

## Unit10\_QB

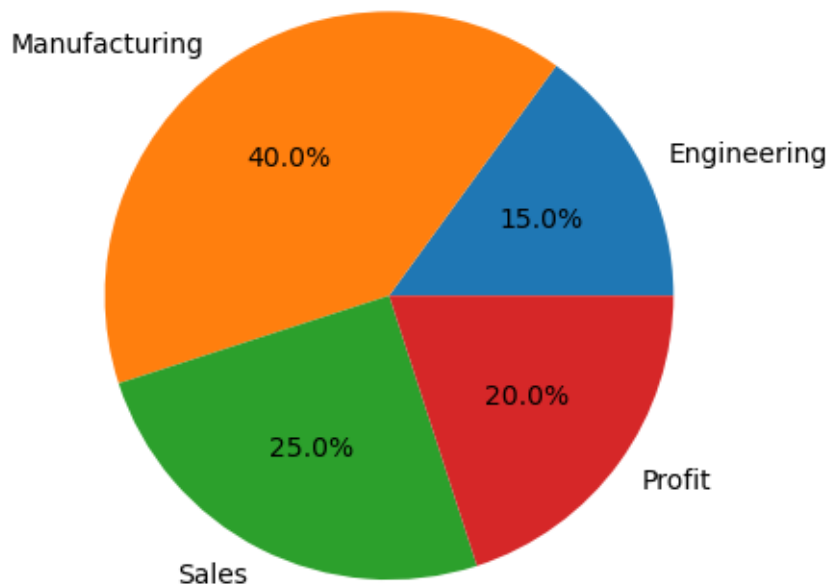
February 18, 2024

### 1 These cost categories applied to a

\$9.00 microcontroller: Engineering \$1.35 Manufacturing \$3.60 Sales \$2.25 Profit \$1.80 Create a pie chart will show the cost breakdown as different sized pieces.

```
[4]: import matplotlib.pyplot as plt
# Data
categories = ['Engineering', 'Manufacturing', 'Sales', 'Profit']
costs = [1.35, 3.60, 2.25, 1.80]
total_cost = sum(costs)
plt.pie(costs, labels=categories, autopct='%1.1f%%')
plt.title('Cost Breakdown of a $9.00 Microcontroller')
# Show the plot
plt.show()
```

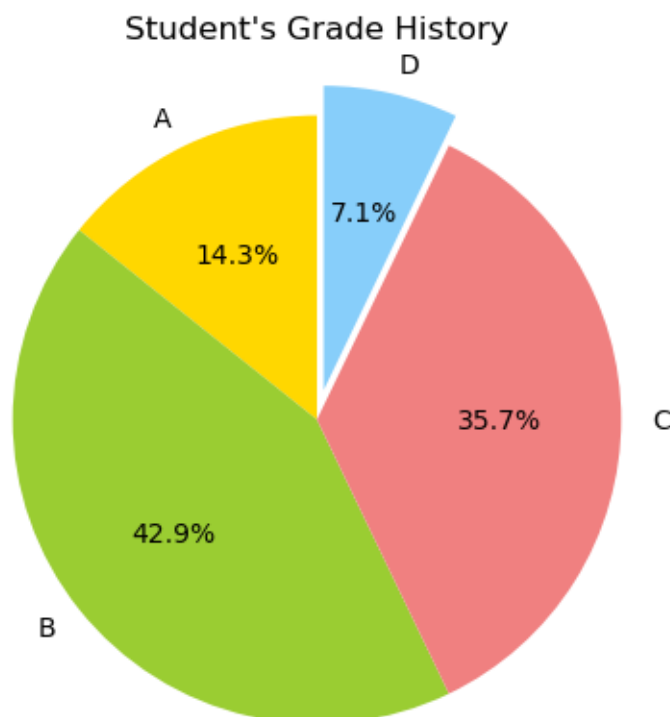
Cost Breakdown of a \$9.00 Microcontroller



## 2 Here is how many students got each grade in the recent test:

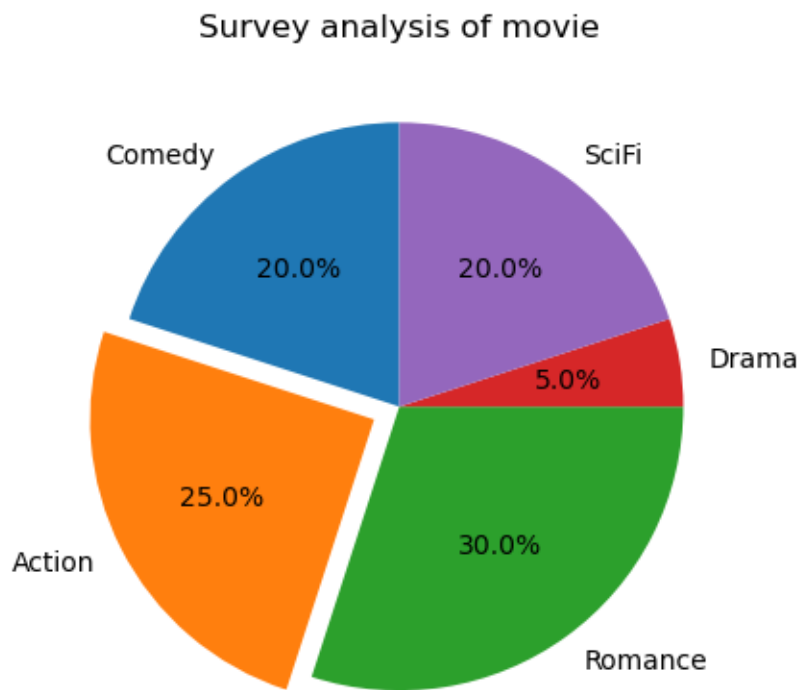
- A- 4, B-12, C-10, D-2. Plot a pie chart for the student grades in the recent chart with different colors for each student grades and create a wedge for D. Also put a chart title as student's grade history.

```
[5]: import matplotlib.pyplot as plt
# Data
grades = ['A', 'B', 'C', 'D']
counts = [4, 12, 10, 2]
colors = ['gold', 'yellowgreen', 'lightcoral', 'lightskyblue']
explode = [0, 0, 0, 0.1]
# Plot
plt.pie(counts, explode=explode, labels=grades, colors=colors, autopct='%1.1f%%', startangle=90)
# Title
plt.title("Student's Grade History")
# Show the plot
plt.axis('equal')
plt.show()
```



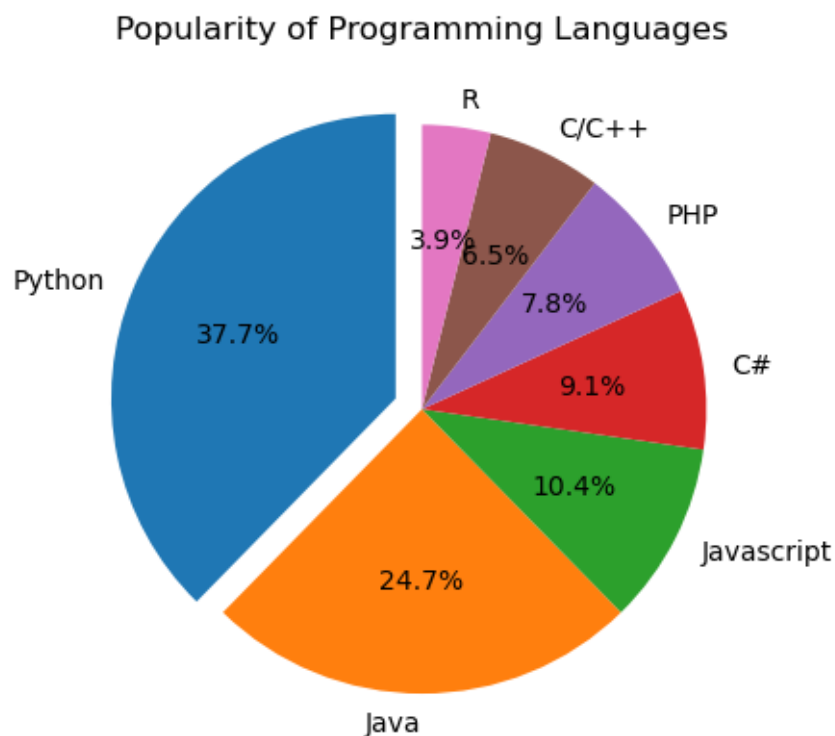
- 3 Imagine you survey your friends to find the kind of movie they like best: Comedy- 4, Action -5, Romance - 6, Drama -1, SciFi - 4. Plot a pie chart for the above survey and use different color for each analysis and create a wedge for action movies. Also put as chart title as “Survey analysis of movie”

```
[6]: import matplotlib.pyplot as plt
labels = ['Comedy', 'Action', 'Romance', 'Drama', 'SciFi']
sizes = [4, 5, 6, 1, 4]
explode = [0, 0.1, 0, 0, 0]
plt.pie(sizes, explode=explode, labels=labels, autopct='%1.1f%%', startangle=90)
plt.title("Survey analysis of movie")
plt.show()
```



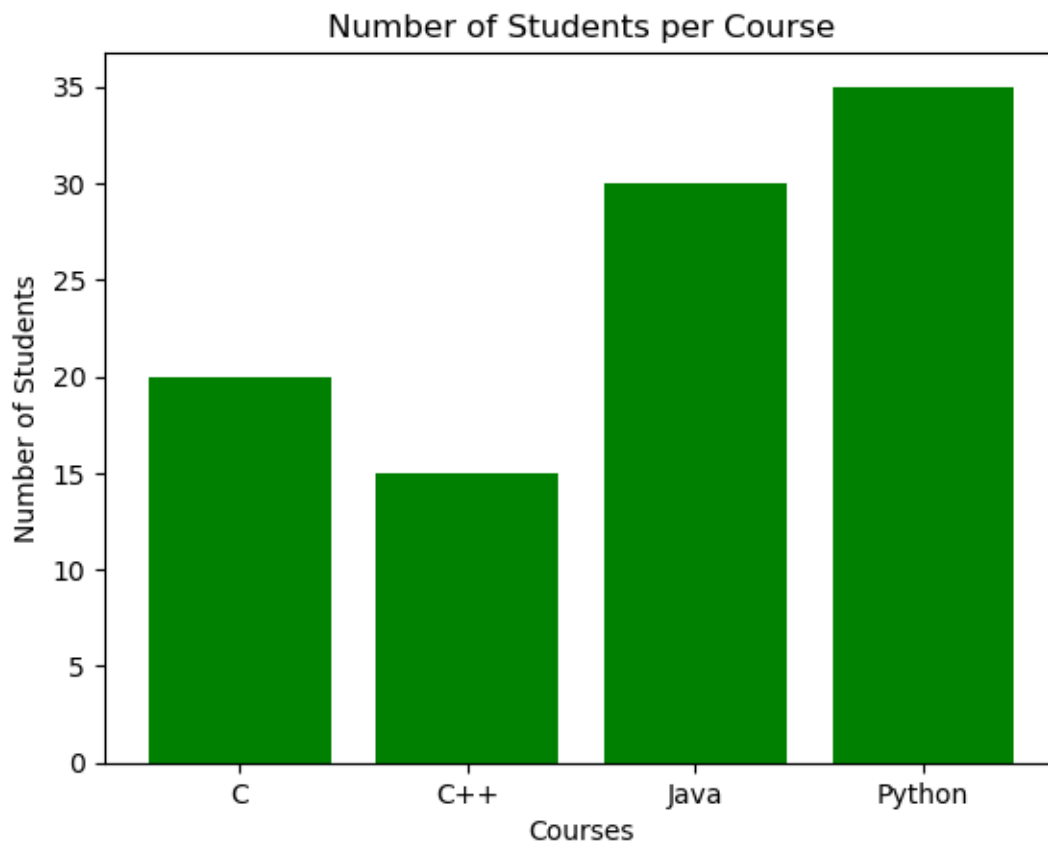
- 4 Create a Pie Chart using Python Program for the popularity data of different programming languages and displayed it as a pie chart using the Matplotlib Python library. For Python- 29, Java – 19, Javascript – 8, C# - 7, PHP – 6, C,C++ - 5, R – 3. Create an exploded view of python and show the % of each programming language in Pie Chart.

```
[7]: import matplotlib.pyplot as plt
languages = ['Python', 'Java', 'Javascript', 'C#', 'PHP', 'C/C++', 'R']
popularity = [29, 19, 8, 7, 6, 5, 3]
explode = [0.1 if lang == 'Python' else 0 for lang in languages]
plt.pie(popularity, labels=languages, explode=explode, autopct='%1.1f%%',
        ↪startangle=90)
plt.title("Popularity of Programming Languages")
plt.show()
```



