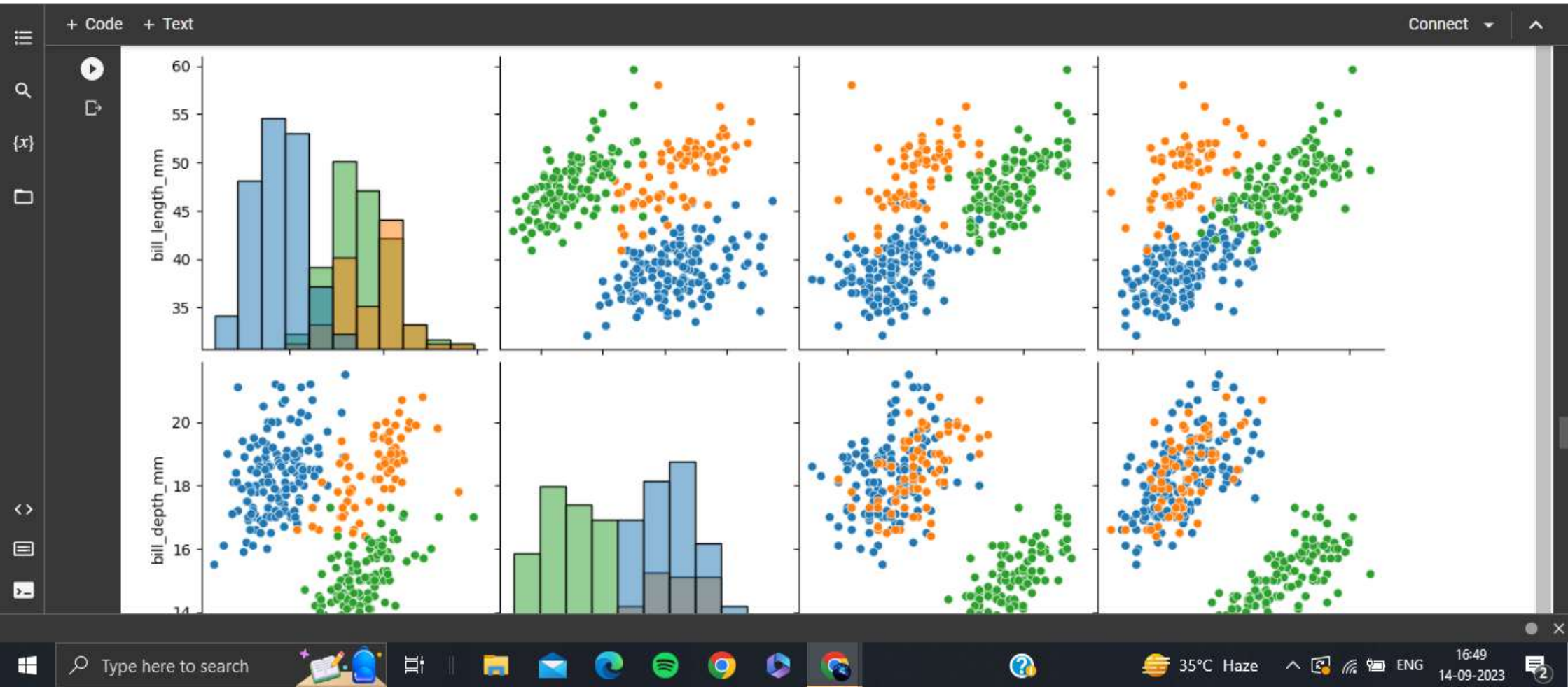
[]



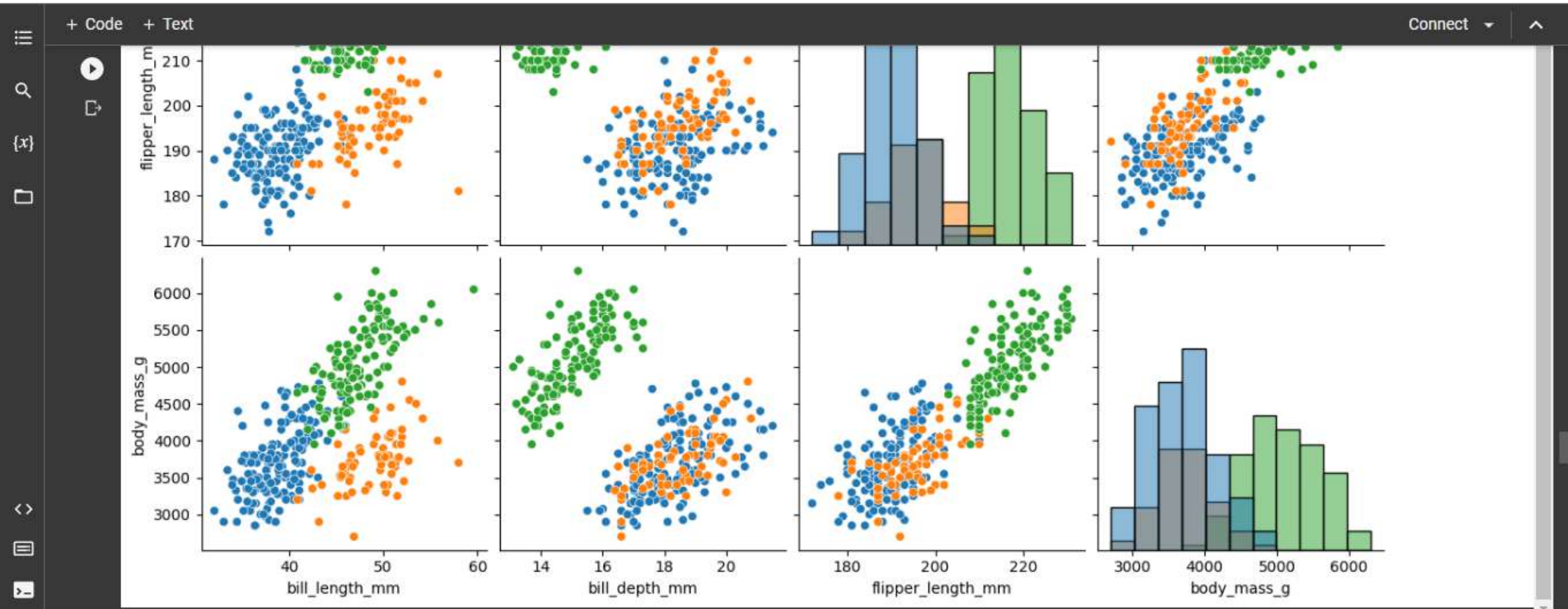
```
sns.pairplot(df, hue="species", size=3,diag_kind="hist")
```

```
/usr/local/lib/python3.10/dist-packages/seaborn/axisgrid.py:2095: UserWarning: The `size` parameter has been renamed to `height`; please update your code.  
warnings.warn(msg, UserWarning)  
<seaborn.axisgrid.PairGrid at 0x78355f96be20>
```









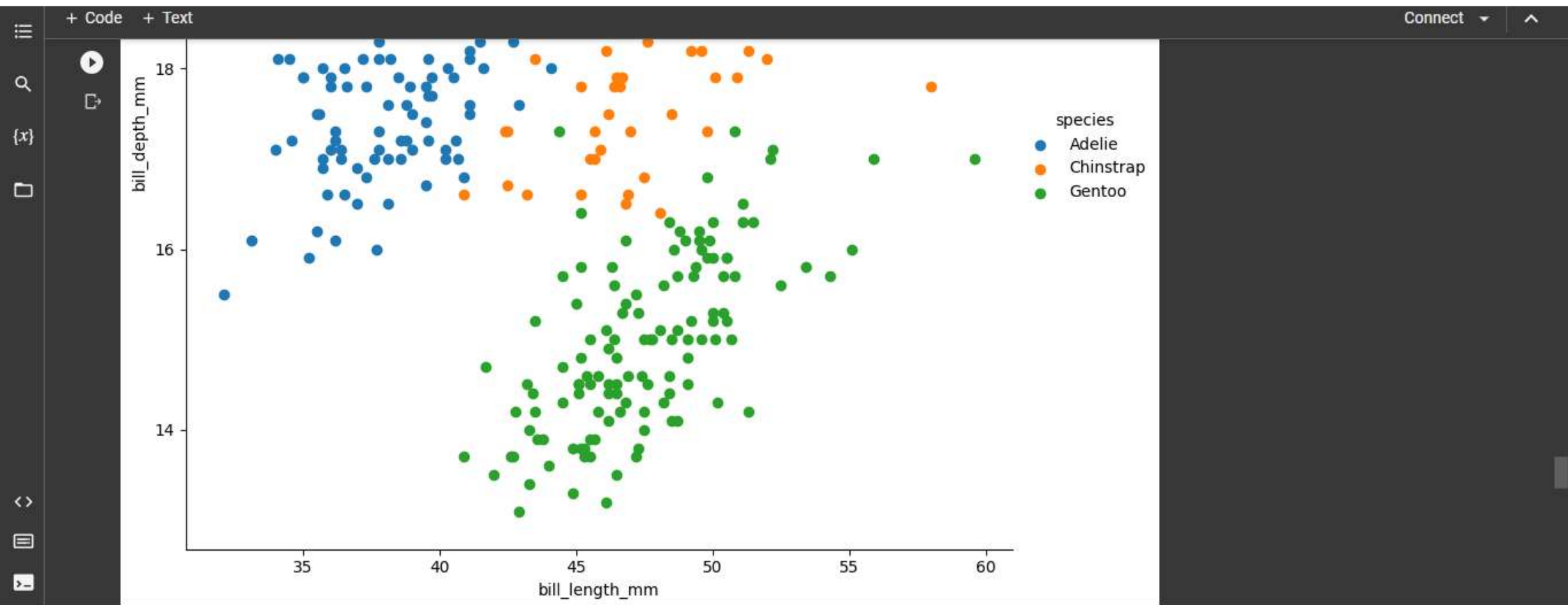
+ Code + Text

Connect ▾

```
▶ sns.FacetGrid(df, hue="species", height=8) \  
  .map(plt.scatter, "bill_length_mm", "bill_depth_mm") \  
  .add_legend()
```

✎ <seaborn.axisgrid.FacetGrid at 0x78355954d5d0>



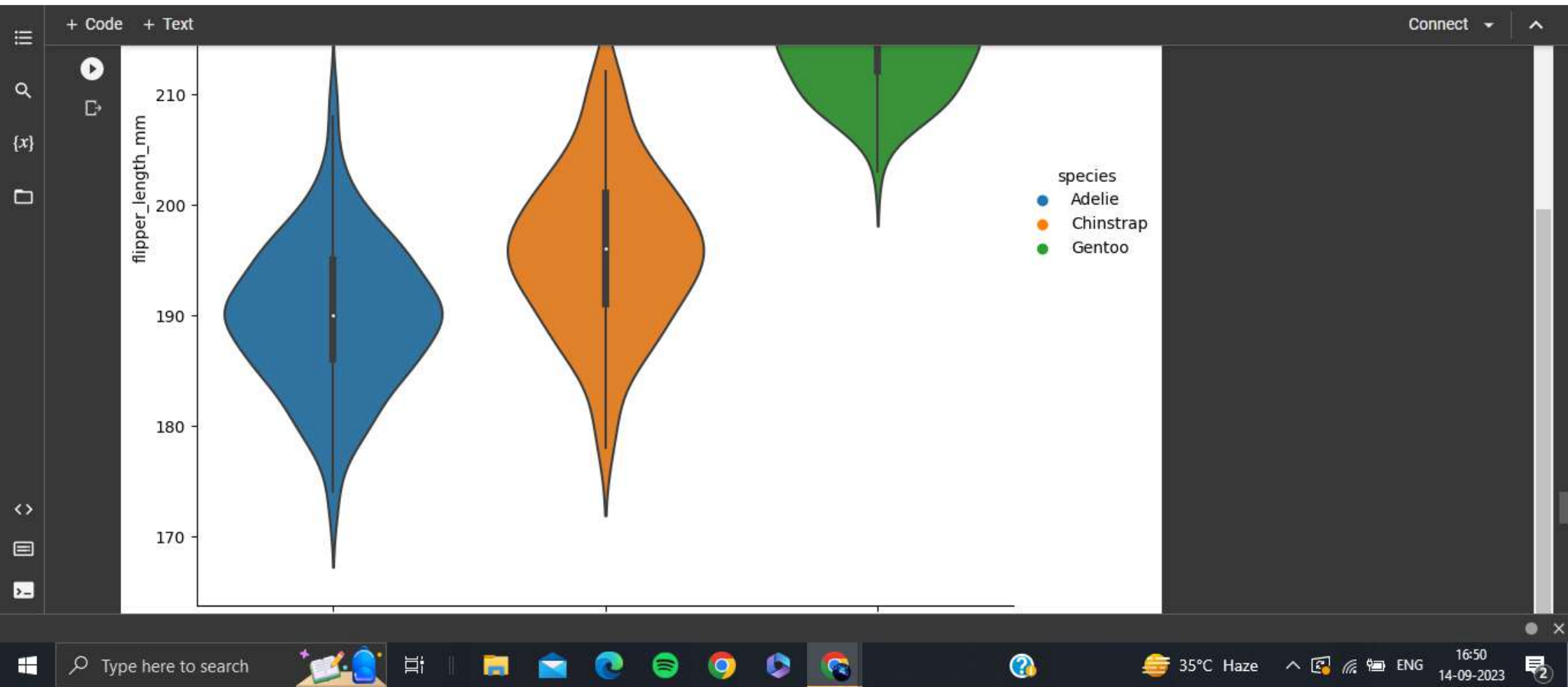


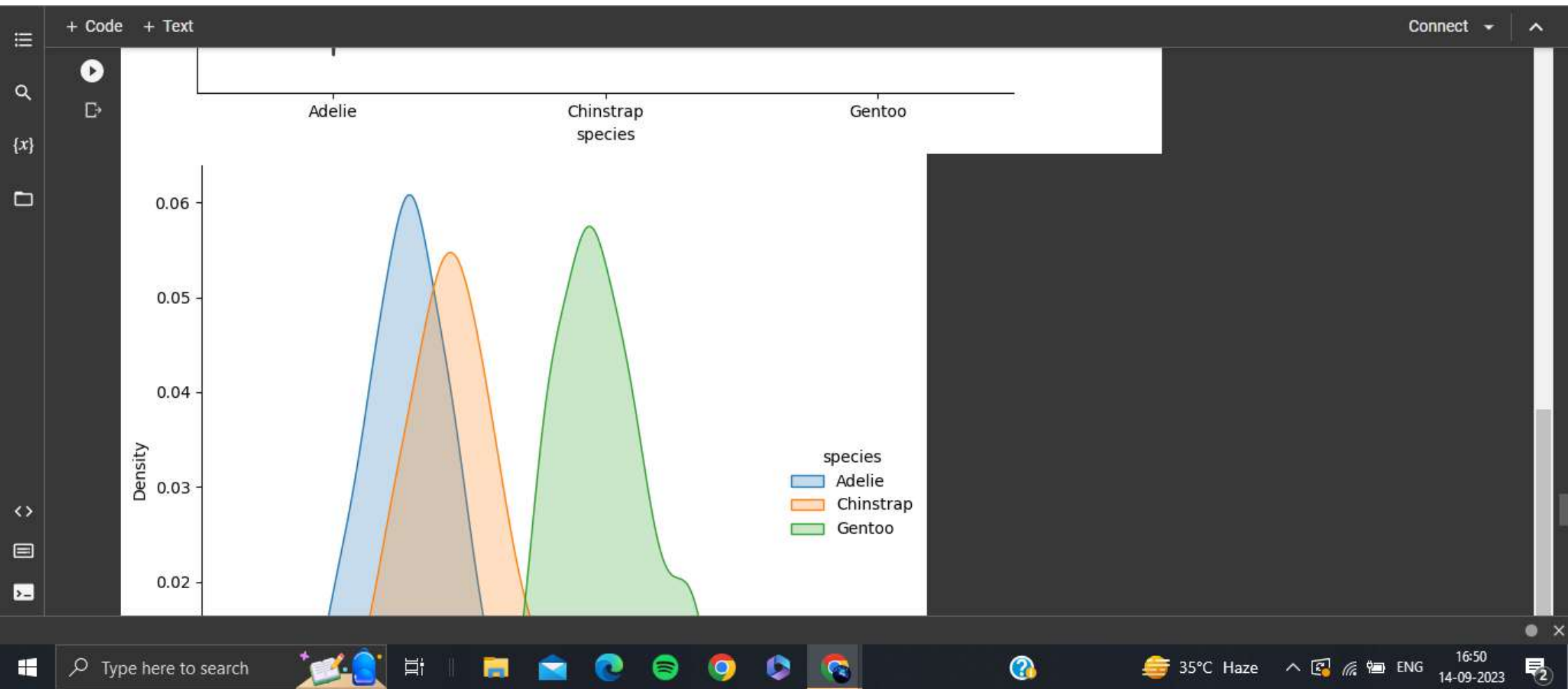
+ Code + Text

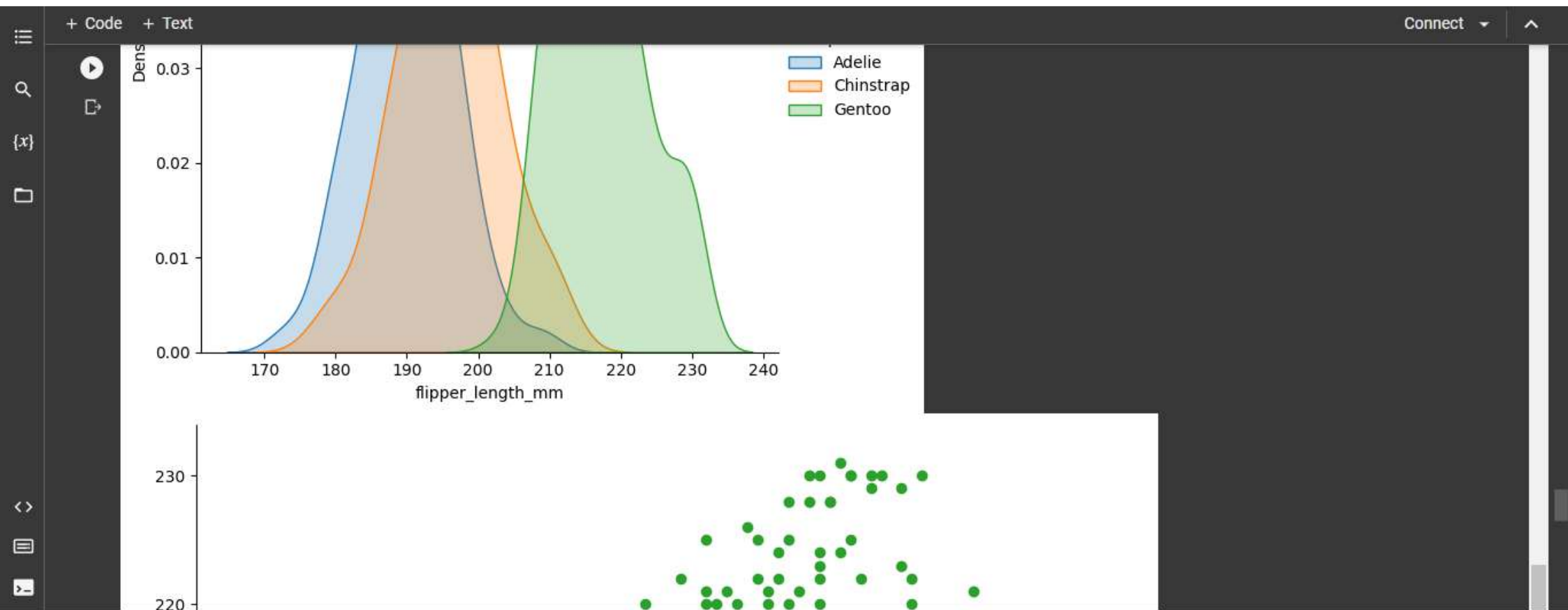
Connect ▾

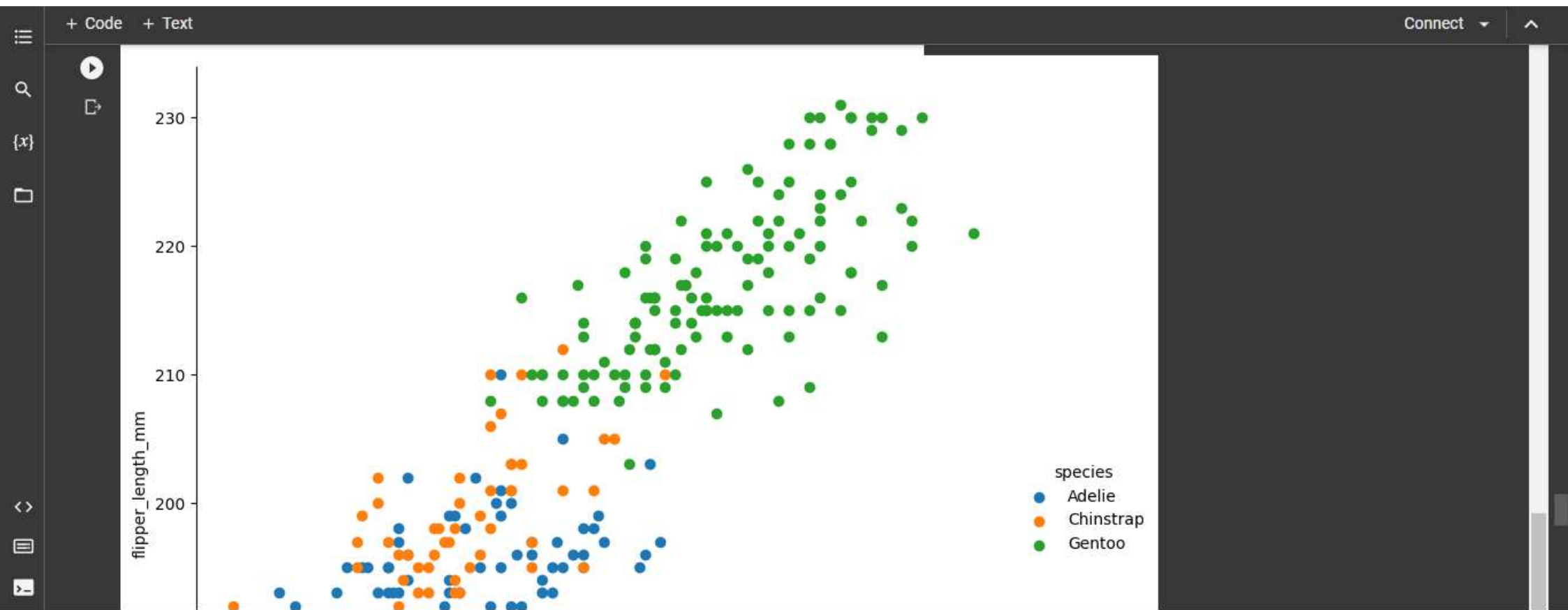
```
▶ sns.FacetGrid(df, hue="species", height=8) \
    .map(plt.scatter, "bill_length_mm", "flipper_length_mm") \
    .add_legend()
ax = sns.violinplot(x="species", y="flipper_length_mm", data=df, height=8)
sns.FacetGrid(df, hue="species", height=6,) \
    .map(sns.kdeplot, "flipper_length_mm", shade=True) \
    .add_legend()
sns.FacetGrid(df, hue="species", height=8) \
    .map(plt.scatter, "body_mass_g", "flipper_length_mm") \
    .add_legend()
```





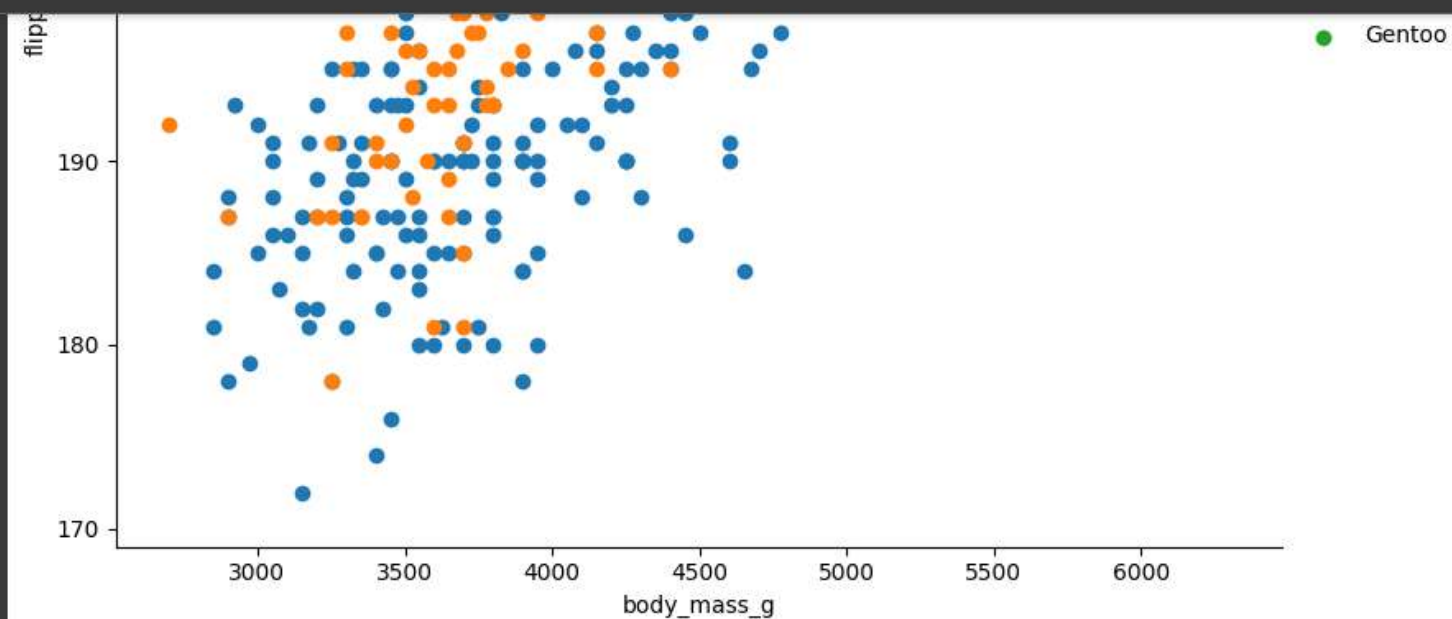




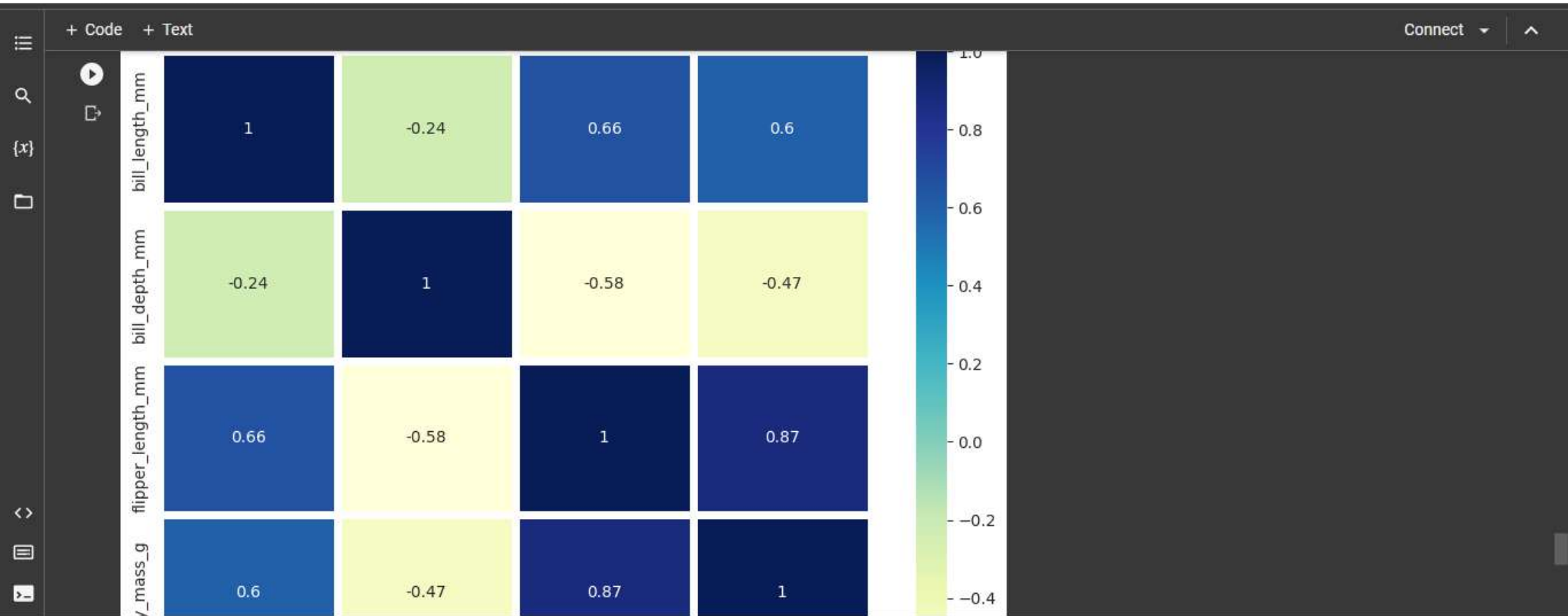


+ Code + Text

Connect

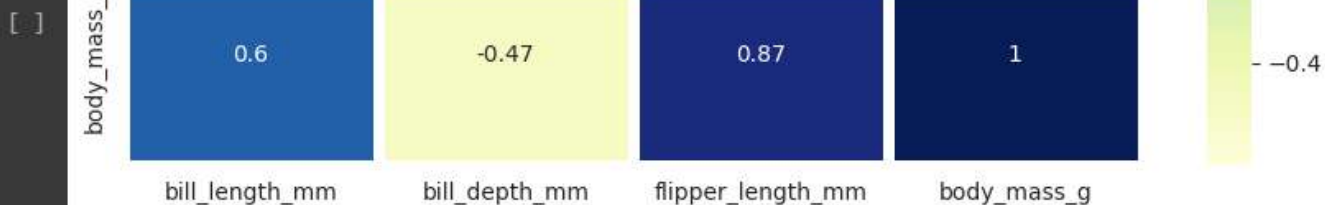


```
[ ] correlation_matrix=df.corr()  
correlation_matrix  
sns.set_style('dark')  
sns.heatmap(correlation_matrix,annot=True,linecolor='white',linewidths=5,cmap="YlGnBu")
```



+ Code + Text

Connect



```
def missing_values_table(df):
    mis_val = df.isnull().sum()

    mis_val_percent = 100 * df.isnull().sum() / len(df)

    mis_val_table = pd.concat([mis_val, mis_val_percent], axis=1)

    mis_val_table_ren_columns = mis_val_table.rename(
        columns = {0 : 'Missing Values', 1 : '% of Total Values'})

    mis_val_table_ren_columns = mis_val_table_ren_columns[
        mis_val_table_ren_columns.iloc[:,1] != 0].sort_values(
        '% of Total Values', ascending=False).round(1)

    print ("Your selected dataframe has " + str(df.shape[1]) + " columns.\n"
          "There are " + str(mis_val_table_ren_columns.shape[0]) +
          " columns that have missing values.")
```


+ Code + Text

Connect



```
mis_val_table_ren_columns = mis_val_table.rename(  
    columns = {0 : 'Missing Values', 1 : '% of Total Values'})
```

```
mis_val_table_ren_columns = mis_val_table_ren_columns[  
    mis_val_table_ren_columns.iloc[:,1] != 0].sort_values(  
    '% of Total Values', ascending=False).round(1)
```

```
print ("Your selected dataframe has " + str(df.shape[1]) + " columns.\n"  
      " columns that have missing values.")
```

```
return mis_val_table_ren_columns
```

```
missing= missing_values_table(df)  
missing
```

```
"There are " + str(mis_val_table_ren_columns.shape[0]) +
```

Your selected dataframe has 7 columns.
There are 5 columns that have missing values.

Missing Values % of Total Values

	Missing Values	% of Total Values
sex	11	3.2
bill_length_mm	2	0.6
bill_depth_mm	2	0.6
flipper_length_mm	2	0.6
body_mass_g	2	0.6

