

pr-8

October 15, 2023

1 Graph plotting with Matplotlib

```
[97]: import pandas as pd
      from matplotlib import pyplot as plt
      import numpy as np

      iris = pd.read_csv('iris.csv')
      x = iris['SepalLengthCm']
      y = iris['SepalWidthCm']
      dimension_2 = iris[['SepalLengthCm', 'SepalWidthCm']]
```

```
[98]: # Set up the plot
      fig, ax = plt.subplots()

      # Plot the line with extra arguments and styling
      ax.plot(x, y, color='red', linestyle='dashed', linewidth=2, marker='o',
              ↪ markersize=10, markerfacecolor='blue', markeredgecolor='black')

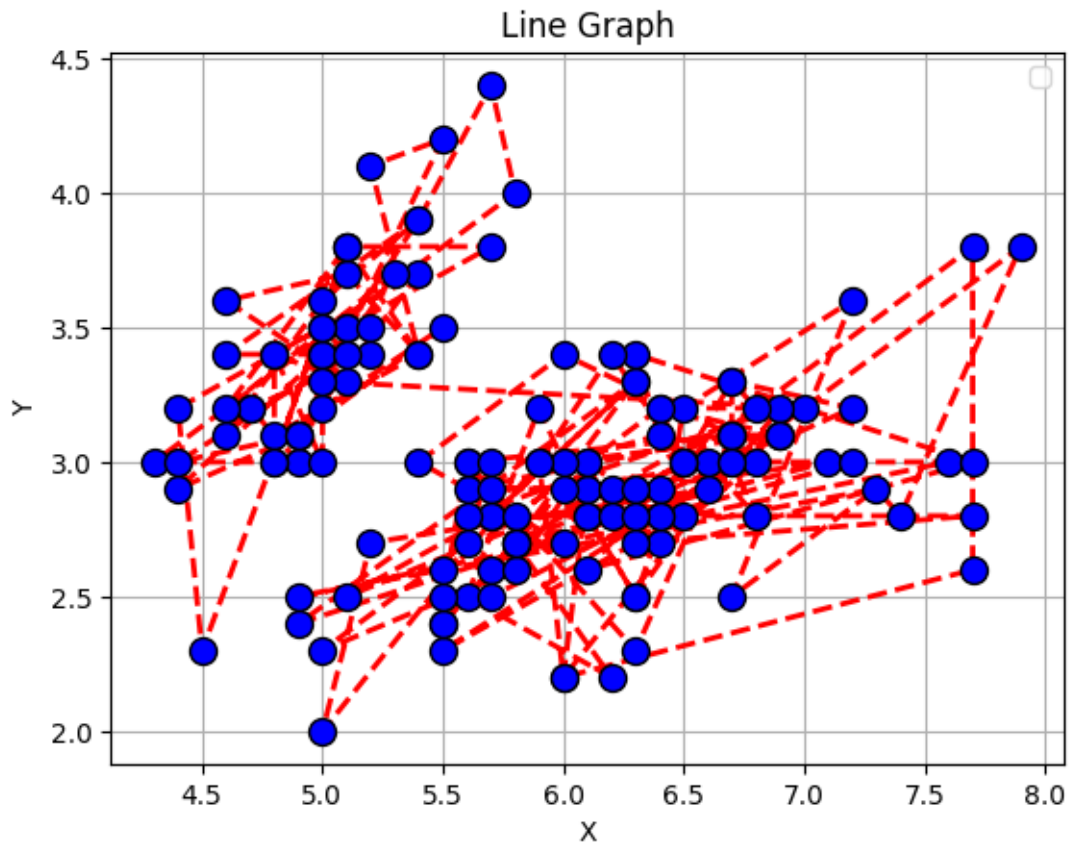
      # Set the axis labels and title
      ax.set_xlabel('X')
      ax.set_ylabel('Y')
      ax.set_title('Line Graph')

      # Add a grid
      ax.grid(True)

      # Add a legend
      ax.legend()

      # Show the plot
      plt.show()
```

No artists with labels found to put in legend. Note that artists whose label start with an underscore are ignored when legend() is called with no argument.



```
[99]: # Create the bar graph
plt.bar(x, y,)

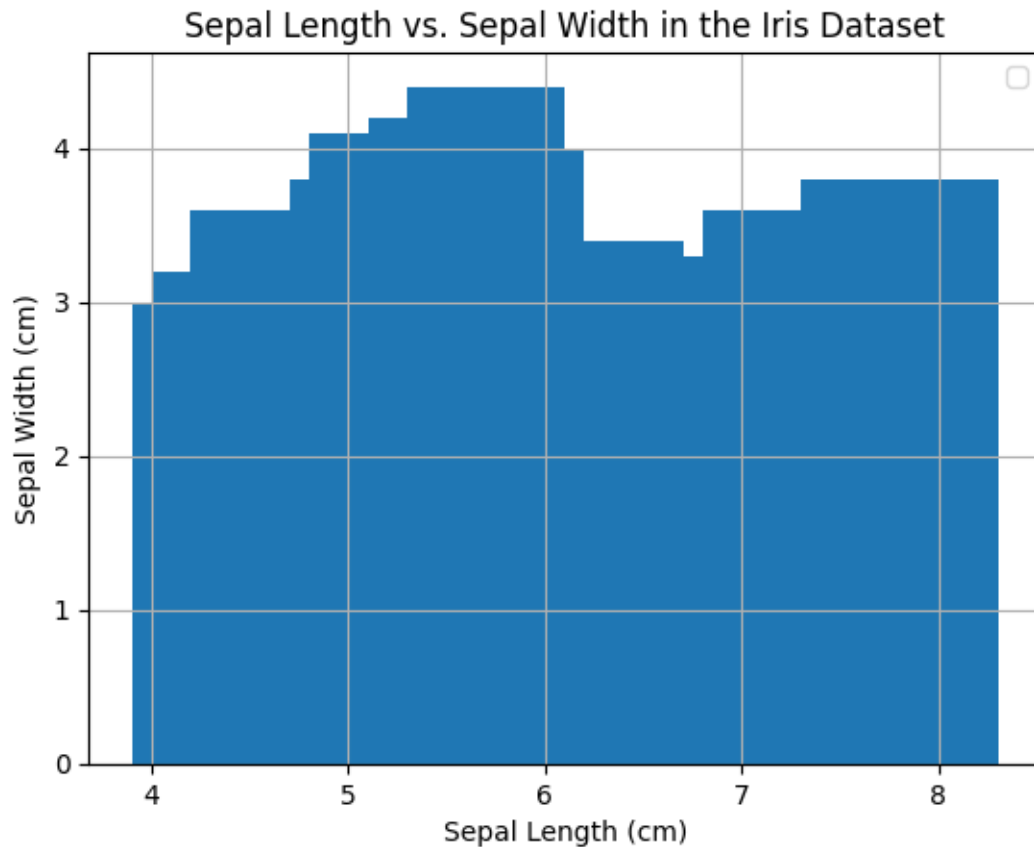
# Set the axis labels and title
plt.xlabel('Sepal Length (cm)')
plt.ylabel('Sepal Width (cm)')
plt.title('Sepal Length vs. Sepal Width in the Iris Dataset')

# Add a grid
plt.grid(True)

# Add a legend
plt.legend()

# Show the plot
plt.show()
```

No artists with labels found to put in legend. Note that artists whose label start with an underscore are ignored when legend() is called with no argument.



```
[100]: # 3. Scatter Plot

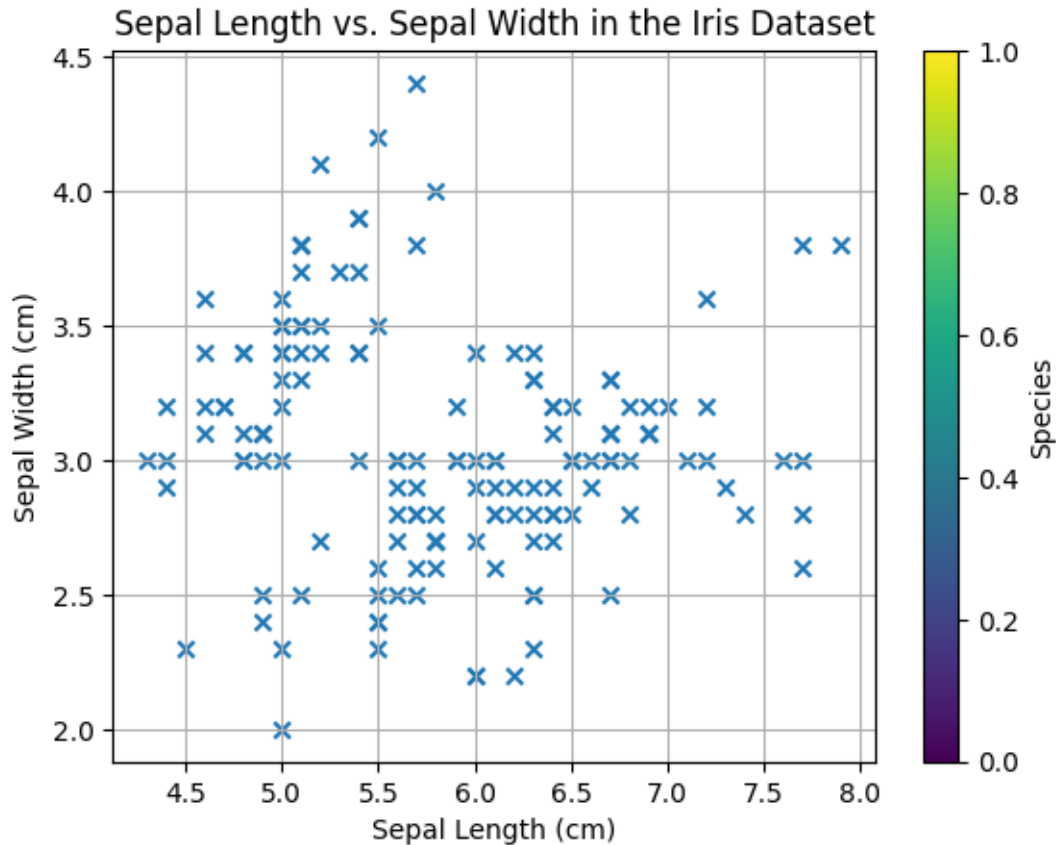
# Create the scatter plot
plt.scatter(x, y, marker='x')

# Set the axis labels and title
plt.xlabel('Sepal Length (cm)')
plt.ylabel('Sepal Width (cm)')
plt.title('Sepal Length vs. Sepal Width in the Iris Dataset')

# Add a grid
plt.grid(True)

# Add a colorbar
plt.colorbar(label='Species')

# Show the plot
plt.show()
```



```
[101]: # 4. Pie Chart

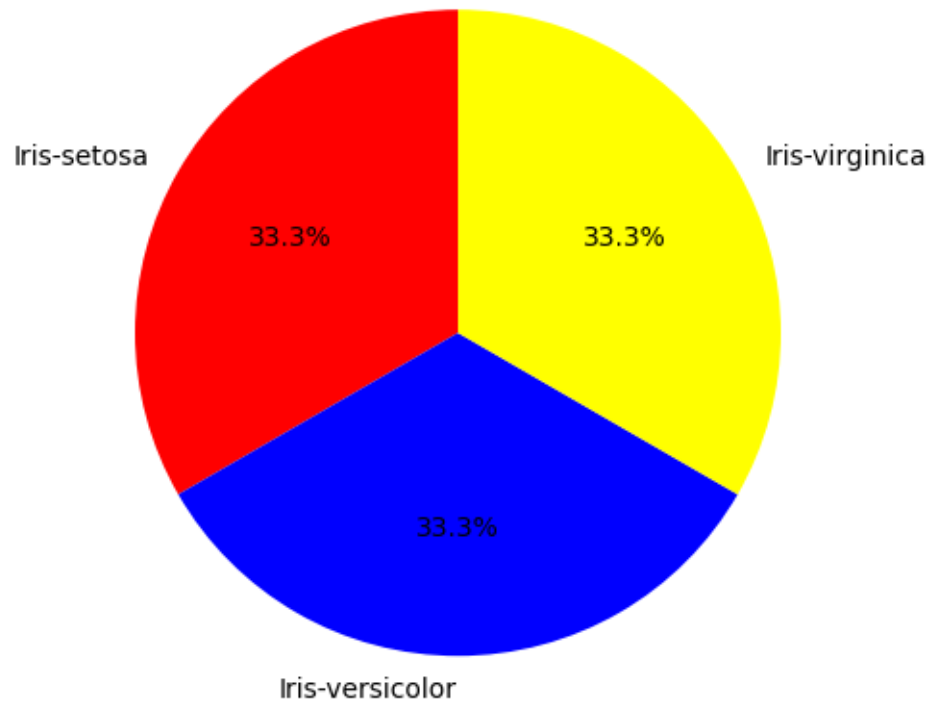
# Calculate the percentage of flowers for each species
iris_species_percentages = iris['Species'].value_counts(normalize=True) * 100

# Get pie chart slice labels
pie_chart_slice_labels = iris_species_percentages.index.to_numpy()

# Get pie chart slice values
pie_chart_slice_values = iris_species_percentages.to_numpy()

# Create a pie chart of the percentage of flowers for each species
plt.pie(pie_chart_slice_values, labels=pie_chart_slice_labels, autopct="%1.1f%%", startangle=90, colors=['Red', 'Blue', 'Yellow'])
plt.title("Pie Chart of Iris Species Distribution (Percentage)")
plt.axis('equal') # Equal aspect ratio ensures a circular pie chart
plt.show()
```

Pie Chart of Iris Species Distribution (Percentage)



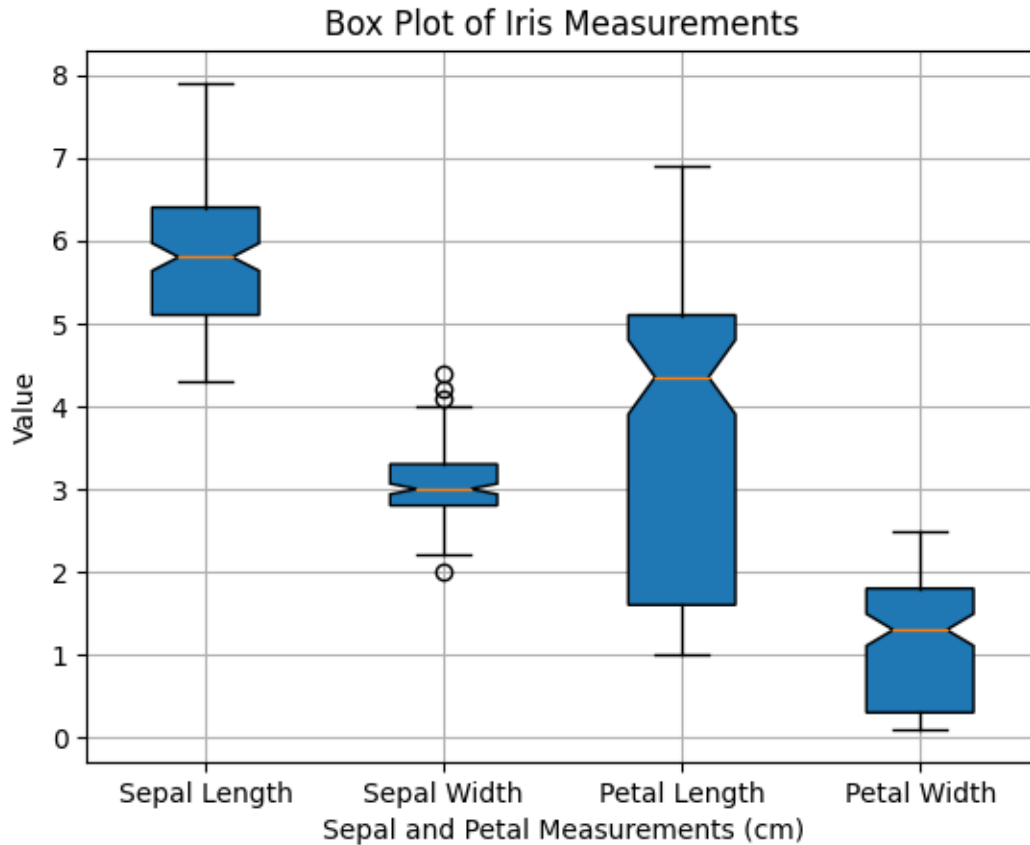
```
[102]: # 5. Box Plot

# Create the box plot
plt.boxplot([iris['SepalLengthCm'], iris['SepalWidthCm'],
             iris['PetalLengthCm'], iris['PetalWidthCm']], labels=['Sepal Length', 'Sepal Width', 'Petal Length', 'Petal Width'], patch_artist=True, notch=True)

# Set the axis labels and title
plt.xlabel('Sepal and Petal Measurements (cm)')
plt.ylabel('Value')
plt.title('Box Plot of Iris Measurements')

# Add a grid
plt.grid(True)

# Show the plot
plt.show()
```



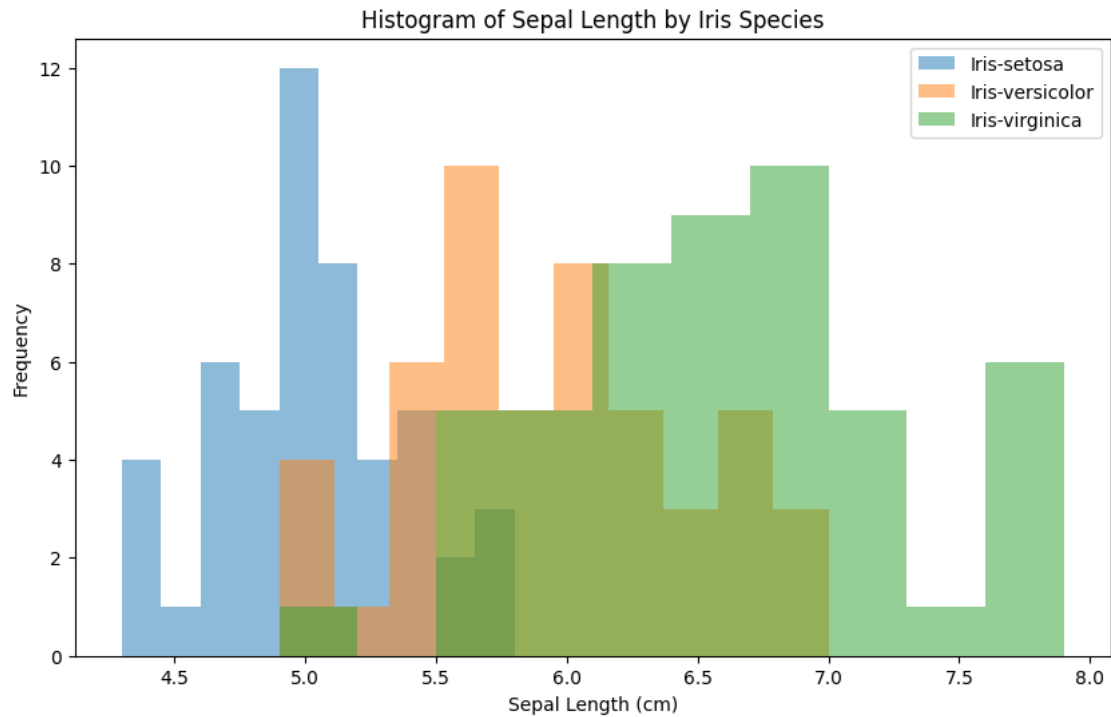
```
[103]: # 6. Histogram

# Create a histogram of the Sepal Length measurements for each iris species
plt.figure(figsize=(10, 6))
for species in iris['Species'].unique():
    sepal_lengths = iris[iris['Species'] == species]['SepalLengthCm']
    plt.hist(sepal_lengths, bins=10, alpha=0.5, label=species)

# Set the axis labels and title
plt.xlabel('Sepal Length (cm)')
plt.ylabel('Frequency')
plt.title('Histogram of Sepal Length by Iris Species')

# Add a legend
plt.legend()

# Show the plot
plt.show()
```



```
[104]: # 7. Violin Plot

# Create the violin plot
plt.violinplot(
    [iris['SepalLengthCm'], iris['SepalWidthCm'], iris['PetalLengthCm'],
    iris['PetalWidthCm']],
    positions=[1, 2, 3, 4],
    showmeans=True,
    showextrema=True,
    vert=True,
    bw_method='silverman',
)

# Set the axis labels and title
plt.xlabel('Iris Measurement')
plt.ylabel('Value (cm)')
plt.title('Violin Plot of Iris Measurements')

# Add a grid
plt.grid(True)

# Show the plot
plt.show()
```