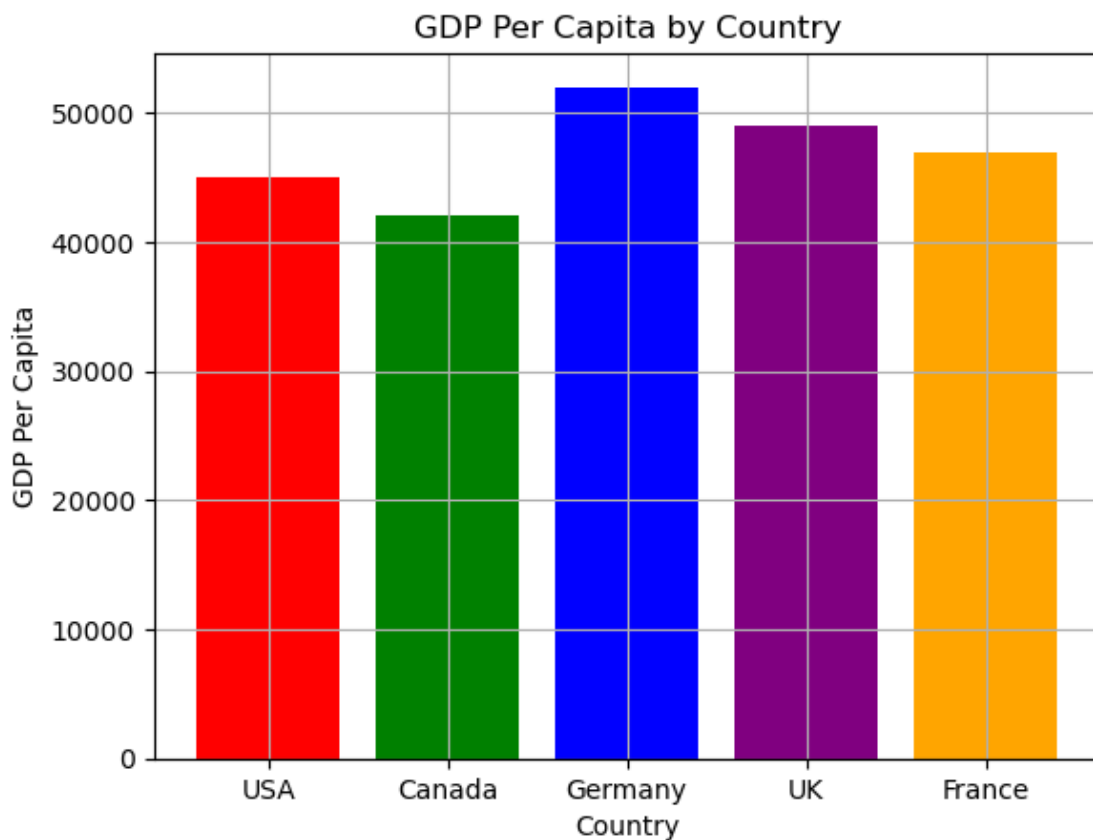
- 5 Write a python program to create a bar plot of course v/s no. of students using the following dictionary with appropriate labels for X and Y axes and colour of the bars green.  
Data={'C':20,'C++':15,'Java':30,'Python':35}

```
[8]: import matplotlib.pyplot as plt
data = {'C': 20, 'C++': 15, 'Java': 30, 'Python': 35}
# Extracting keys and values from the dictionary
courses = list(data.keys())
no_of_students = list(data.values())
# Creating a bar plot
plt.bar(courses, no_of_students, color='green')
# Adding labels for x and y axes
plt.xlabel("Courses")
plt.ylabel("Number of Students")
# Adding a title to the plot
plt.title("Number of Students per Course")
# Displaying the plot
plt.show()
```



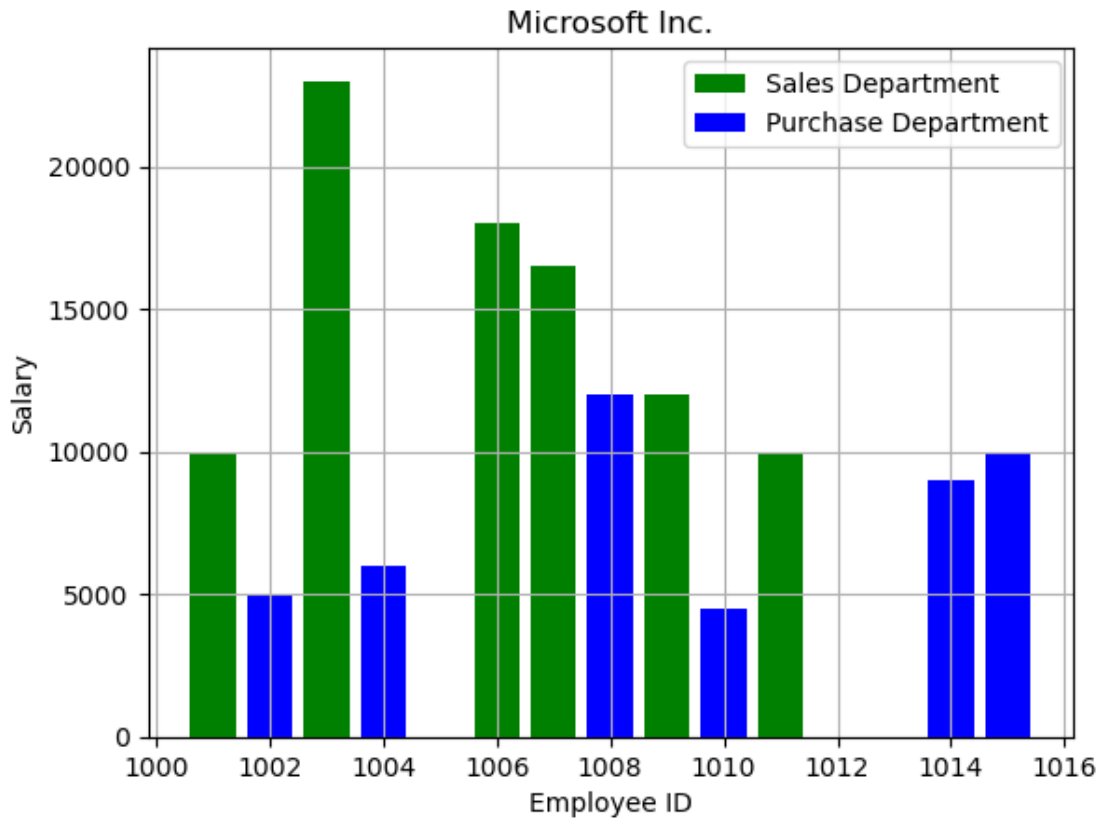
- 6 Create a bar chart for the following dataset: Country = ['USA','Canada','Germany','UK','France'] GDP\_Per\_Capita = [45000,42000,52000,49000,47000]. Also plot title, X-Axis, Y-Axis, different color for each country and the grid should be visible

```
[9]: import matplotlib.pyplot as plt
# Data
Country = ['USA', 'Canada', 'Germany', 'UK', 'France']
GDP_Per_Capita = [45000, 42000, 52000, 49000, 47000]
# Plot
plt.bar(Country, GDP_Per_Capita, color=['red', 'green', 'blue', 'purple', 'orange'])
plt.title('GDP Per Capita by Country')
plt.xlabel('Country')
plt.ylabel('GDP Per Capita')
plt.grid(True)
# Show plot
plt.show()
```



- 7 A Bar Chart to display employee id numbers on X-axis and their salaries as Y-axis in the form of a bar graph for two departments of a company. There are two departments like sales department and purchase department. For sales department their id's and salaries are mentioned as : x= [1001,1003,1006,1007,1009,1011] and y= [10000, 23000.50,18000.33,16500.5,12000.75, 9999.99] and for purchase department their id's and salaries are mentioned as:x=[1002,1004,1010,1008,1014,1015] and y=[5000,6000,4500.5,12000,9000,10000]. Make the chart title as “Microsoft Inc.”, x-axis as employee id and Y axis as Salary. Use different colors for sales and purchase department.

```
[10]: import matplotlib.pyplot as plt
# Sales Department data
sales_x = [1001, 1003, 1006, 1007, 1009, 1011]
sales_y = [10000, 23000.50, 18000.33, 16500.5, 12000.75, 9999.99]
# Purchase Department data
purchase_x = [1002, 1004, 1010, 1008, 1014, 1015]
purchase_y = [5000, 6000, 4500.5, 12000, 9000, 10000]
# Plot the data as bar charts
plt.bar(sales_x, sales_y, color='green', label='Sales Department')
plt.bar(purchase_x, purchase_y, color='blue', label='Purchase Department')
# Set chart title and axis labels
plt.title('Microsoft Inc.')
plt.xlabel('Employee ID')
plt.ylabel('Salary')
# Show grid
plt.grid(True)
# Add a legend
plt.legend()
# Show the chart
plt.show()
```



- 8 Write a program to build two bar graphs using subplot function for given two dictionaries in which one graph is in 1st row and another in second row which is horizontal representation of bar graph. D1={"aryan":66,"bob":70,"jack":66,"seema":34} D2={"joy":45,"sid":85,"hina":90} And also make a title of graph at top as "BAR PLOT".

```
[16]: import matplotlib.pyplot as plt
# Input data
D1 = {"aryan": 66, "bob": 70, "jack": 66, "seema": 34}
D2 = {"joy": 45, "sid": 85, "hina": 90}
# Set up the plot

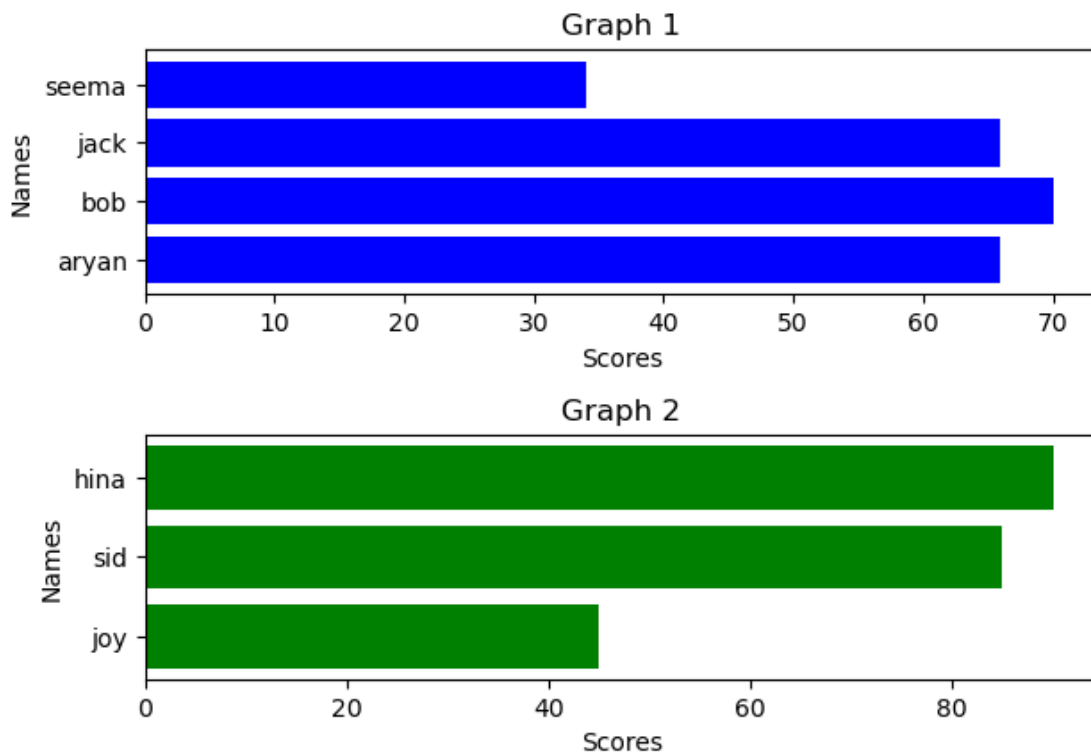
plt.suptitle("BAR PLOT", fontsize=14)
# Plot the first graph
plt.subplot(2,1,1)
plt.barh(list(D1.keys()), list(D1.values()), color='blue')
plt.xlabel("Scores")
```



```
plt.ylabel("Names")
plt.title("Graph 1")

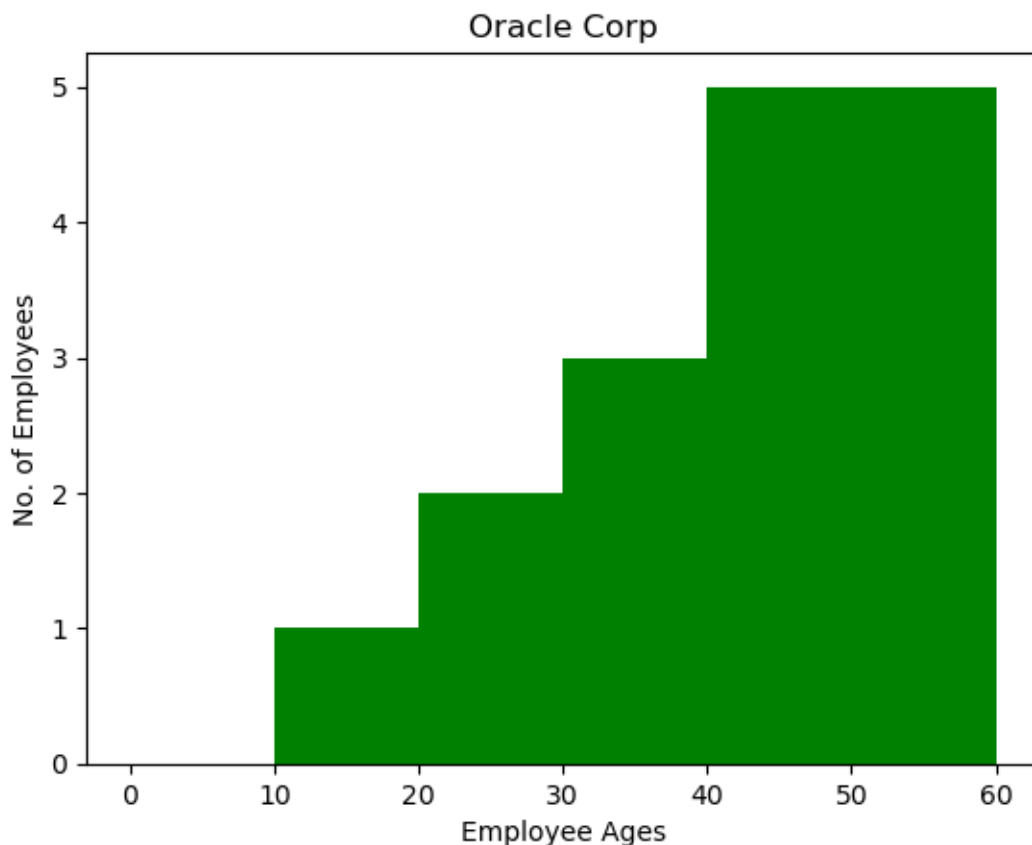
# Plot the second graph
plt.subplot(2,1,2)
plt.barh(list(D2.keys()), list(D2.values()), color='green')
plt.xlabel("Scores")
plt.ylabel("Names")
plt.title("Graph 2")
plt.tight_layout()
plt.show()
```

## BAR PLOT



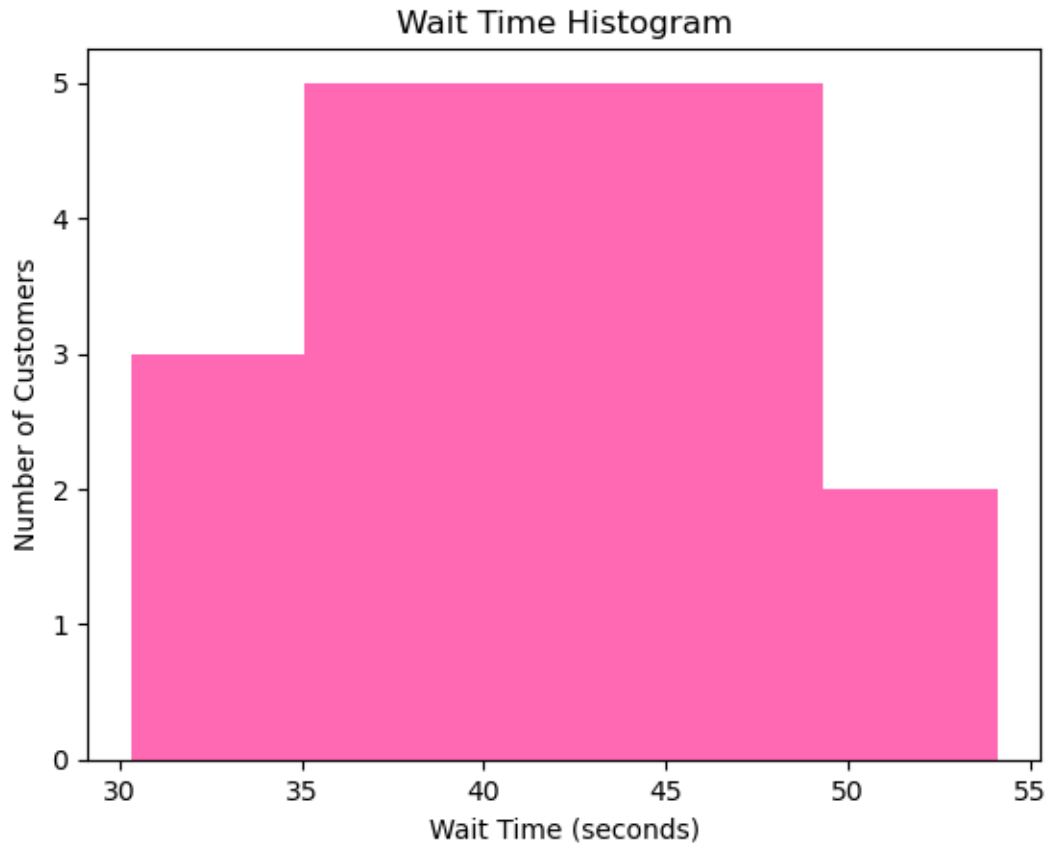
- 9 A program to display a histogram showing the number of employees in specific age groups. The data is shown: emp\_ages=[22,45,30,59,58,56,57,45,43,43,50,40,34,33,25,19] and their bins are [0,10,20,30,40,50,60]. Create a histogram with x-axis label as “employee ages” and y axis label as ” no. of employees”. Create a title of the plot as “Oracle Corp”. Also tehcolor of histogram created should be cyan.

```
[18]: import matplotlib.pyplot as plt
emp_ages = [22,45,30,59,58,56,57,45,43,43,50,40,34,33,25,19]
bins = [0,10,20,30,40,50,60]
plt.hist(emp_ages, bins=bins, color='green')
plt.xlabel('Employee Ages')
plt.ylabel('No. of Employees')
plt.title('Oracle Corp')
plt.show()
```



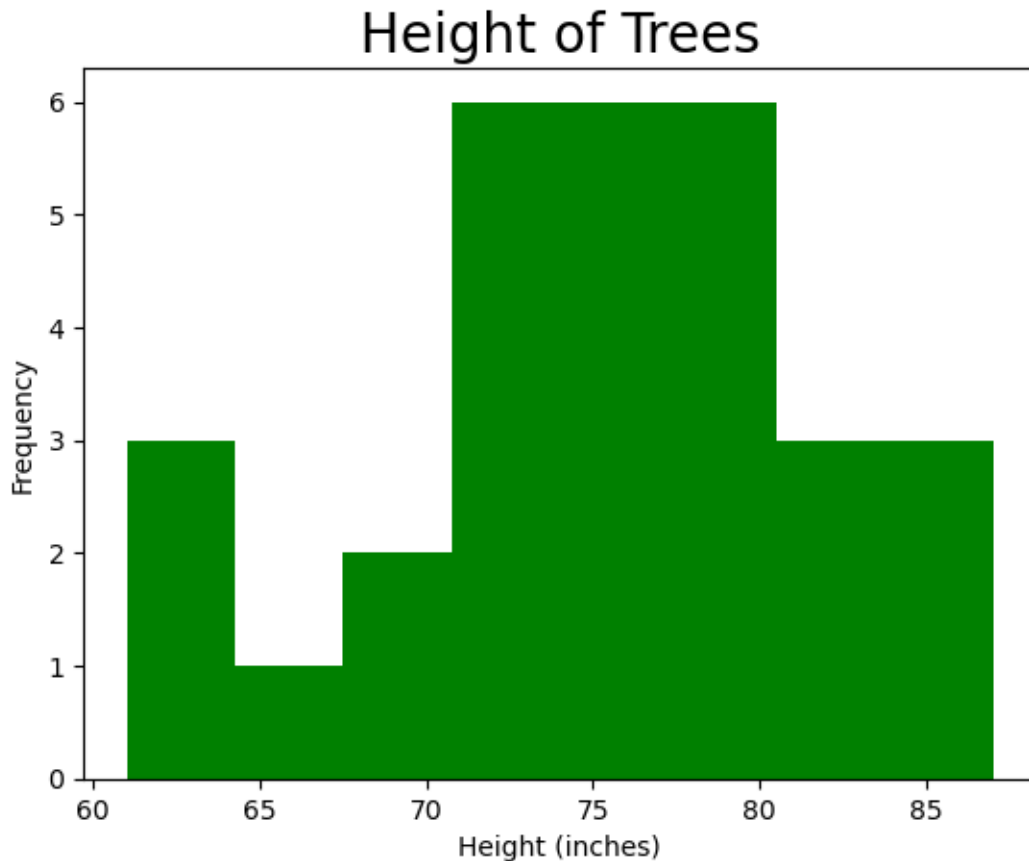
- 10 Jeff is a branch manager at a local bank. Recently Jeff's is receiving customer feedback saying that the wait times for a client to be served by a customer service representative are too long. Jeff decides to observe and write down the time spent by each customer on waiting. Here are his findings from observing and writing down the wait times spent by 20 customers (in seconds): 43.1, 35.6, 37.6, 36.5, 45.3, 43.5, 40.3, 50.2, 47.3, 31.2, 42.2, 45.5, 30.3, 31.4, 35.6, 45.2, 54.1, 45.6, 36.5, 43.1. Plot a histogram for the above data. Calculate bins and bins width and plot with calculated bins.

```
[20]: import matplotlib.pyplot as plt
# Wait time data
wait_times = [43.1, 35.6, 37.6, 36.5, 45.3, 43.5, 40.3, 50.2, 47.3, 31.2,
              42.2, 45.5, 30.3, 31.4, 35.6, 45.2, 54.1, 45.6, 36.5, 43.1]
# Calculate bins and bin width
num_bins = 5
bin_width = (max(wait_times) - min(wait_times)) / num_bins
# Create histogram
plt.hist(wait_times, bins=num_bins, color='hotpink')
# Set plot title and axis labels
plt.title("Wait Time Histogram")
plt.xlabel("Wait Time (seconds)")
plt.ylabel("Number of Customers")
# Show plot
plt.show()
```



- 11 Uncle Bruno owns a garden with 30 black cherry trees. Each tree is of a different height. The height of the trees (in inches): 61, 63, 64, 66, 68, 69, 71, 71.5, 72, 72.5, 73, 73.5, 74, 74.5, 76, 76.2, 76.5, 77, 77.5, 78, 78.5, 79, 79.2, 80, 81, 82, 83, 84, 85, 87. Plot a histogram with color green, title of the chart should be height of trees along with it's fontsize as 20.

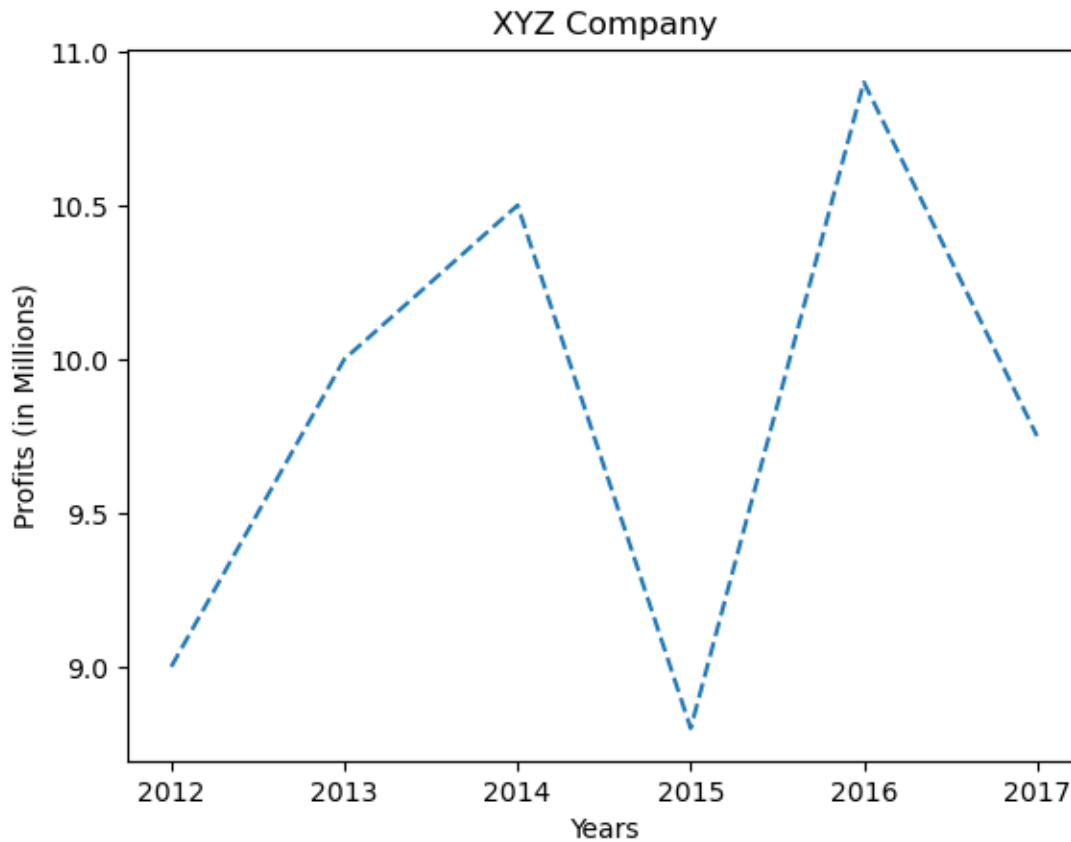
```
[21]: import matplotlib.pyplot as plt
tree_heights = [61, 63, 64, 66, 68, 69, 71, 71.5, 72, 72.5, 73, 73.5, 74,
               74.5, 76, 76.2, 76.5, 77, 77.5, 78, 78.5, 79, 79.2, 80, 81, 82, 83,
               84, 85, 87]
plt.hist(tree_heights, bins=8, color='green')
plt.title('Height of Trees', fontsize=20)
plt.xlabel('Height (inches)')
plt.ylabel('Frequency')
plt.show()
```



- 12 A Program to create a line graph to show the profits of a company in various years. The data is as mentioned: x axis as years and y axis as profits (in Millions). X=[2012,2013,2014,2015,2016,2017] and y = [9,10,10.5,8.8,10.9,9.75]. Plot a line chart with x axis as “Years” and y axis as “Profits (in Millions)” and title of the line chart as “XYZ Company”. Also the linestyle should be dashed one.

```
[22]: import matplotlib.pyplot as plt
# data
x = [2012, 2013, 2014, 2015, 2016, 2017]
y = [9, 10, 10.5, 8.8, 10.9, 9.75]
# plot line graph
plt.plot(x, y, linestyle='--')
# set title and axis labels
```

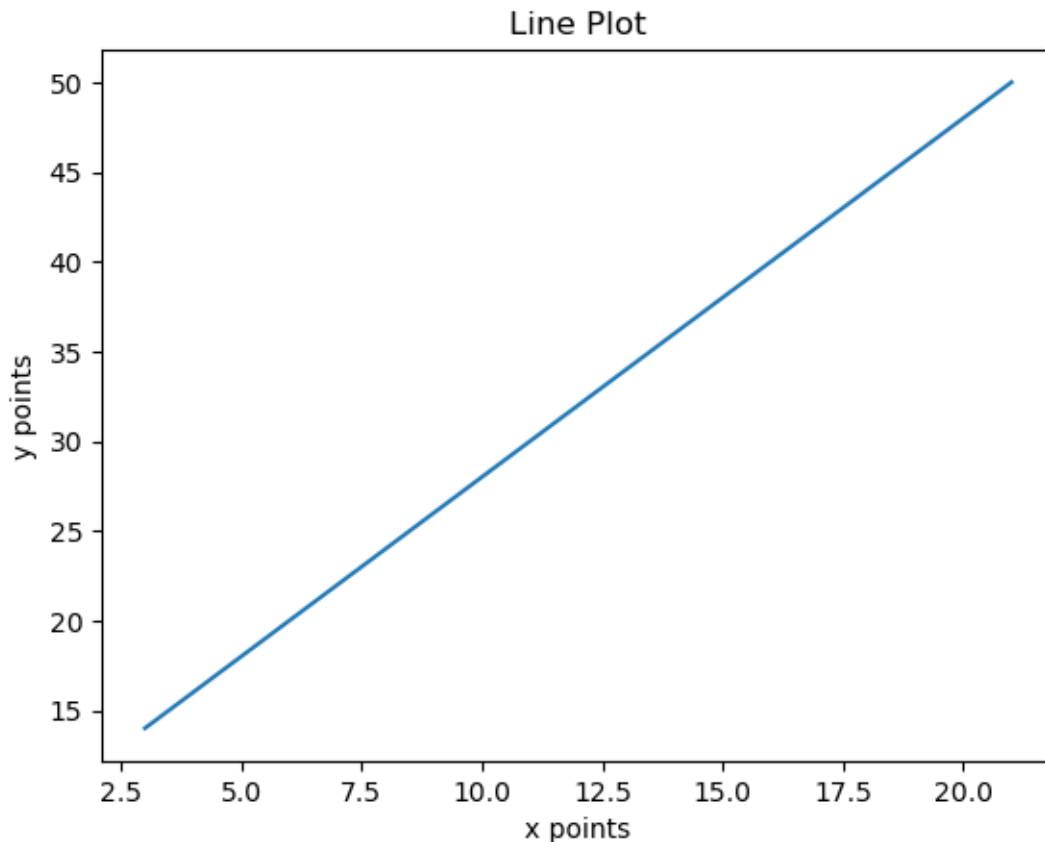
```
plt.title('XYZ Company')
plt.xlabel('Years')
plt.ylabel('Profits (in Millions)')
# show the plot
plt.show()
```



- 13 Create a Line Plot in which x points varies from 3 to 21. And the y points varies according to the equation  $y=2x+8$ .

```
[23]: import matplotlib.pyplot as plt
# x points varying from 3 to 21
x = range(3, 22)
# y points calculated using the equation y=2x+8
y = [2*i+8 for i in x]
# create a line plot
plt.plot(x, y)
# add labels for x and y axis
plt.xlabel('x points')
```

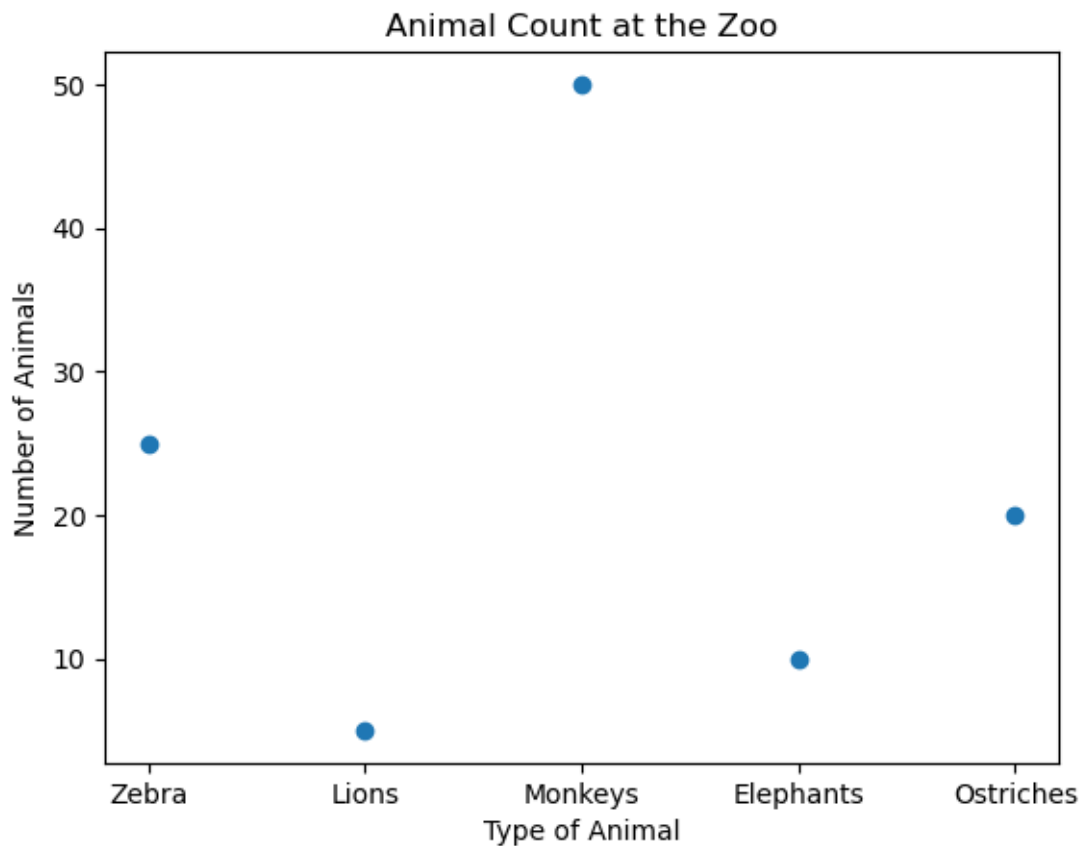
```
plt.ylabel('y points')
# add a title for the plot
plt.title('Line Plot')
# show the plot
plt.show()
```



- 14 Laurell had visited a zoo recently and had collected the following data. How can Laurell use a scatter plot to represent this data? .Take Type of Animal as X-axis and no. in Y axis. The data is as follows: Zebra-25, Lions- 5, Monkeys- 50, Elephants -10, Ostriches - 20

```
[24]: import matplotlib.pyplot as plt
animals = ['Zebra', 'Lions', 'Monkeys', 'Elephants', 'Ostriches']
no_of_animals = [25, 5, 50, 10, 20]
plt.scatter(animals, no_of_animals)
plt.title('Animal Count at the Zoo')
```

```
plt.xlabel('Type of Animal')
plt.ylabel('Number of Animals')
plt.show()
```

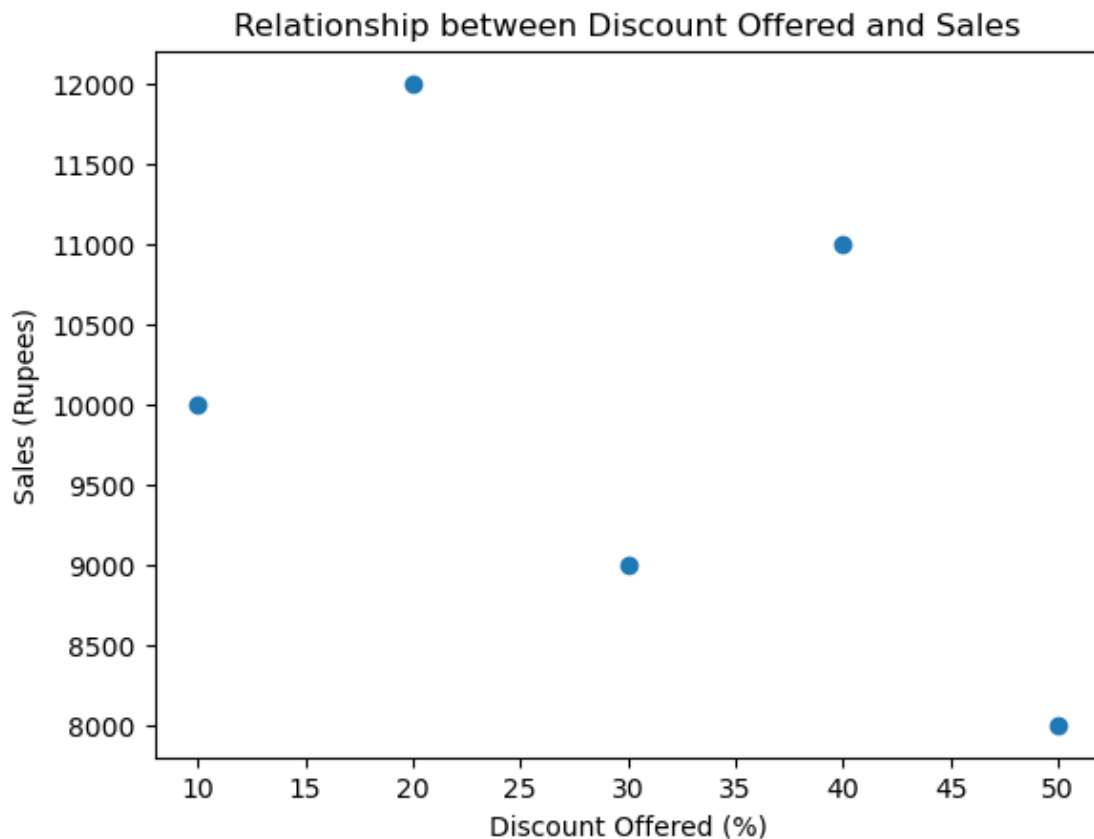


- 15 Prayatna sells designer bags and wallets. During the sales season, he gave discounts ranging from 10% to 50% over a period of 5 weeks. He recorded his sales for each type of discount in an array. Draw a scatter plot to show a relationship between the discount offered and sales made. Take 5 sales in Rupees as user defined values

```
[25]: import matplotlib.pyplot as plt
# user defined sales data for 5 weeks in Rupees
sales = [10000, 12000, 9000, 11000, 8000]
# discounts offered in percentage
discounts = [10, 20, 30, 40, 50]
# create scatter plot
```



```
plt.scatter(discounts, sales)
# set x and y labels and title
plt.xlabel('Discount Offered (%)')
plt.ylabel('Sales (Rupees)')
plt.title('Relationship between Discount Offered and Sales')
# display the plot
plt.show()
```



- 15.0.1 Plot a subplot showing the marks of 5 students for 6 subjects (Digital Electronics, Probability and Stochastics, Python, Full Stack Development, IELTS (Reading), and Data Structure). All the marks for each subject and for each student is to be taken user defined. Subplot should be prepared for each subject. Each Subplot should have a title of subject along with a main title of a plot.

```
[1]: import matplotlib.pyplot as plt

# Marks data (sample data)
subjects = ['Digital Electronics', 'Probability and Stochastics', 'Python', 'Full Stack Development', 'IELTS (Reading)', 'Data Structure']
```

```

marks = [
    [90, 78, 85, 92, 88],
    [70, 82, 75, 88, 95],
    [85, 90, 92, 78, 80],
    [78, 85, 80, 92, 88],
    [95, 92, 88, 80, 85],
    [80, 75, 85, 92, 78]
]

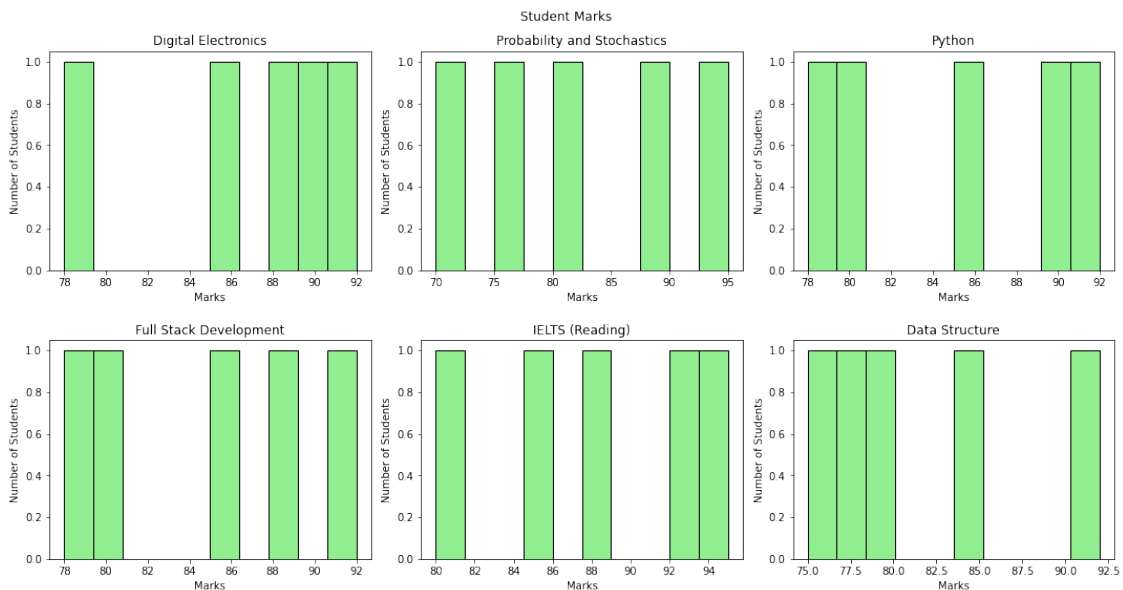
# Create subplots
fig, axs = plt.subplots(2, 3, figsize=(15, 8))
fig.suptitle('Student Marks')

# Plot for each subject
for i in range(len(subjects)):
    row, col = divmod(i, 3)
    axs[row, col].hist(marks[i], bins=10, color='lightgreen', edgecolor='black')
    axs[row, col].set_title(subjects[i])
    axs[row, col].set_xlabel('Marks')
    axs[row, col].set_ylabel('Number of Students')

# Display the subplots
plt.tight_layout(h_pad=2.0) # Increase the vertical spacing between the plots

plt.show()

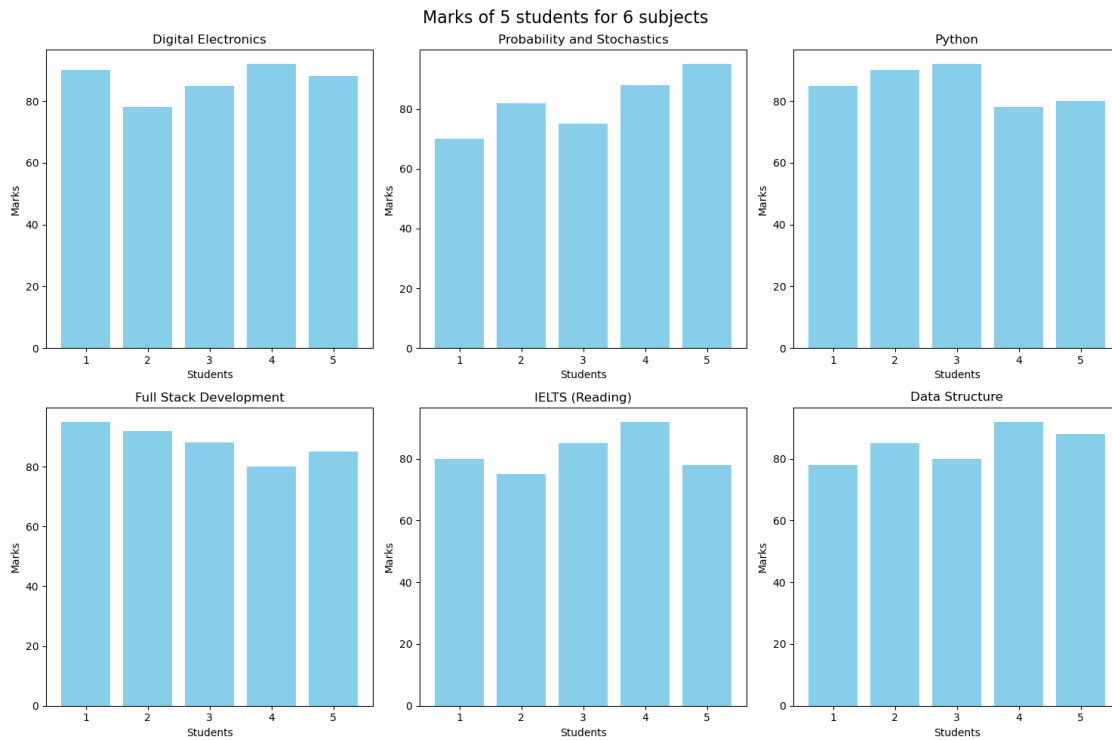
```



```
[26]: import matplotlib.pyplot as plt
# Function to get marks for a subject from the user
def get_marks(subject):
    marks = []
    for i in range(1, 6):
        marks.append(float(input(f"Enter marks of Student {i} for {subject}:"))
    return marks
# Subjects
subjects = ["Digital Electronics", "Probability and Stochastics", "Python",
    "Full Stack Development", "IELTS (Reading)", "Data Structure"]
# Marks for each subject
marks_data = {}
for subject in subjects:
    marks_data[subject] = get_marks(subject)
# Plotting subplots
plt.figure(figsize=(15, 10))
plt.suptitle('Marks of 5 students for 6 subjects', fontsize=16)
for i in range(6):
    subject = subjects[i]
    marks = marks_data[subject]
    plt.subplot(2, 3, i+1)
    plt.bar(range(1, 6), marks, color='skyblue')
    plt.title(subject)
    plt.xlabel('Students')
    plt.ylabel('Marks')
plt.tight_layout()
plt.show()
```

```
Enter marks of Student 1 for Digital Electronics: 90
Enter marks of Student 2 for Digital Electronics: 78
Enter marks of Student 3 for Digital Electronics: 85
Enter marks of Student 4 for Digital Electronics: 92
Enter marks of Student 5 for Digital Electronics: 88
Enter marks of Student 1 for Probability and Stochastics: 70
Enter marks of Student 2 for Probability and Stochastics: 82
Enter marks of Student 3 for Probability and Stochastics: 75
Enter marks of Student 4 for Probability and Stochastics: 88
Enter marks of Student 5 for Probability and Stochastics: 95
Enter marks of Student 1 for Python: 85
Enter marks of Student 2 for Python: 90
Enter marks of Student 3 for Python: 92
Enter marks of Student 4 for Python: 78
Enter marks of Student 5 for Python: 80
Enter marks of Student 1 for Full Stack Development: 95
Enter marks of Student 2 for Full Stack Development: 92
Enter marks of Student 3 for Full Stack Development: 88
Enter marks of Student 4 for Full Stack Development: 80
```

Enter marks of Student 5 for Full Stack Development: 85  
 Enter marks of Student 1 for IELTS (Reading): 80  
 Enter marks of Student 2 for IELTS (Reading): 75  
 Enter marks of Student 3 for IELTS (Reading): 85  
 Enter marks of Student 4 for IELTS (Reading): 92  
 Enter marks of Student 5 for IELTS (Reading): 78  
 Enter marks of Student 1 for Data Structure: 78  
 Enter marks of Student 2 for Data Structure: 85  
 Enter marks of Student 3 for Data Structure: 80  
 Enter marks of Student 4 for Data Structure: 92  
 Enter marks of Student 5 for Data Structure: 88



## 16 Write a Python program to draw a scatter plot comparing two subject marks of Mathematics and Science. Use marks of 10 students.

Test Data:

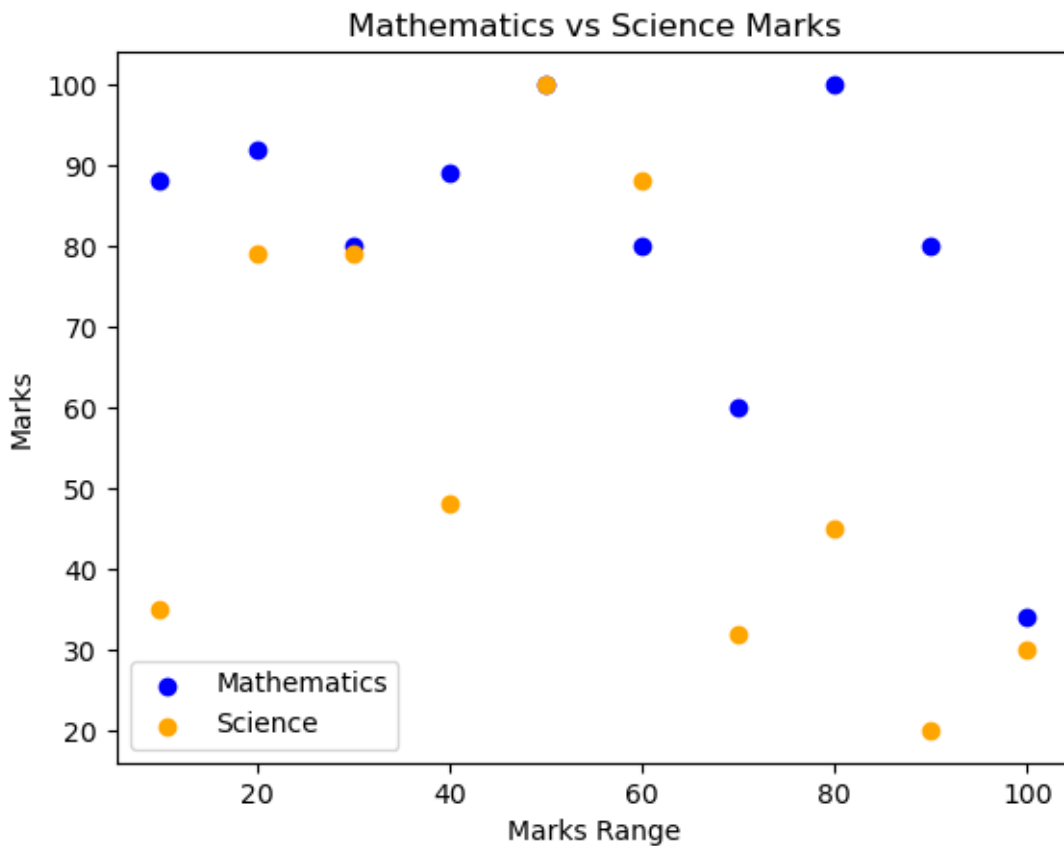
```
math_marks = [88, 92, 80, 89, 100, 80, 60, 100, 80, 34]
```

```
science_marks = [35, 79, 79, 48, 100, 88, 32, 45, 20, 30]
```

```
marks_range = [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
```

Add appropriate labels, title and legend.

```
[27]: import matplotlib.pyplot as plt
# Data
math_marks = [88, 92, 80, 89, 100, 80, 60, 100, 80, 34]
science_marks = [35, 79, 79, 48, 100, 88, 32, 45, 20, 30]
marks_range = [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
# Create a scatter plot
plt.scatter(marks_range, math_marks, label='Mathematics', color='blue')
plt.scatter(marks_range, science_marks, label='Science', color='orange')
# Add labels and title
plt.xlabel('Marks Range')
plt.ylabel('Marks')
plt.title('Mathematics vs Science Marks')
plt.legend()
# Display the scatter plot
plt.show()
```



- 17 Draw multiple plots in one figure using subplot function. The multiple plots include below according to order: Plot a scatter plot with following data:  $x = [5, 7, 8, 7, 2, 17, 2, 9, 4, 11, 12, 9, 6]$   $y = [99, 86, 87, 88, 111, 86, 103, 87, 94, 78, 77, 85, 86]$  The x axis represents the age of car while y axis represents the speed of car. The title of the graph should be age v/s speed of car. Also in graph there should be x and y labels. The marker used should be star. The marker color should be green. The marker size should be 60. (Entire Scatter plot 2 marks) Plot a horizontal bar with following data:  $x = ["A", "B", "C", "D"]$   $y = [3, 8, 1, 10]$  The x axis represents the name of car while y axis represents the selling of car. The title of the graph should be name v/s selling of car. Also in graph there should be x and y label. The horizontal bar chart's height should be 0.1. The color of bar should be yellow. (Entire bar plot 2 marks) Plot a histogram with following data:  $data = [1, 3, 3, 3, 3, 9, 9, 5, 4, 4, 8, 8, 8, 6, 7]$   $bins = 4$ , the title of the graph should be histogram of cars. The orientation should be vertical. The color of plot should be violet (Entire histogram plot 2 marks) Plot a pie with following data:  $y = [35, 25, 25, 15]$   $mylabels = ['Apple', 'Bananas', 'Cherries', 'Dates']$  The title of the graph should be pie chart. The exploded view should be shown with 0.2 value for 'Apple' (Entire pie chart of 2 marks) Also need to provide a superior title to the subplot prepared i.e 'My Subplot for cars' (0.5 marks) and subplot preparation (0.5 mark) For clear visualization can use the following syntax after importing matplotlib `plt.figure(figsize=(10,10))`

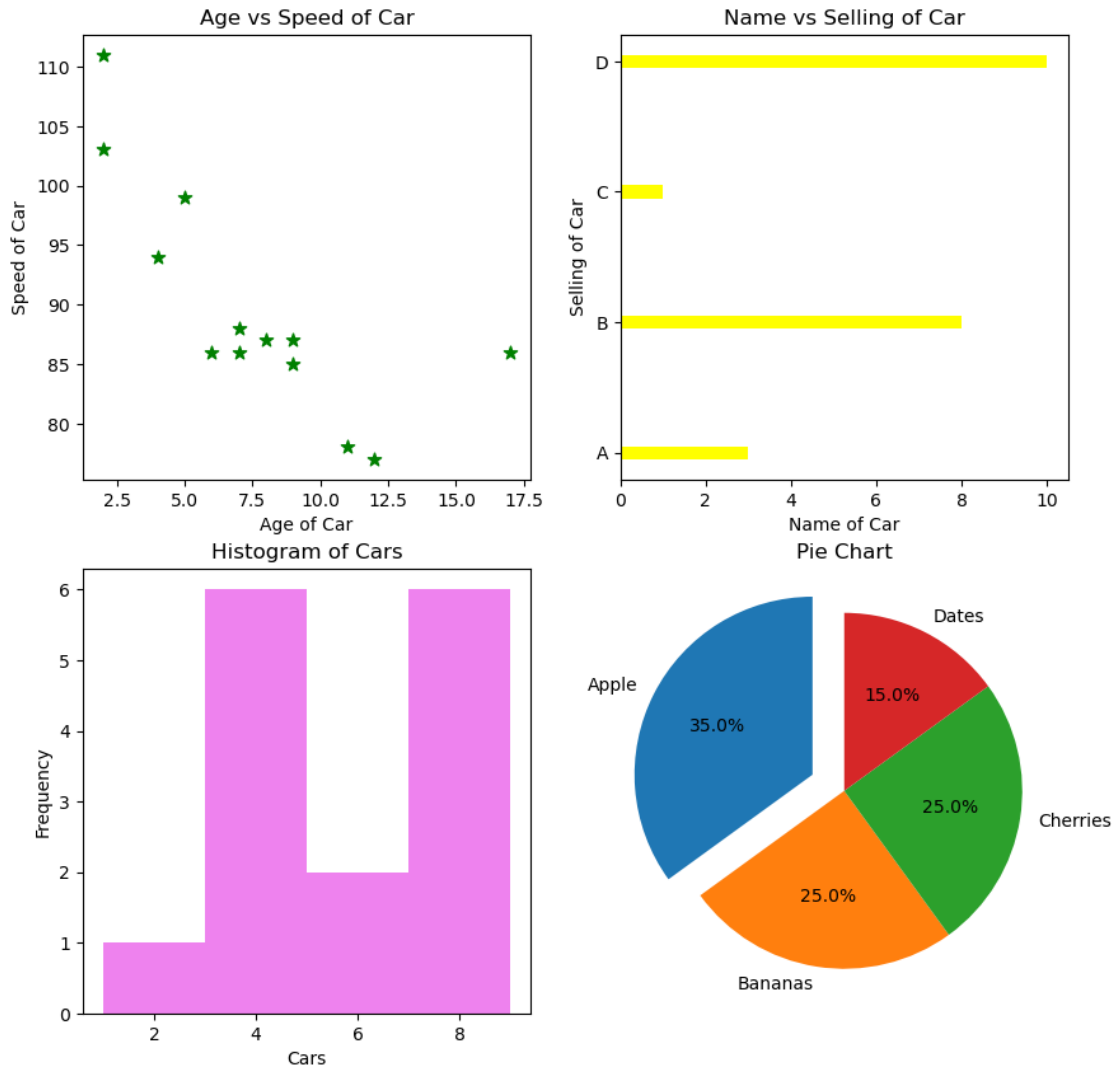
```
[28]: import matplotlib.pyplot as plt
# Data for scatter plot
x_scatter = [5, 7, 8, 7, 2, 17, 2, 9, 4, 11, 12, 9, 6]
y_scatter = [99, 86, 87, 88, 111, 86, 103, 87, 94, 78, 77, 85, 86]
# Data for horizontal bar plot
x_bar = ["A", "B", "C", "D"]
y_bar = [3, 8, 1, 10]
# Data for histogram
data_hist = [1, 3, 3, 3, 3, 9, 9, 5, 4, 4, 8, 8, 8, 6, 7]
bins_hist = 4
# Data for pie chart
```

```

y_pie = [35, 25, 25, 15]
labels_pie = ['Apple', 'Bananas', 'Cherries', 'Dates']
# Subplot preparation
plt.figure(figsize=(10, 10))
# Subplot 1: Scatter Plot
plt.subplot(2, 2, 1)
plt.scatter(x_scatter, y_scatter, marker='*', color='green', s=60)
plt.title('Age vs Speed of Car')
plt.xlabel('Age of Car')
plt.ylabel('Speed of Car')
# Subplot 2: Horizontal Bar Plot
plt.subplot(2, 2, 2)
plt.barh(x_bar, y_bar, height=0.1, color='yellow')
plt.title('Name vs Selling of Car')
plt.xlabel('Name of Car')
plt.ylabel('Selling of Car')
# Subplot 3: Histogram
plt.subplot(2, 2, 3)
plt.hist(data_hist, bins=bins_hist, orientation='vertical', color='violet')
plt.title('Histogram of Cars')
plt.xlabel('Cars')
plt.ylabel('Frequency')
# Subplot 4: Pie Chart
plt.subplot(2, 2, 4)
plt.pie(y_pie, labels=labels_pie, autopct='%1.1f%%', explode=[0.2, 0, 0, 0],
        ↪startangle=90)
plt.title('Pie Chart')
# Superior Title
plt.suptitle('My Subplot for Cars')
# Show the subplot
plt.show()

```

My Subplot for Cars



18 There is an array of scores of 5 Batsmen in 4 T20 Matches. Which is given below.

- Scores= [[13, 10, 9, 33], [63, 46, 90, 42], [39, 76, 13, 29], [82, 9, 29, 78], [67, 61, 59, 36]] Further you are asked to perform below tasks.
- (i). Add scores of every batsman of 5th Match given below in the same array and print the array. Match\_6= [41, 87, 72, 36, 92]
- (ii). Add two new batsmen's scores in respective 5 T20 Matches in the array created in task (i) above and print the array.
- Batsman\_6= [77, 83, 98, 95, 89] Batsman\_7= [92, 71, 52, 61, 53]



- (iii). Add extra column with sum of all 5 T20 Matches' scores of each batsman in the array created in task (ii) and print the final array. Note: Use Numpy module for all the Arrays given above. "Using the final array created in task(iii) above, generate graphs mentioned below:
- (a). Make a line chart of Total Scores of each batsman which is stored in last column of final array v/s No. of Batsman. Use dashed line in graph, with black color. Give label on x-axis as "No. of Batsman" and label on y-axis as "Scores". Give title to the chart as "Leader Board" with bold fonts.
- (b). Make one Bar chart of scores of Batsman\_1 and Batsman\_2 for all 5 T20 matches. Give color for bars of Batsman\_1 as Purple and for Batsman\_2 Dark red. Also show required legend in bar chart.
- (c). Make a pie chart of Total Scores of each batsman which is stored in last column of final array. Show the pie chart with exploded view of all pieces with 0.1 amount. Also display percentage in the pie chart. Also show required legend for pie chart. Note: Passing the values using numpy Array Slicing from array created in task(iii) for creating graphs above is compulsory."

```
[4]: import numpy as np
import matplotlib.pyplot as plt
# Task (i)
Scores = np.array([[13, 10, 9, 33],
                   [63, 46, 90, 42],
                   [39, 76, 13, 29],
                   [82, 9, 29, 78],
                   [67, 61, 59, 36]])

Match_6 = np.array([41, 87, 72, 36, 92]).reshape(5,1)
# Append Match_6 as a new column to Scores
Scores = np.concatenate((Scores, Match_6), axis=1) # Append along columns
↳(axis=1)

print("Task (i):")
print(Scores)
print()

# Task (ii)
Batsman_6 = np.array([77, 83, 98, 95, 89])
Batsman_7 = np.array([92, 71, 52, 61, 53])

Scores = np.concatenate((Scores, Batsman_6.reshape(-1, 1), Batsman_7.
↳reshape(-1, 1)), axis=1)

print("Task (ii):")
print(Scores)
print()
```

```

# Task (iii)
Total_Scores = np.sum(Scores[:, :-2], axis=1)
Scores = np.concatenate((Scores, Total_Scores.reshape(-1, 1)), axis=1)

print("Task (iii):")
print(Scores)
print()

# Graphs
# (a) Line chart
batsman_numbers = np.arange(1, Scores.shape[0] + 1) # Adjusted for the number
↳ of batsmen
total_scores = Scores[:, -1]

plt.figure(figsize=(10, 6))
plt.plot(batsman_numbers, total_scores, linestyle='dashed', color='black')
plt.xlabel('No. of Batsman')
plt.ylabel('Scores')
plt.title('Leader Board', fontweight='bold')
plt.show()

# (b) Bar chart
batsman_1_scores = Scores[:, -4]
batsman_2_scores = Scores[:, -3]

plt.figure(figsize=(10, 6))
plt.bar(batsman_numbers - 0.2, batsman_1_scores, width=0.4, color='purple',
↳ label='Batsman_1')
plt.bar(batsman_numbers + 0.2, batsman_2_scores, width=0.4, color='darkred',
↳ label='Batsman_2')
plt.xlabel('T20 Matches')
plt.ylabel('Scores')
plt.legend()
plt.show()

# (c) Pie chart
labels = [f'Batsman_{i}' for i in range(1, Scores.shape[1] - 2)] # Adjusted
↳ for the number of batsmen
explode = [0.1] * len(labels)

plt.figure(figsize=(8, 8))
plt.pie(total_scores, labels=labels, explode=explode, autopct='%1.1f%%',
↳ startangle=90)
plt.title('Total Scores Distribution')
plt.legend(labels, loc='upper left')
plt.show()

```

Task (i):

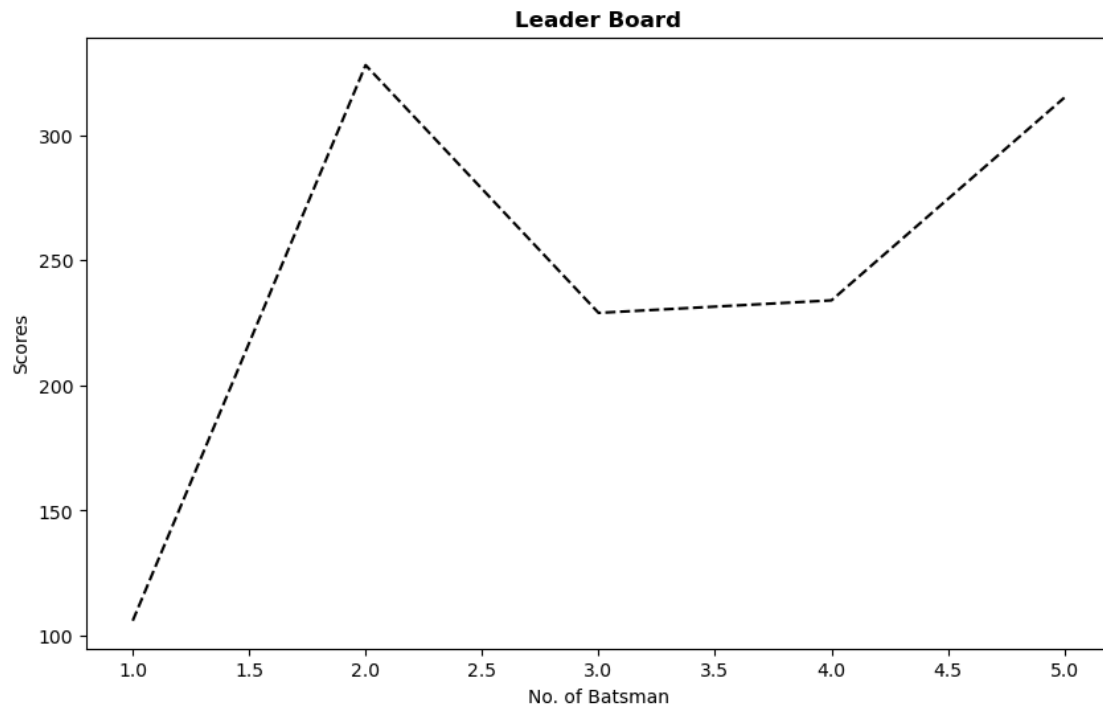
```
[[13 10 9 33 41]
 [63 46 90 42 87]
 [39 76 13 29 72]
 [82 9 29 78 36]
 [67 61 59 36 92]]
```

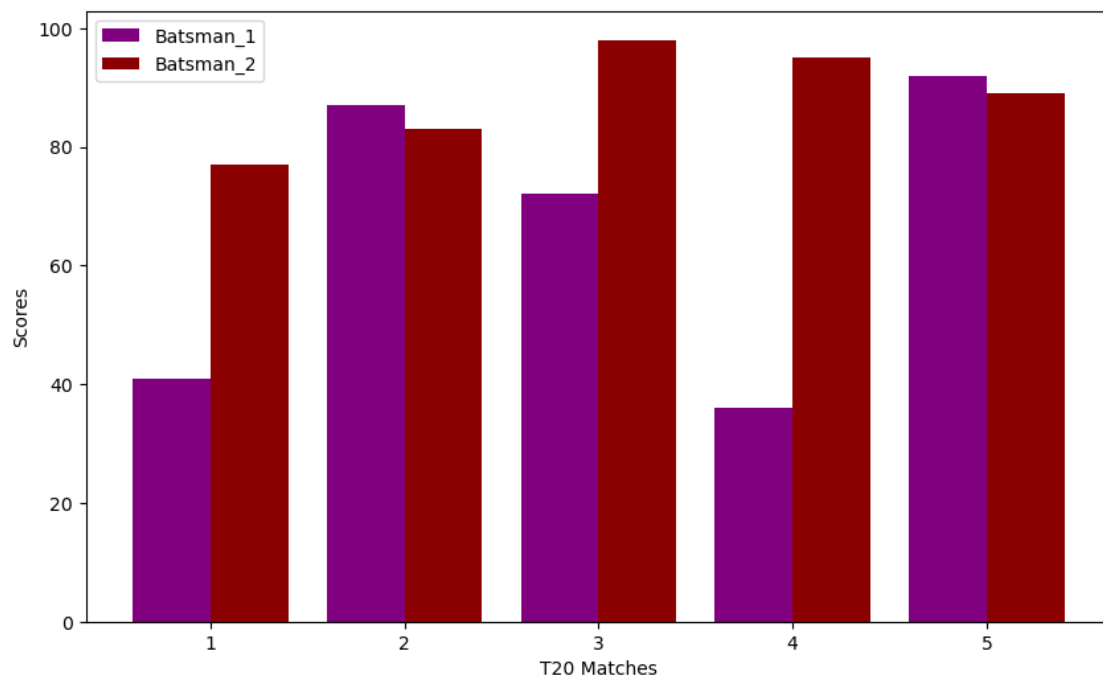
Task (ii):

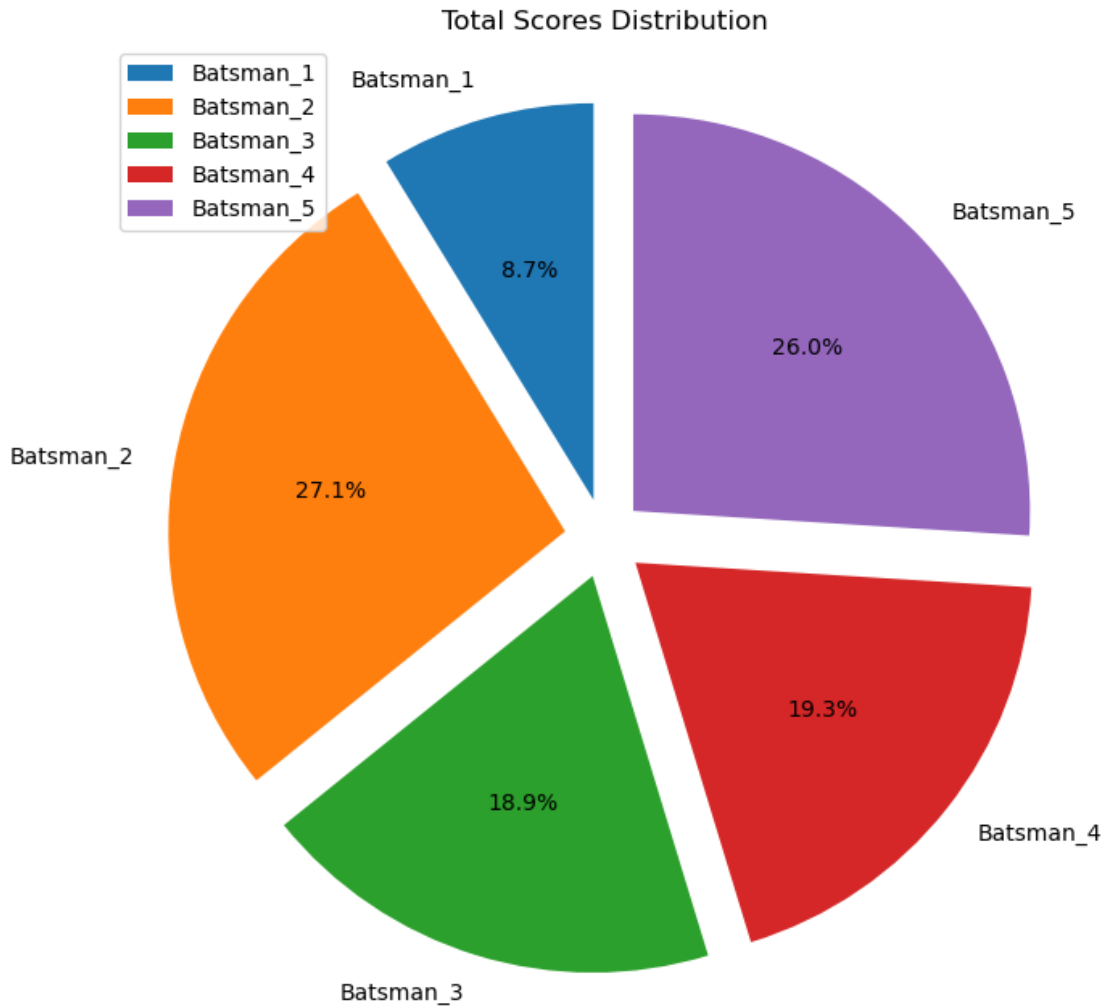
```
[[13 10 9 33 41 77 92]
 [63 46 90 42 87 83 71]
 [39 76 13 29 72 98 52]
 [82 9 29 78 36 95 61]
 [67 61 59 36 92 89 53]]
```

Task (iii):

```
[[ 13 10 9 33 41 77 92 106]
 [ 63 46 90 42 87 83 71 328]
 [ 39 76 13 29 72 98 52 229]
 [ 82 9 29 78 36 95 61 234]
 [ 67 61 59 36 92 89 53 315]]
```







**18.0.1 Write a program to build 6 graphs(3 row and 2 column) using subplot function for given data:**

Subplot 1: Draw a line from (5,5) to (10,17) to (25,25) to (60,40) to (80,30) with suitable label in the x axis, y axis and a title. Line color should be green. Line style should be dotted. Marker should be diamond. Subplot 2: Write a Python program to create bar plot of scores by group and gender. Give suitable label in the x axis, y axis and a title. Colors of all label should be black and title should be bold. Color of bar plot of men and women scores should be green and red. Data: Scores\_men = (22, 30, 35, 35, 26) Scores\_women = (25, 32, 30, 35, 29) Subplot 3: Write a Python programming to create a pie chart with a title of the popularity of Car company. Make multiple wedges of the pie. Also show the percentage. data: Car : Maruti Suzuki, Hyundai, Kia, Toyota, Honda Popularity: 25,50,30,20,35 Subplot 4: Write a Python program to draw a scatter plot comparing two subject marks of Mathematics and Science. Use marks of 10 students. Marker of mathematics and science should be circle and star. Colors of marker of mathematics and science should be yellow and blue. Test Data: math\_marks = [88, 92, 80, 89, 100, 80, 60,

100, 80, 34] science\_marks = [35, 79, 79, 48, 100, 88, 32, 45, 20, 30] Subplot 5: Write a Python programming to display a horizontal bar chart of the popularity of programming Languages. Colors of all programming Languages should be different. Give suitable label in the x axis, y axis and a title. data: Programming languages: Java, Python, PHP, JavaScript, C, C++ Popularity: 20,100,25,30,45,50 Subplot 6: Write a Python programming to display a Histogram chart for given data. Color of chart should be red. Data=[10,20,20,30,30,30,40,40,40,40,50,50,50,60,60,70]"

```
[10]: import matplotlib.pyplot as plt

# Subplot 1
plt.subplot(3, 2, 1)
x_values = [5, 10, 25, 60, 80]
y_values = [5, 17, 25, 40, 30]
plt.plot(x_values, y_values, color='green', linestyle='dotted', marker='D')
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.title('Line Plot with Dotted Line and Diamond Marker')

# Subplot 2
plt.subplot(3, 2, 2)
Scores_men = (22, 30, 35, 35, 26)
Scores_women = (25, 32, 30, 35, 29)
groups = ['Group 1', 'Group 2', 'Group 3', 'Group 4', 'Group 5']
bar_width = 0.35
bar_positions_men = range(len(groups))
bar_positions_women = [pos + bar_width for pos in bar_positions_men]

plt.bar(bar_positions_men, Scores_men, color='green', width=bar_width,
        label='Men')
plt.bar(bar_positions_women, Scores_women, color='red', width=bar_width,
        label='Women')
plt.xlabel('Groups')
plt.ylabel('Scores')
plt.title('Bar Plot of Scores by Group and Gender')
# plt.xticks([pos + bar_width / 2 for pos in bar_positions_men], groups)
# plt.legend(fontsize='small')

# Subplot 3
plt.subplot(3, 2, 3)
car_labels = ['Maruti Suzuki', 'Hyundai', 'Kia', 'Toyota', 'Honda']
popularity = [25, 50, 30, 20, 35]
plt.pie(popularity, labels=car_labels, autopct='%1.1f%%', startangle=90)
plt.title('Popularity of Car Companies')

# Subplot 4
plt.subplot(3, 2, 4)
math_marks = [88, 92, 80, 89, 100, 80, 60, 100, 80, 34]
```

```

science_marks = [35, 79, 79, 48, 100, 88, 32, 45, 20, 30]
plt.scatter(range(1, 11), math_marks, marker='o', color='yellow',
    ↪label='Mathematics')
plt.scatter(range(1, 11), science_marks, marker='*', color='blue',
    ↪label='Science')
plt.xlabel('Students')
plt.ylabel('Marks')
plt.title('Scatter Plot: Mathematics vs Science')
# plt.legend(fontsize='small')

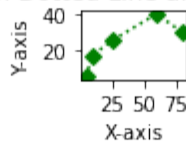
# Subplot 5
plt.subplot(3, 2, 5)
languages = ['Java', 'Python', 'PHP', 'JavaScript', 'C', 'C++']
popularity_languages = [20, 100, 25, 30, 45, 50]
plt.barh(languages, popularity_languages, color=['orange', 'green', 'blue',
    ↪'red', 'purple', 'pink'])
plt.xlabel('Popularity')
plt.ylabel('Prog Lang')
plt.title('Popularity of Programming Languages')

# Subplot 6
plt.subplot(3, 2, 6)
data_histogram = [10, 20, 20, 30, 30, 30, 40, 40, 40, 40, 50, 50, 50, 60, 60,
    ↪70]
plt.hist(data_histogram, color='red', bins=10, edgecolor='black')
plt.xlabel('Values')
plt.ylabel('Frequency')
plt.title('Histogram Chart')

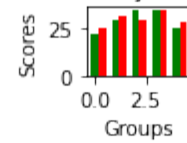
plt.tight_layout(w_pad=20.0) # Increase both vertical and horizontal spacing,
    ↪add rect parameter for title
plt.show()

```

Line Plot with Dotted Line and Diamond Marker



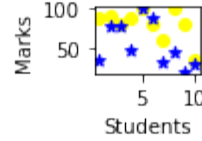
Bar Plot of Scores by Group and Gender



Popularity of Car Companies



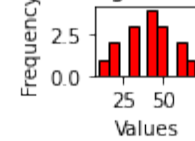
Scatter Plot: Mathematics vs Science



Popularity of Programming Languages



Histogram Chart



**18.0.2 Get total profit of all months and show line plot with the following Style properties:**

- Line Style dotted and Line-color should be red
- Show legend at the lower right location.
- X label name = Month Number
- Y label name = Total Profits
- Add a circle marker
- Line marker color as blue
- Line marker size as 5
- Line width should be 3

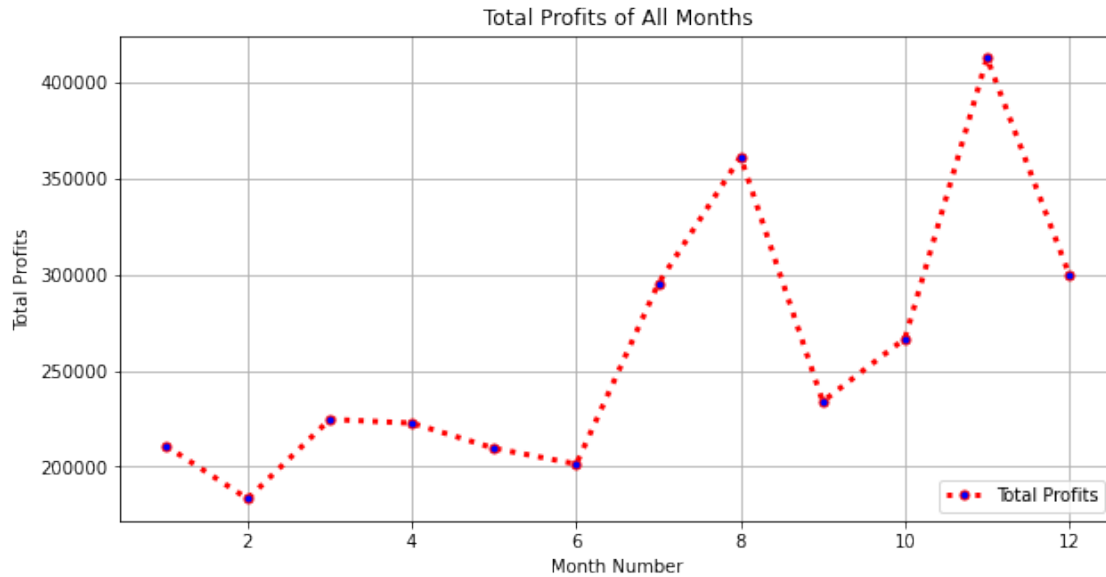
```
[11]: import matplotlib.pyplot as plt

# Month Number and Total Profit data
month_numbers = list(range(1, 13))
total_profits = [
    211000, 183300, 224700, 222700,
    209600, 201400, 295500, 361400,
    234000, 266700, 412800, 300200
]

plt.figure(figsize=(10, 5))
plt.plot(month_numbers, total_profits,
         linestyle='dotted', color='red', marker='o',
         markerfacecolor='blue', markersize=5,
         linewidth=3, label='Total Profits')

plt.xlabel('Month Number')
plt.ylabel('Total Profits')
plt.title('Total Profits of All Months')
plt.legend(loc="lower right")
plt.grid(True)
plt.show()
```





## 19 There is an array of scores of 5 Batsmen in 4 T20 Matches. Which is given below.

Scores= [[31, 12, 19, 53], [67, 48, 95, 83], [59, 67, 13, 59], [62, 29, 99, 88], [87, 91, 69, 76]]

1. Find the maximum score in T\_20-3 and print it (use only the numpy module)
2. Find the minimum score of YUVRAJ and print it (use only the numpy module)
3. Add an extra column with the sum of all 4 T20 Matches' scores of each batsman in the array created and print it. (use only the numpy module)
4. Sort the array (non-ascending order) on the total score of each batsman column (one added in the above task.) and print the sorted array.
5. make a new array with the sum of all five-batsman runs of each match and print a new array (use only the numpy module)
6. make a new array which represents if the batsman's score greater than 30 denote 1 else 0(use only the numpy module)
7. Create a Histogram of the SACHIN score in all matches, showing the frequency count in a range of 0-20,20-40, and 40-60. (use the Scores array for the SACHIN score)

Note: You can't directly print the answers to any question without using logic.

In the below example, for that code, you got zero marks.

Example: find the maximum run from all matches:

```
print("99")
```

```
print(Scores[3,2])
```

```
[6]: import numpy as np
import matplotlib.pyplot as plt
scores= [[31, 12, 19, 53],
[67, 48, 95, 83],
[59, 67, 13, 59],
[62, 29, 99, 88],
[87, 91, 69, 76]]
scores=np.array(scores)#-----0.5 mark
print("1.Find the maximum score in T_20-3 and print it (use only the numpy_
↳module)")
max=0
for i in scores[ : ,2]:
    if i>max:
        max=i
print("The maximum score in T_20-3",max)#-----output with logic 1_
↳mark
print("2. Find the minimum score of YUVRAJ and print it (use only the numpy_
↳module)")
min=scores[2,0] #-----here student take min=100 or any grater value
for i in scores[2, : ]:
    if i<min:
        min=i
print("minimum score of YUVRAJ",min)#-----output with logic 1 mark
print("""3.Add an extra column with the sum of all 4 T20 Matches'
scores of each batsman in the array created and print it. (use only the numpy_
↳module)""")
print(" ")
batsman_total=[]
for i in range(0,5):
    add=0
    for j in range(0,4):
        add+=scores[i,j]
    batsman_total.append(add)
batsman_total=np.array(batsman_total).reshape(5,1)
scores=np.concatenate((scores,batsman_total),axis=1)
print("updated array",scores)#-----output with logic 1.5 mark
print("""4. Sort the array (non-ascending order) on the total score of each_
↳batsman column
(one added in the above task.) and print the sorted array.""")
new_list=list(scores)
sorted_array=sorted(new_list,key=lambda x:x[-1],reverse=True)
print("sorted_array")
sorted_array=np.array(sorted_array)
print(sorted_array)#-----output with logic 1 mark
```

```

print("""5.make a new array with the sum of all five-batsman runs of each match_
↳and
print a new array (use only the numpy module)""")
match_total=[]
for i in range(0,4):
    add=0
    for j in range(0,5):
        add+=scores[j,i]
    match_total.append(add)
match_total=np.array(match_total).reshape(1,4)
new=np.concatenate((scores[:,-1],match_total),axis=0)
print("new array")
print(new)#-----output with logic 1.5 mark
print("""6. make a new array which represents if the batsman's score greater_
↳than
30 denote 1 else 0(use only the numpy module)""")
print(np.where(scores[: , :-1]>30,1,0))#-----output with logic 1 mark
print("""7.Create a Histogram of the SACHIN score in all matches,
showing the frequency count in a range of 0-20,20-40, and 40-60. (use the_
↳Scores array for the SACHIN score)""")
plt.hist(scores[0],bins=[0,20,40,60])
plt.show()#-----output with logic 1.5 mark

```

- 1.Find the maximum score in T\_20-3 and print it (use only the numpy module)  
The maximum score in T\_20-3 99
2. Find the minimum score of YUVRAJ and print it (use only the numpy module)  
minimum score of YUVRAJ 13
- 3.Add an extra column with the sum of all 4 T20 Matches'  
scores of each batsman in the array created and print it. (use only the numpy module)

```

updated array [[ 31  12  19  53 115]
[ 67  48  95  83 293]
[ 59  67  13  59 198]
[ 62  29  99  88 278]
[ 87  91  69  76 323]]

```

4. Sort the array (non-ascending order) on the total score of each batsman column

(one added in the above task.) and print the sorted array.

```

sorted_array
[[ 87  91  69  76 323]
[ 67  48  95  83 293]
[ 62  29  99  88 278]
[ 59  67  13  59 198]
[ 31  12  19  53 115]]

```

- 5.make a new array with the sum of all five-batsman runs of each match and print a new array (use only the numpy module)

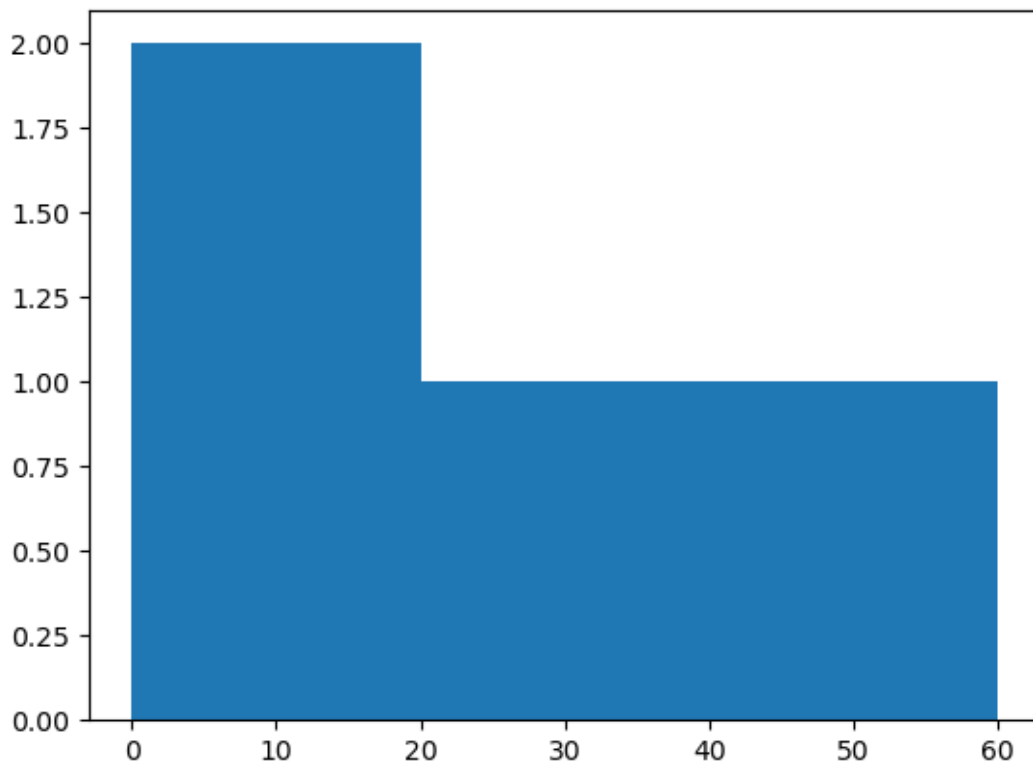
new array

```
[[ 31  12  19  53]
 [ 67  48  95  83]
 [ 59  67  13  59]
 [ 62  29  99  88]
 [ 87  91  69  76]
 [306 247 295 359]]
```

6. make a new array which represents if the batsman's score greater than 30 denote 1 else 0(use only the numpy module)

```
[[1 0 0 1]
 [1 1 1 1]
 [1 1 0 1]
 [1 0 1 1]
 [1 1 1 1]]
```

7. Create a Histogram of the SACHIN score in all matches, showing the frequency count in a range of 0-20, 20-40, and 40-60. (use the Scores array for the SACHIN score)



[ ]: