Intro to data visualization with matplotlib

Pie Chart

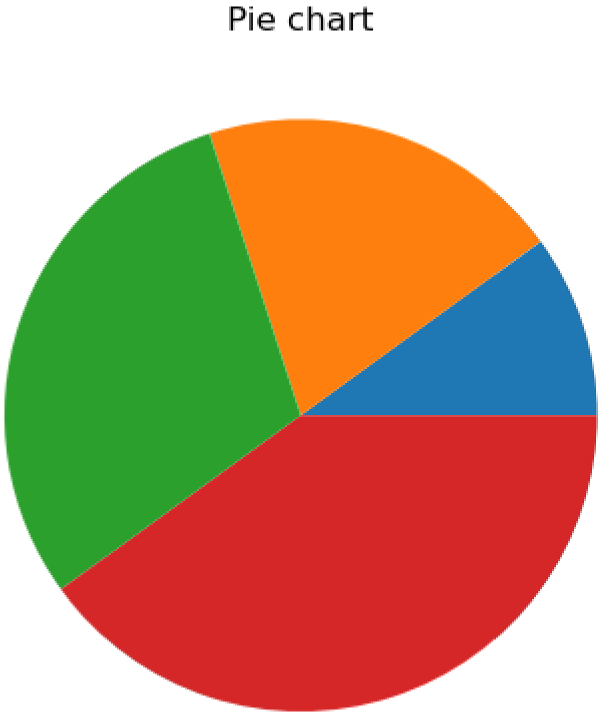
A Pie Chart is a circular statistical plot that can display only one series of data. The area of the chart is the total percentage of the given data. The area of slices of the pie represents the percentage of the parts of the data. The slices of pie are called wedges. The area of the wedge is determined by the length of the arc of the wedge. It can be created using the pie() method.

In [1]:

```
# importing matplotlib
import matplotlib.pyplot as plt
```

In [2]:

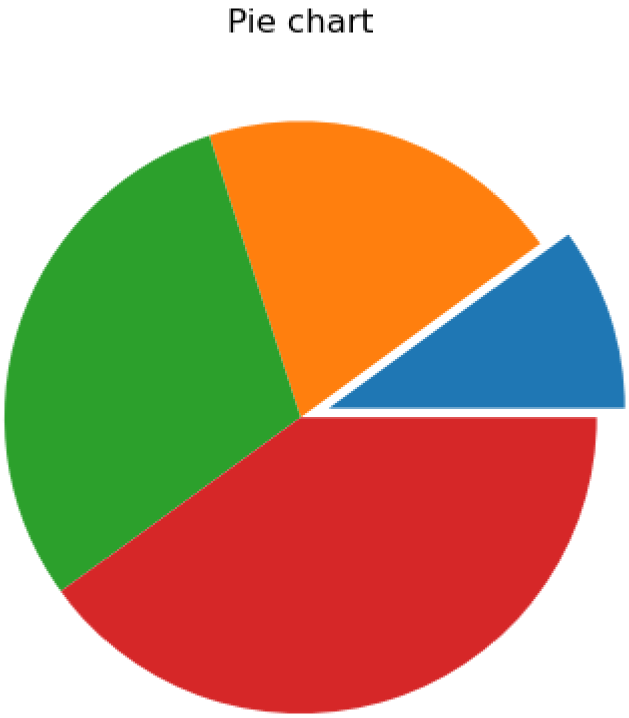
```
# data to display on plots
x = [1, 2, 3, 4]

# this will explode the 1st wedge
# i.e. will separate the 1st wedge
# from the chart
e =(0.1, 0, 0, 0)

# This will plot a simple pie chart
plt.pie(x)

# Title to the plot
plt.title("Pie chart")
plt.show()

```



Pie chart

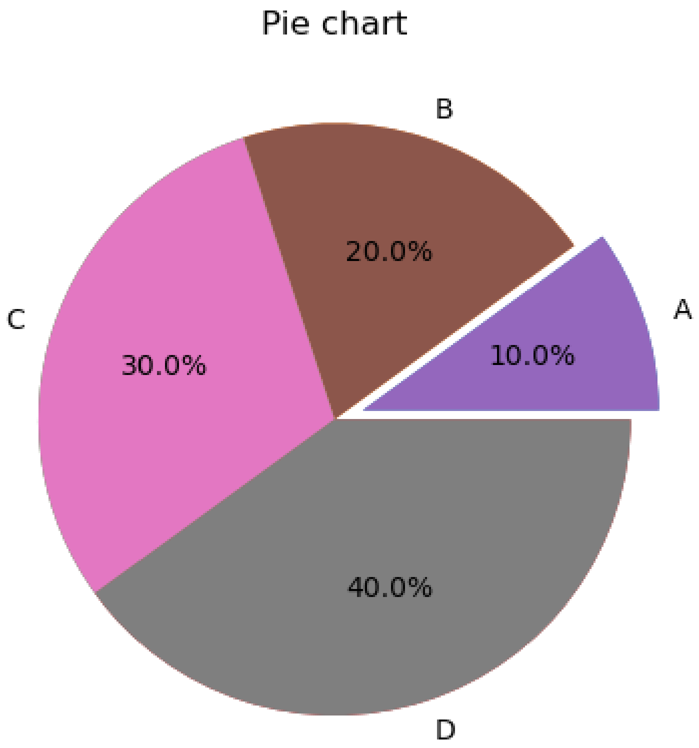
In [3]:

```
import matplotlib.pyplot as plt
# data to display on plots
x = [1, 2, 3, 4]

# this will explode the 1st wedge
# i.e. will separate the 1st wedge
# from the chart
e =(0.1, 0, 0, 0)

# This will plot a simple pie chart
plt.pie(x, explode = e)

# Title to the plot
plt.title("Pie chart")
plt.show()

```



In [4]:

```
import matplotlib.pyplot as plt
# data to display on plots
x = [1, 2, 3, 4]

# this will explode the 1st wedge
# i.e. will separate the 1st wedge
# from the chart
e =(0.1, 0, 0, 0)

# This will plot a simple pie chart
plt.pie(x, explode = e)

# Create a pie chart with percentage labels
plt.pie(x, labels=['A', 'B', 'C', 'D'], autopct='%1.1f%%', explode=e)

# Title to the plot
plt.title("Pie chart")
plt.show()

```

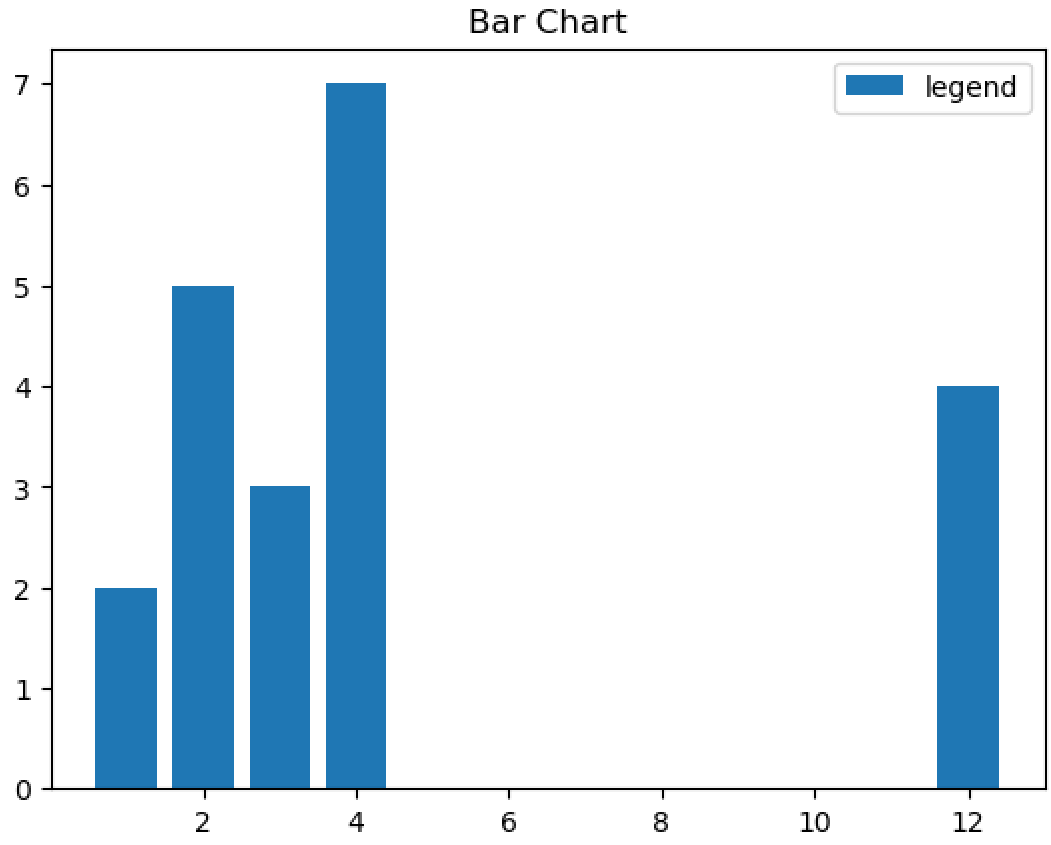


Bar Plot

A bar plot or bar chart is a graph that represents the category of data with rectangular bars with lengths and heights that is proportional to the values which they represent. The bar plots can be plotted horizontally or vertically. A bar chart describes the comparisons between the discrete categories. It can be created using the bar() method.

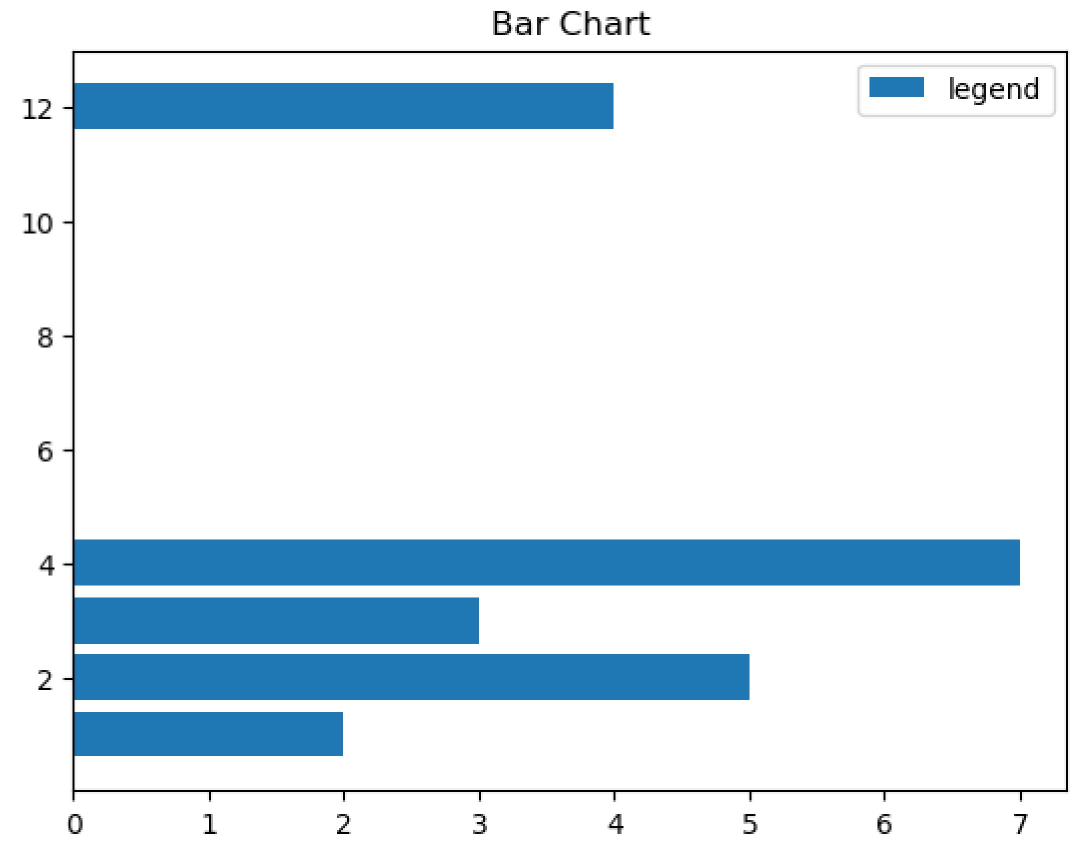
In [6]:

```
# This will plot a simple vertical bar chart
# data to display on plots
x = [3, 1, 3, 12, 2, 4, 4]
y = [3, 2, 1, 4, 5, 6, 7]
plt.bar(x, y)

# Title to the plot
plt.title("Bar Chart")

# Adding the legends
plt.legend(["legend"])
plt.show()
```



In [7]:

```python
# This will plot a simple horizontal bar chart
plt.barh(x, y)

# Title to the plot
plt.title("Bar Chart")

# Adding the legends
plt.legend(["legend"])
plt.show()
```

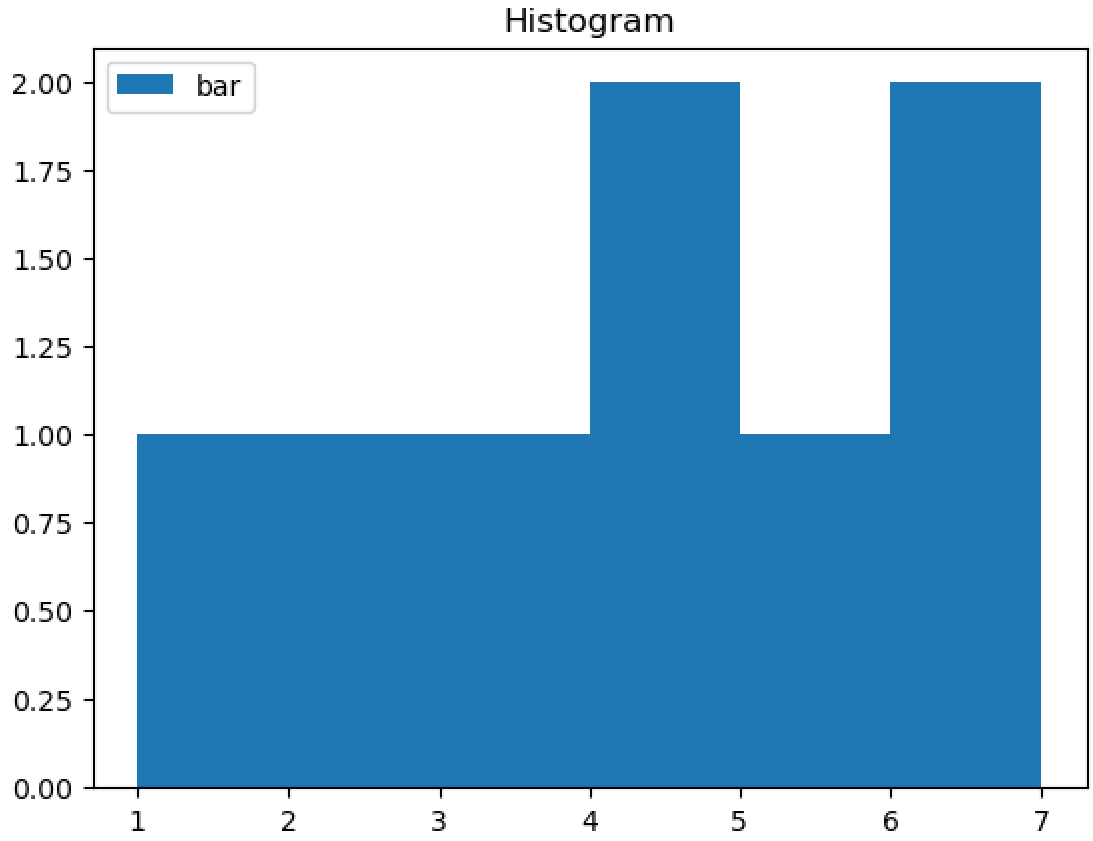


Histograms

A histogram is basically used to represent data in the form of some groups. It is a type of bar plot where the X-axis represents the bin ranges while the Y-axis gives information about frequency. To create a histogram the first step is to create a bin of the ranges, then distribute the whole range of the values into a series of intervals, and count the values which fall into each of the intervals. Bins are clearly identified as consecutive, non-overlapping intervals of variables. The hist() function is used to compute and create histogram of x.

In [8]:

```
# data to display on plots
x = [1, 2, 3, 4, 5, 6, 7, 4]
# This will plot a simple histogram
plt.hist(x, bins = [1, 2, 3, 4, 5, 6, 7])
# Title to the plot
plt.title("Histogram")
# Adding the legends
plt.legend(["bar"])
plt.show()
```

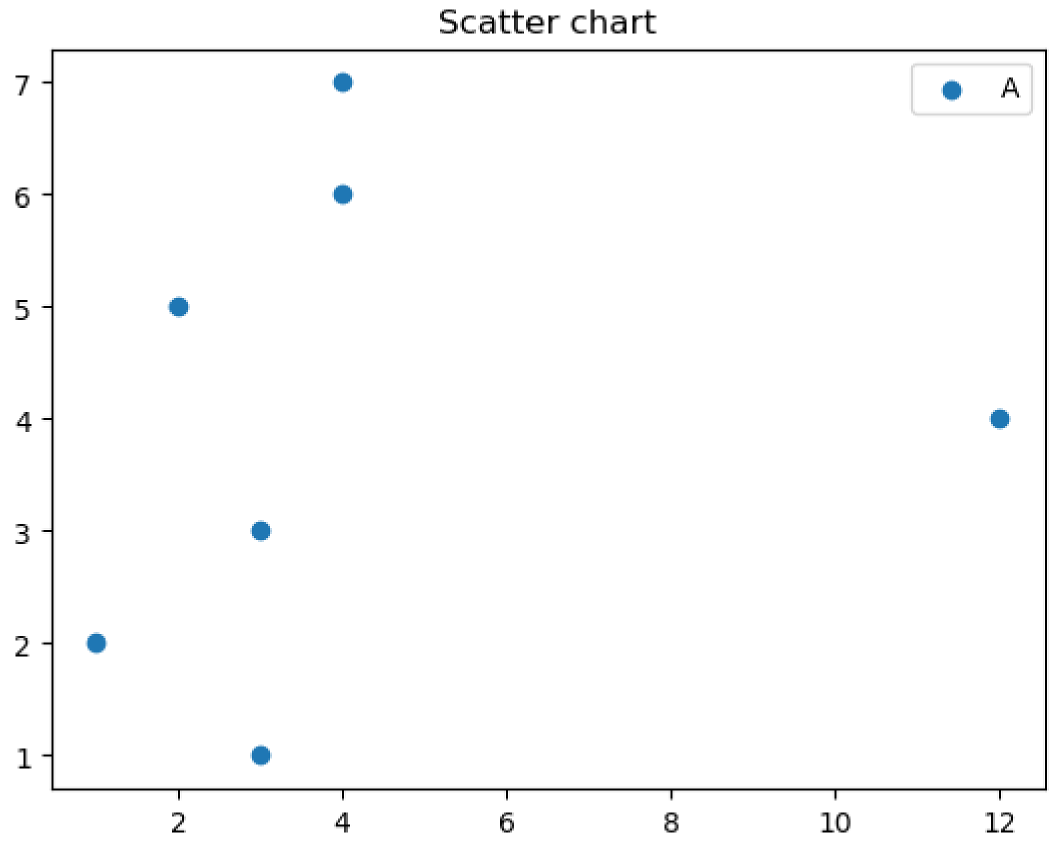


Scatter Plot

Scatter plots are used to observe the relationship between variables and use dots to represent the relationship between them. The scatter() method in the matplotlib library is used to draw a scatter plot.

In [10]:

```
 # data to display on plots
x = [3, 1, 3, 12, 2, 4, 4]
y = [3, 2, 1, 4, 5, 6, 7]
# This will plot a simple scatter chart
plt.scatter(x, y)

# Adding legend to the plot
plt.legend("A")

# Title to the plot
plt.title("Scatter chart")
plt.show()
```



In [ ]:

```

```