SANJANA DEVI

AI/ML Student | Intern @ CodroidHub Private Limited | First-Year Engineering Student.

Artificial Intelligence / Machine Learning Bootcamp

Data Analysis and Visualization

AGENDA

1. **Import Statement For the Libraries**
2. **Load Iris dataset**
3. **Bar chart**
4. **Histogram**
5. **Summary**

Import Statement For the Libraries

In [2]:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import sklearn as sk
```

`import pandas as pd`

- **What it does**: Imports the **Pandas** library and gives it a shortcut name `pd`.
- **Why it's used**: Pandas is mainly used to work with **tabular data** (like Excel or CSV files). It helps you:
  - Read files (CSV, Excel)
  - Clean, filter, and organize data
  - Perform data analysis

Example:

```
df = pd.read_csv("data.csv")  # Load a CSV file into a DataFrame
print(df.head())              # Show first 5 rows of data
```

`import matplotlib.pyplot as plt`

- **What it does**: Imports the **Pyplot** module from the **Matplotlib** library and gives it a shortcut name `plt`.
- **Why it's used**: Pyplot is used to make basic **charts and graphs**, like:
  - Line plots
  - Bar charts
  - Histograms
  - Pie charts

Example:

```
x = [1, 2, 3]
y = [4, 5, 6]
plt.plot(x, y)        # Create a line plot
plt.title("Line Graph")
plt.show()              # Display the graph
```

`import seaborn as sns`

- **What it does**: Imports the **Seaborn** library and gives it a shortcut name `sns`.
- **Why it's used**: Seaborn is built on top of Matplotlib and is used to make **more attractive and statistical plots** easily:
    - Heatmaps
    - Correlation plots
    - Box plots
    - Distribution plots

Example:

```
import seaborn as sns
tips = sns.load_dataset("tips")   # Sample dataset
sns.boxplot(x="day", y="total_bill", data=tips)
plt.show()
```

`import sklearn as sk`

- **What it does**: Imports the **Scikit-learn** library and gives it a shortcut name `sk` (not commonly used, but still works).
- **Why it's used**: Scikit-learn is a powerful library for **Machine Learning**. It allows you to:
    - Build and train models (e.g., Linear Regression, Decision Tree)
    - Split data into training and test sets
    - Evaluate model performance

Note: Usually, we **import parts of `sklearn` directly**, like this:

```
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
```

Example:

```
from sklearn.linear_model import LinearRegression
model = LinearRegression()
model.fit(X_train, y_train)
```

Summary Table

Import	Library	Purpose
`import pandas as pd`	Pandas	Data handling (CSV, tables)
`import matplotlib.pyplot as plt`	Matplotlib	Basic charts (line, bar)
`import seaborn as sns`	Seaborn	Beautiful charts with stats
`import sklearn as sk`	Scikit-learn	Machine Learning tools

Load Iris Dataset

In [3]:

```
# Load Iris Dataset
from sklearn import datasets

iris = datasets.load_iris()
print(iris.feature_names)
print(iris)
df = pd.DataFrame(iris.data, columns=iris.feature_names)
df['species'] = iris.target

# Preview the data
df.head()
```

```
['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)', 'petal width (cm)']
{'data': array([[5.1, 3.5, 1.4, 0.2],
       [4.9, 3. , 1.4, 0.2],
       [4.7, 3.2, 1.3, 0.2],
       [4.6, 3.1, 1.5, 0.2],
       [5. , 3.6, 1.4, 0.2],
       [5.4, 3.9, 1.7, 0.4],
       [4.6, 3.4, 1.4, 0.3],
       [5. , 3.4, 1.5, 0.2],
       [4.4, 2.9, 1.4, 0.2],
       [4.9, 3.1, 1.5, 0.1],
       [5.4, 3.7, 1.5, 0.2],
       [4.8, 3.4, 1.6, 0.2],
       [4.8, 3. , 1.4, 0.1],
       [4.3, 3. , 1.1, 0.1],
       [5.8, 4. , 1.2, 0.2],
       [5.7, 4.4, 1.5, 0.4],
       [5.4, 3.9, 1.3, 0.4],
       [5.1, 3.5, 1.4, 0.3],
       [5.7, 3.8, 1.7, 0.3],
       [5.1, 3.8, 1.5, 0.3],
       [5.4, 3.4, 1.7, 0.2],
       [5.1, 3.7, 1.5, 0.4],
       [4.6, 3.6, 1. , 0.2],
       [5.1, 3.3, 1.7, 0.5],
       [4.8, 3.4, 1.9, 0.2],
       [5. , 3. , 1.6, 0.2],
       [5. , 3.4, 1.6, 0.4],
       [5.2, 3.5, 1.5, 0.2],
       [5.2, 3.4, 1.4, 0.2],
       [4.7, 3.2, 1.6, 0.2],
       [4.8, 3.1, 1.6, 0.2],
       [5.4, 3.4, 1.5, 0.4],
       [5.2, 4.1, 1.5, 0.1],
       [5.5, 4.2, 1.4, 0.2],
       [4.9, 3.1, 1.5, 0.2],
       [5. , 3.2, 1.2, 0.2],
       [5.5, 3.5, 1.3, 0.2],
       [4.9, 3.6, 1.4, 0.1],
       [4.4, 3. , 1.3, 0.2],
       [5.1, 3.4, 1.5, 0.2],
       [5. , 3.5, 1.3, 0.3],
       [4.5, 2.3, 1.3, 0.3],
       [4.4, 3.2, 1.3, 0.2],
       [5. , 3.5, 1.6, 0.6],
       [5.1, 3.8, 1.9, 0.4],
       [4.8, 3. , 1.4, 0.3],
       [5.1, 3.8, 1.6, 0.2],
       [4.6, 3.2, 1.4, 0.2],
       [5.3, 3.7, 1.5, 0.2],
       [5. , 3.3, 1.4, 0.2],
       [7. , 3.2, 4.7, 1.4],
       [6.4, 3.2, 4.5, 1.5],
       [6.9, 3.1, 4.9, 1.5],
       [5.5, 2.3, 4. , 1.3],
       [6.5, 2.8, 4.6, 1.5],
       [5.7, 2.8, 4.5, 1.3],
       [6.3, 3.3, 4.7, 1.6],
       [4.9, 2.4, 3.3, 1. ],
       [6.6, 2.9, 4.6, 1.3],
       [5.2, 2.7, 3.9, 1.4],
       [5. , 2. , 3.5, 1. ],
       [5.9, 3. , 4.2, 1.5],
       [6. , 2.2, 4. , 1. ],
       [6.1, 2.9, 4.7, 1.4],
       [5.6, 2.9, 3.6, 1.3],
       [6.7, 3.1, 4.4, 1.4],
       [5.6, 3. , 4.5, 1.5],
       [5.8, 2.7, 4.1, 1. ],
       [6.2, 2.2, 4.5, 1.5],
       [5.6, 2.5, 3.9, 1.1],
       [5.9, 3.2, 4.8, 1.8],
       [6.1, 2.8, 4. , 1.3],
       [6.3, 2.5, 4.9, 1.5],
       [6.1, 2.8, 4.7, 1.2],
       [6.4, 2.9, 4.3, 1.3],
       [6.6, 3. , 4.4, 1.4],
       [6.8, 2.8, 4.8, 1.4],
       [6.7, 3. , 5. , 1.7],
       [6. , 2.9, 4.5, 1.5],
       [5.7, 2.6, 3.5, 1. ],
       [5.5, 2.4, 3.8, 1.1],
       [5.5, 2.4, 3.7, 1. ],
       [5.8, 2.7, 3.9, 1.2],
       [6. , 2.7, 5.1, 1.6],
       [5.4, 3. , 4.5, 1.5],
       [6. , 3.4, 4.5, 1.6],
       [6.7, 3.1, 4.7, 1.5],
       [6.3, 2.3, 4.4, 1.3],
       [5.6, 3. , 4.1, 1.3],
       [5.5, 2.5, 4. , 1.3],
       [5.5, 2.6, 4.4, 1.2],
       [6.1, 3. , 4.6, 1.4],
       [5.8, 2.6, 4. , 1.2],
       [5. , 2.3, 3.3, 1. ],
       [5.6, 2.7, 4.2, 1.3],
       [5.7, 3. , 4.2, 1.2],
       [5.7, 2.9, 4.2, 1.3],
       [6.2, 2.9, 4.3, 1.3],
       [5.1, 2.5, 3. , 1.1],
       [5.7, 2.8, 4.1, 1.3],
       [6.3, 3.3, 6. , 2.5],
       [5.8, 2.7, 5.1, 1.9],
       [7.1, 3. , 5.9, 2.1],
       [6.3, 2.9, 5.6, 1.8],
       [6.5, 3. , 5.8, 2.2],
       [7.6, 3. , 6.6, 2.1],
       [4.9, 2.5, 4.5, 1.7],
       [7.3, 2.9, 6.3, 1.8],
       [6.7, 2.5, 5.8, 1.8],
       [7.2, 3.6, 6.1, 2.5],
       [6.5, 3.2, 5.1, 2. ],
       [6.4, 2.7, 5.3, 1.9],
       [6.8, 3. , 5.5, 2.1],
       [5.7, 2.5, 5. , 2. ],
       [5.8, 2.8, 5.1, 2.4],
       [6.4, 3.2, 5.3, 2.3],
       [6.5, 3. , 5.5, 1.8],
       [7.7, 3.8, 6.7, 2.2],
       [7.7, 2.6, 6.9, 2.3],
       [6. , 2.2, 5. , 1.5],
       [6.9, 3.2, 5.7, 2.3],
       [5.6, 2.8, 4.9, 2. ],
       [7.7, 2.8, 6.7, 2. ],
       [6.3, 2.7, 4.9, 1.8],
       [6.7, 3.3, 5.7, 2.1],
       [7.2, 3.2, 6. , 1.8],
       [6.2, 2.8, 4.8, 1.8],
       [6.1, 3. , 4.9, 1.8],
       [6.4, 2.8, 5.6, 2.1],
       [7.2, 3. , 5.8, 1.6],
       [7.4, 2.8, 6.1, 1.9],
       [7.9, 3.8, 6.4, 2. ],
       [6.4, 2.8, 5.6, 2.2],
       [6.3, 2.8, 5.1, 1.5],
       [6.1, 2.6, 5.6, 1.4],
       [7.7, 3. , 6.1, 2.3],
       [6.3, 3.4, 5.6, 2.4],
       [6.4, 3.1, 5.5, 1.8],
       [6. , 3. , 4.8, 1.8],
       [6.9, 3.1, 5.4, 2.1],
       [6.7, 3.1, 5.6, 2.4],
       [6.9, 3.1, 5.1, 2.3],
       [5.8, 2.7, 5.1, 1.9],
       [6.8, 3.2, 5.9, 2.3],
       [6.7, 3.3, 5.7, 2.5],
       [6.7, 3. , 5.2, 2.3],
       [6.3, 2.5, 5. , 1.9],
       [6.5, 3. , 5.2, 2. ],
       [6.2, 3.4, 5.4, 2.3],
       [5.9, 3. , 5.1, 1.8]]), 'target': array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2]), 'frame': None, 'target_names': array(['setosa',
'versicolor', 'virginica'], dtype='<U10'), 'DESCR': '.. _iris_dataset:\n\nIris plants dataset\n----------------
----\n\n**Data Set Characteristics:**\n\n:Number of Instances: 150 (50 in each of three classes)\n:Number of At
tributes: 4 numeric, predictive attributes and the class\n:Attribute Information:\n    - sepal length in cm\n
- sepal width in cm\n    - petal length in cm\n    - petal width in cm\n    - class:\n            - Iris-Setosa
\n            - Iris-Versicolour\n            - Iris-Virginica\n\n:Summary Statistics:\n\n============== ==== =
=== ======= ===== ====================\n                Min  Max   Mean    SD   Class Correlation\n============
== ==== ==== ======= ===== ====================\nsepal length:   4.3  7.9   5.84   0.83    0.7826\nsepal width:
2.0  4.4   3.05   0.43   -0.4194\npetal length:   1.0  6.9   3.76   1.76    0.9490  (high!)\npetal width:    0.
1  2.5   1.20   0.76    0.9565  (high!)\n============== ==== ==== ======= ===== ====================\n\n:Missin
g Attribute Values: None\n:Class Distribution: 33.3% for each of 3 classes.\n:Creator: R.A. Fisher\n:Donor: Mic
hael Marshall (MARSHALL%PLU@io.arc.nasa.gov)\n:Date: July, 1988\n\nThe famous Iris database, first used by Sir
```

```
R.A. Fisher. The dataset is taken\nfrom Fisher\'s paper. Note that it\'s the same as in R, but not as in the UC
I\nMachine Learning Repository, which has two wrong data points.\n\nThis is perhaps the best known database to
be found in the\npattern recognition literature.  Fisher\'s paper is a classic in the field and\nis referenced
frequently to this day.  (See Duda & Hart, for example.)  The\ndata set contains 3 classes of 50 instances each
, where each class refers to a\ntype of iris plant.  One class is linearly separable from the other 2; the\nlat
ter are NOT linearly separable from each other.\n\n.. dropdown:: References\n\n  - Fisher, R.A. "The use of mul
tiple measurements in taxonomic problems"\n    Annual Eugenics, 7, Part II, 179-188 (1936); also in "Contributi
ons to\n    Mathematical Statistics" (John Wiley, NY, 1950).\n  - Duda, R.O., & Hart, P.E. (1973) Pattern Class
ification and Scene Analysis.\n    (Q327.D83) John Wiley & Sons.  ISBN 0-471-22361-1.  See page 218.\n  - Dasar
athy, B.V. (1980) "Nosing Around the Neighborhood: A New System\n    Structure and Classification Rule for Reco
gnition in Partially Exposed\n    Environments".  IEEE Transactions on Pattern Analysis and Machine\n    Intell
igence, Vol. PAMI-2, No. 1, 67-71.\n  - Gates, G.W. (1972) "The Reduced Nearest Neighbor Rule".  IEEE Transacti
ons\n    on Information Theory, May 1972, 431-433.\n  - See also: 1988 MLC Proceedings, 54-64.  Cheeseman et al
"s AUTOCLASS II\n    conceptual clustering system finds 3 classes in the data.\n  - Many, many more ...\n', 'fe
ature_names': ['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)', 'petal width (cm)'], 'filename': '
iris.csv', 'data_module': 'sklearn.datasets.data'}
```

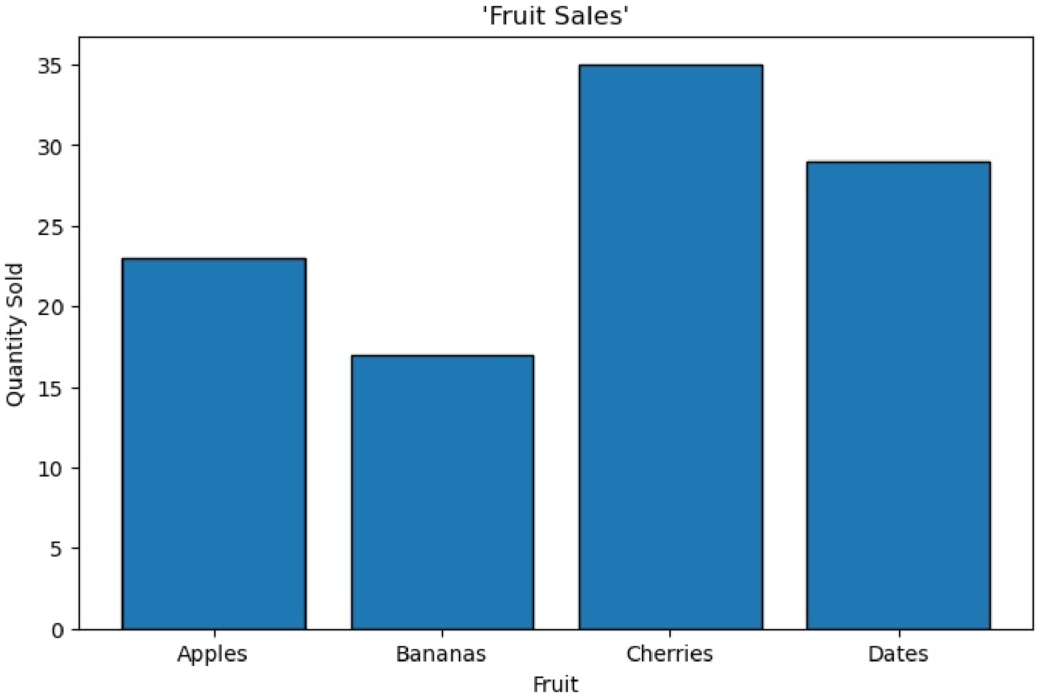
Out[3]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	species
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0

Bar Charts

In [7]:

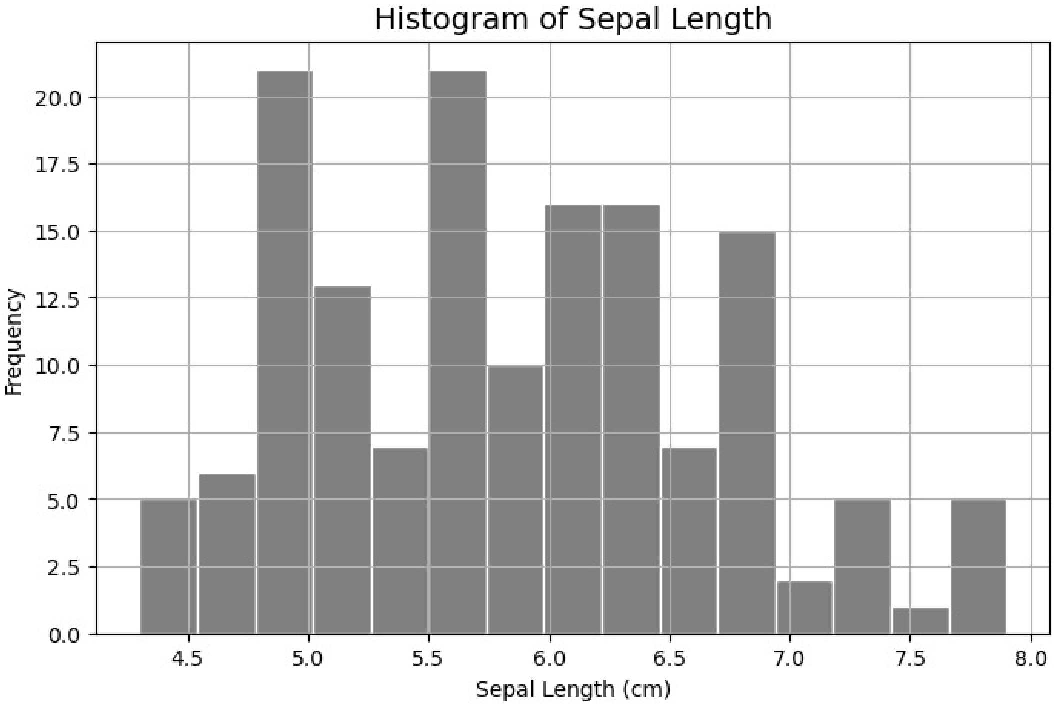
```python
# Sample data
Fruits = ['Apples', 'Bananas', 'Cherries', 'Dates']
Qunitity = [23, 17, 35, 29]

fig , ax = plt.subplots(figsize = (8,5))
# Plotting the bar chart
fig.patch.set_facecolor('white')
#ax.set_facecolor('black')
plt.bar(Fruits, Qunitity, edgecolor = 'black')

# Adding title and labels
plt.title("'Fruit Sales'")
plt.xlabel('Fruit')
plt.ylabel('Quantity Sold')

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



Histogram

In [8]:

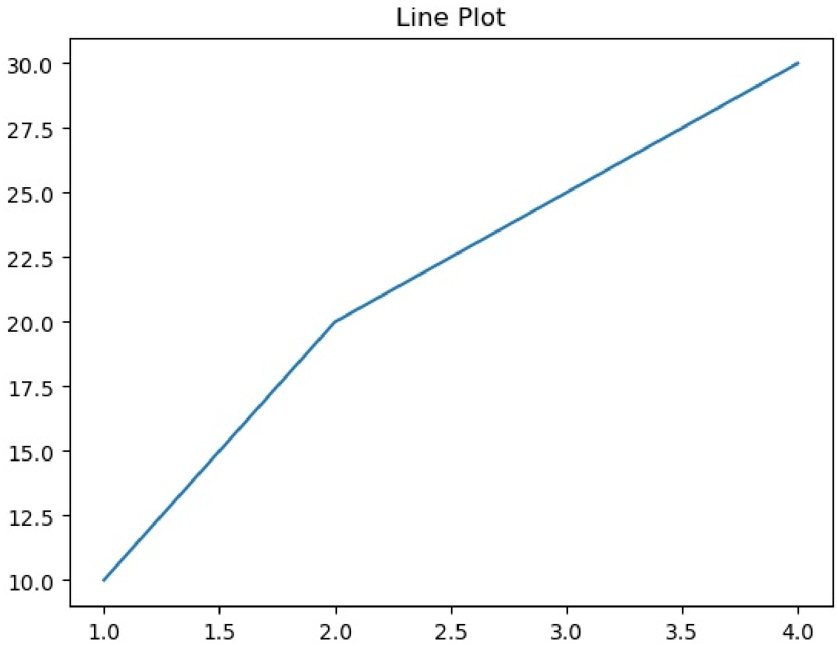
```
plt.figure(figsize=(8, 5))
plt.hist(df['sepal length (cm)'], bins=15, color='grey', edgecolor='white')

# Customization
plt.title("Histogram of Sepal Length", fontsize=14)
plt.xlabel("Sepal Length (cm)")
plt.ylabel("Frequency")
plt.grid(True)

plt.show()
```

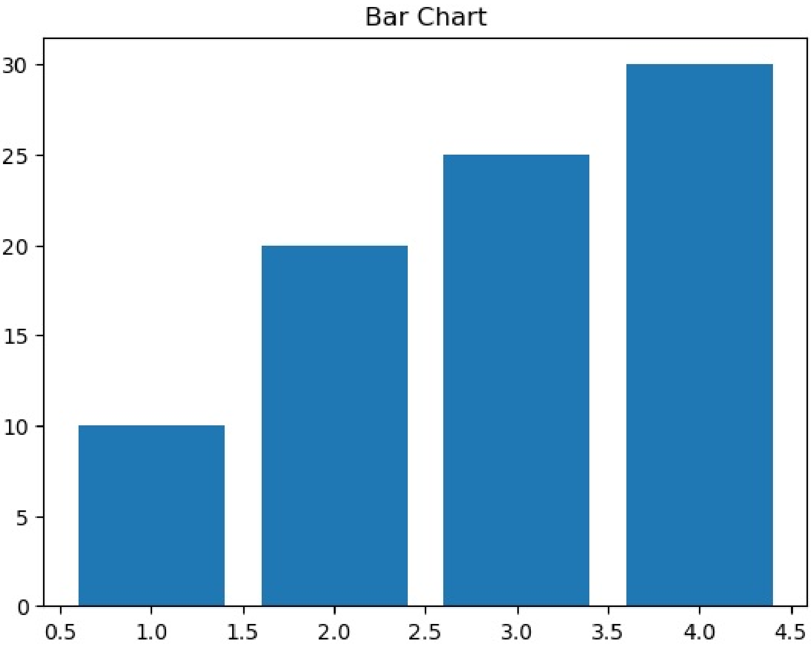


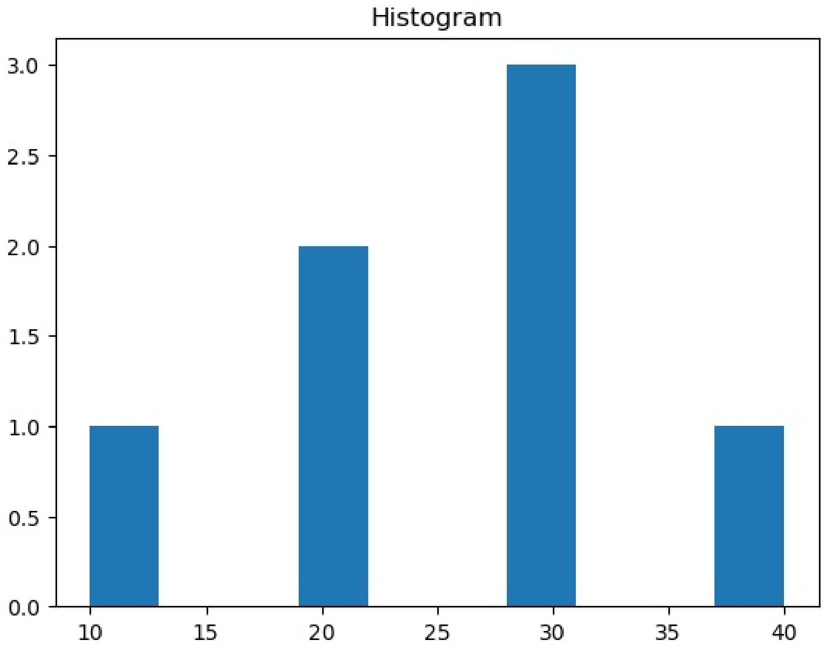
Summary

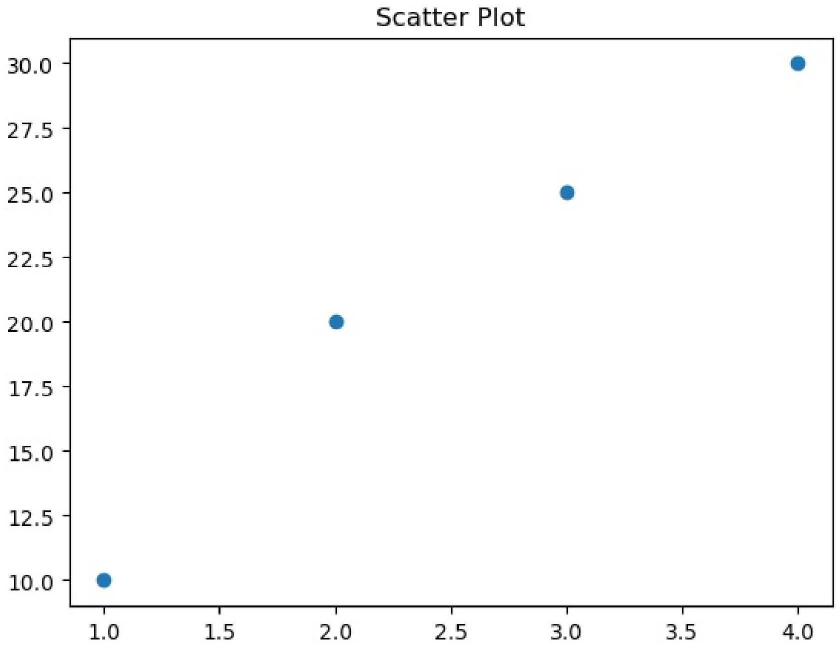
In [11…

```
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd

# Example data
x = [1, 2, 3, 4]
y = [10, 20, 25, 30]

# Line plot
plt.plot(x, y)
plt.title("Line Plot")
plt.show()

# Bar chart
plt.bar(x, y)
plt.title("Bar Chart")
plt.show()

# Histogram
data = [10, 20, 20, 30, 30, 30, 40]
plt.hist(data)
plt.title("Histogram")
plt.show()

# Scatter Plot
plt.scatter(x, y)
plt.title("Scatter Plot")
plt.show()
```


Line Plot
30.0
27.5
25.0
22.5
20.0
17.5
15.0
12.5
10.0
1.0
1.5
2.0
2.5
3.0
3.5
4.0


Bar Chart
30
25
20
15
10
5
0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5


Histogram
3.0
2.5
2.0
1.5
1.0
0.5
0.0
10
15
20
25
30
35
40



Connect @

Mail (Sanjana): sanjanadevibihana@gmail.com

Contact: +91-6283762268

SANJANA DEVI

AI/ML Student | Intern @ CodroidHub Private Limited | First-Year Engineering Student.

SANJANA DEVI