

Lab – 22

Student Name: Navneet P

Student ID: AF0411619

Topic: Matplotlib Bar Plot and Histogram

Bar Plots:

A bar plot uses bars to represent each category or subcategory, with the length of the bar corresponding to the numeric value. The categorical variable is plotted on one axis, and the numeric value is plotted on the other axis. Bar plots are a common and simple way to compare objects with the same parameters. They can help audiences understand the insights and ideas suggested by the data.

1. **Vertical Bar Plots:** Vertical bar plots represent the data vertically. These are bar graphs whose bars are drawn vertically.
2. **Horizontal Bar Plots:** Horizontal bar graphs represent the data horizontally. These are sideways bar graphs whose bars are drawn horizontally.

Functions used in this assignment:

1. **Bar():** a much more elaborate way to specify xlabels, ylabels, colors, bar labels, is often used when subplots are created initially.
2. **Bar_label():** a much more simpler method to directly add labels to the bars and customize its features such as font size, bar edge color etc.
3. **Stem():** Creates a stem plot, it plots vertical lines from a baseline to the y-coordinate and places a marker at the tip.
4. **Barh():** Creates a horizontal bar plot
5. **Hist():** Creates a histogram which is used to plot continuous values.

Example 1: Small Demo on adding some data labels.

Solution:

We could create subplots and use the `bar()` method to add labels or we could use the latest `bar_label()` method which is a little easier to implement. [Ref](#)

Code:

```
Example 1

# Import the necessary libraries
import matplotlib.pyplot as plt # For plotting
import numpy as np # For numerical operations

# Define species of penguins
species = ('Adelie', 'Chinstrap', 'Gentoo')

# Dictionary holding the counts of males and females for each species
sex_counts = {
    'Male': np.array([73, 34, 61]), # Number of males for each species
    'Female': np.array([73, 34, 58]), # Number of females for each species
}

# Set the width of the bars in the bar chart
width = 0.6 # Width of the bars, can be adjusted for visual clarity

# Create a new figure and axes for the plot
fig, ax = plt.subplots()

# Initialize the bottom variable to track the stacking of the bars
# Starts with zeros because we haven't plotted anything yet
bottom = np.zeros(3)

# Loop over the dictionary containing the counts for each sex
for sex, sex_count in sex_counts.items():
    # Create a bar plot for the current sex (Male or Female)
    p = ax.bar(species, sex_count, width, label=sex, bottom=bottom)

    # Update the bottom values for the next bar stack
    bottom += sex_count

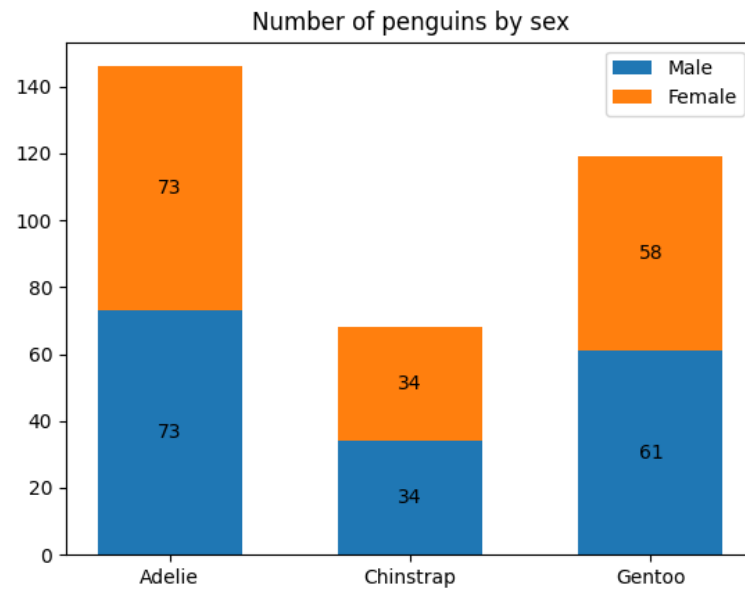
    # Add labels to the bars showing the values (centered on the bar)
    ax.bar_label(p, label_type='center')

# Set the title of the plot
ax.set_title('Number of penguins by sex')

# Add a legend to indicate which bar corresponds to which sex
ax.legend()

# Display the final plot
plt.show()
```

Output:



Example 2: Creating a stem plot

Solution:

A stem plot plots vertical lines from a baseline to the y-coordinate and places a marker at the tip. [Ref](#)

```
Example 2

# Import the necessary libraries
import matplotlib.pyplot as plt # For plotting
import numpy as np # For numerical operations

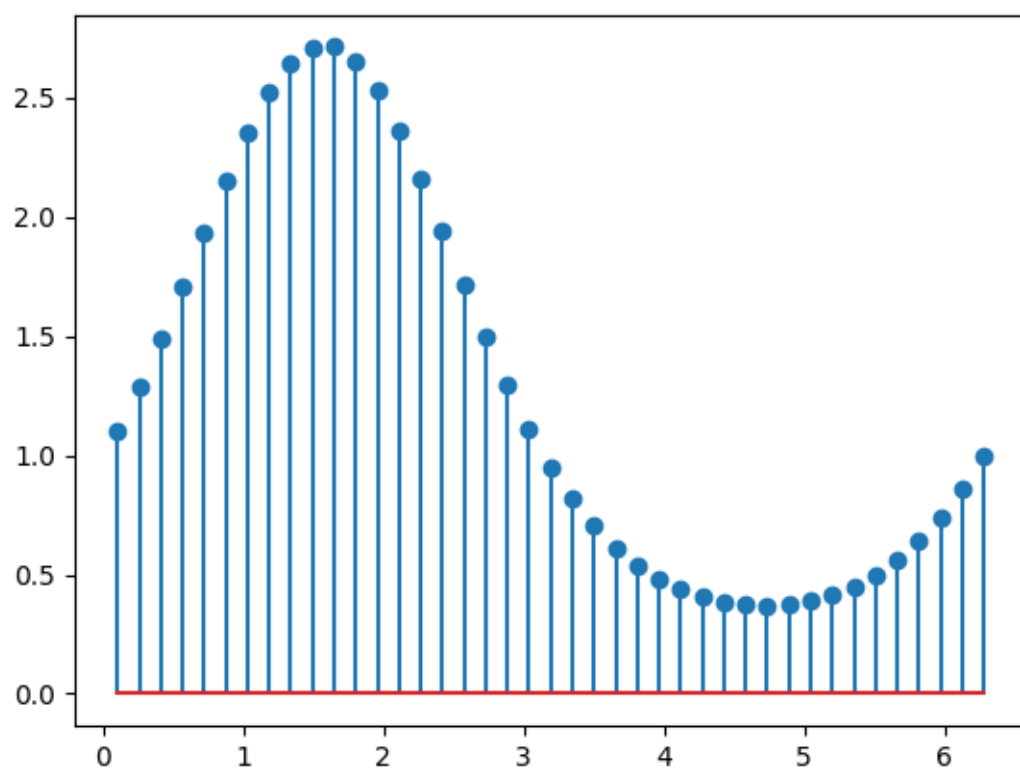
# Generate an array of 41 values evenly spaced between 0.1 and 2π
x = np.linspace(0.1, 2 * np.pi, 41)

# Compute the y-values by applying the function y = exp(sin(x)) for each x-value
y = np.exp(np.sin(x))

# Create a stem plot of the data (discrete markers with vertical lines)
plt.stem(x, y)

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

Output:



Q1. Visualize daily temperature in a specific location.

Solution:

```
Question 1

import matplotlib.pyplot as plt # For creating plots
import numpy as np # For numerical operations

# Generate an array representing the days of the month (from 1 to 30)
days = np.arange(1, 31)

# Define a list of temperatures corresponding to each day
temperatures = [68, 70, 72, 75, 77, 80, 82, 83, 81, 78, 75, 72, 71, 70, 72, 74, 77, 79,
                80, 82, 84, 86, 88, 87, 85, 82, 80, 77, 75, 73]

# Convert the temperatures list to a NumPy array for easier manipulation
temp = np.array(temperatures)

# Create a line plot of temperature vs. days
# 'marker="o"' adds circular markers at each data point
plt.plot(days, temp, marker='o')

# Add a grid to the plot for better readability
plt.grid()

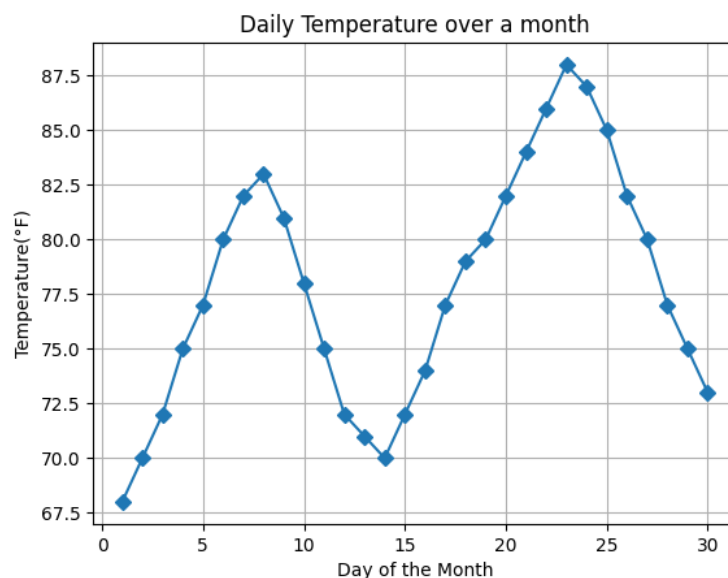
# Label the x-axis as "Day of the Month"
plt.xlabel("Day of the Month")

# Label the y-axis as "Temperature(°F)"
plt.ylabel("Temperature(°F)")

# Set the title of the plot
plt.title("Daily Temperature over a month")

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

Output:



Q2. Visualize the number of books sold in a bookstore by genre over a year.

Solution:

```
Question 2

# Import the necessary libraries
import matplotlib.pyplot as plt # For creating plots
import numpy as np # For numerical operations (not used in this case but imported)

# Define the categories (book genres) and the number of books sold for each genre
genres = ["Mystery", "Romance", "Science Fiction", "Fantasy", "Thriller"]
books_sold = [120, 90, 80, 110, 70]

# Create a bar chart
# edgecolor: The color of the borders of the bars, set to navy
# linewidth: The thickness of the border lines
# color: The fill color of the bars, set to aquamarine
barplt = plt.bar(genres, books_sold, edgecolor="navy", linewidth=1.5, color="aquamarine")

# Add labels on the bars
# bar_label: Adds the 'books_sold' values as labels inside each bar
# label_type='center': Positions the labels at the center of the bars
# color: Sets the text color of the labels to dark green
plt.bar_label(barplt, label=books_sold, color="darkgreen", label_type="center")

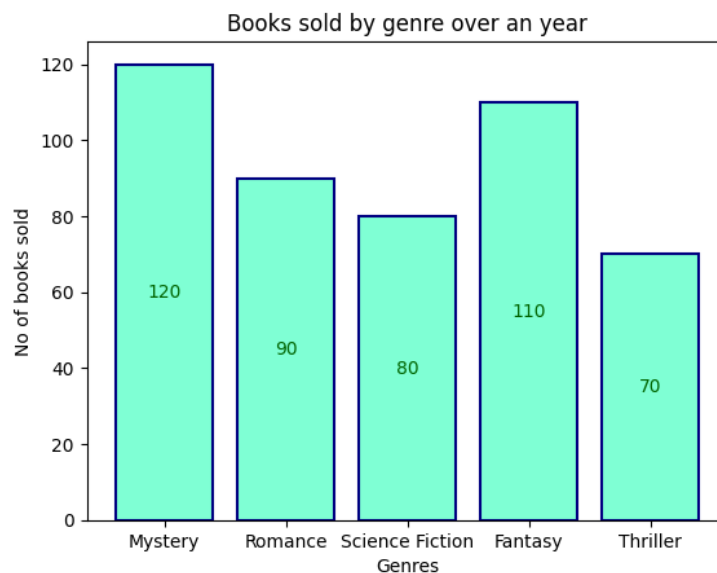
# Label the x-axis as "Genres"
plt.xlabel("Genres")

# Label the y-axis as "No of books sold"
plt.ylabel("No of books sold")

# Set the title of the chart
plt.title("Books sold by genre over an year")

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

Output:



Q3. Horizontal Bar Plot in Python over age groups and population.

Solution:

```
Question 3

import matplotlib.pyplot as plt # For creating plots

# Define the age groups and corresponding population values for each group
age_groups = ["0-10", "11-20", "21-30", "31-40", "41-50", "51-60", "61-70", "71+"]
population = [15000, 22000, 30000, 28000, 25000, 18000, 12000, 8000]

# Create a horizontal bar chart
# color: The fill color of the bars, set to "skyblue"
# linewidth: The thickness of the edge/border lines of the bars
# edgecolor: The color of the borders of the bars, set to "black"
hbar = plt.barh(age_groups, population, color="skyblue", linewidth=1.5, edgecolor="black")

# Add labels inside the horizontal bars
# labels=population: Specifies the population values to be displayed as labels
# label_type="center": Positions the labels at the center of each bar
plt.bar_label(hbar, labels=population, label_type="center")

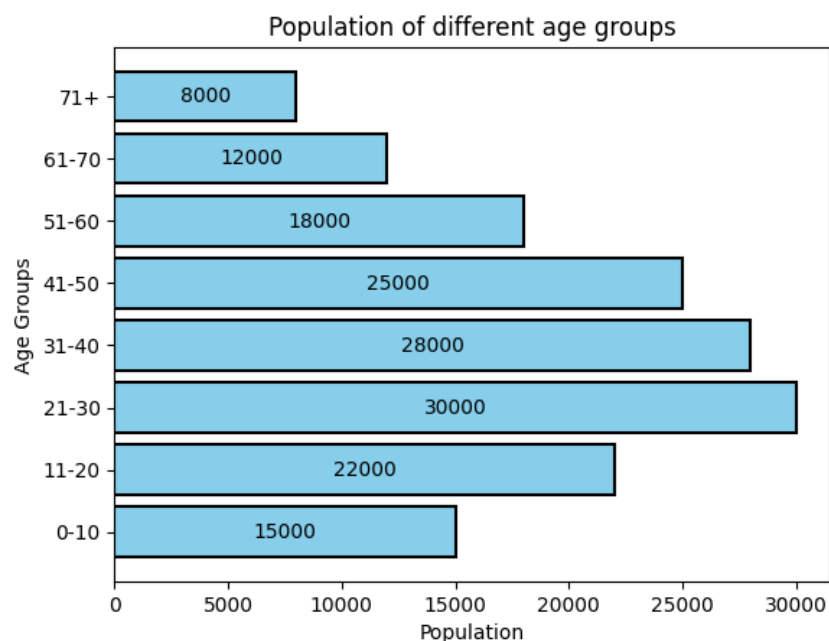
# Label the x-axis as "Age Groups"
plt.xlabel("Age Groups")

# Label the y-axis as "Population"
plt.ylabel("Population")

# Set the title of the chart
plt.title("Population of different age groups")

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

Output:



Q4. Histogram of Ages of Survey Respondents.

Solution:

```
Question 4

import matplotlib.pyplot as plt # For creating plots

# List of ages representing survey respondents
ages = [1, 1, 2, 3, 3, 5, 7, 8, 9, 10,
        10, 11, 11, 13, 13, 15, 16, 17, 18, 18,
        18, 19, 20, 21, 21, 23, 24, 24, 25, 25,
        25, 25, 26, 26, 26, 27, 27, 27, 27, 27,
        29, 30, 30, 31, 33, 34, 34, 34, 35, 36,
        36, 37, 37, 38, 38, 39, 40, 41, 41, 42,
        43, 44, 45, 45, 46, 47, 48, 48, 49, 50,
        51, 52, 53, 54, 55, 55, 56, 57, 58, 60,
        61, 63, 64, 65, 66, 68, 70, 71, 72, 74,
        75, 77, 81, 83, 84, 87, 89, 90, 90, 91]

# Define the bins for the histogram; each bin represents a range of 10 years
b = [0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100]

# Create a histogram of the age data
# ages: Data for which we want to plot the histogram
# bins: Specifies the bin edges (ranges of ages in this case)
# edgecolor: Color of the bar borders, set to navy
# linewidth: Thickness of the bar borders
values, bins, bars = plt.hist(ages, bins=b, edgecolor="navy", linewidth=1.5)

# Add labels to the bars showing the count of respondents in each bin
# bar_label: Adds the numerical labels inside the bars
# color="navy": Sets the color of the labels to navy
# fontsize=10: Sets the font size of the labels
plt.bar_label(bars, color="navy", fontsize=10)

# Label the x-axis as "Ages"
plt.xlabel("Ages")

# Label the y-axis as "Number of Respondents"
plt.ylabel("Number of Respondents")

# Set the title of the plot to "Ages of Survey Respondents"
plt.title("Ages of Survey Respondents")

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


Output:

