Level 1 : Beginner

Task 1 : Data Overview

Steps:

1. Load the dataset: Import the dataset into a data analysis tool such as Python with pandas or spreadsheet software.
2. Descriptive Statistics: Use descriptive functions (e.g., info() in pandas) to gather information about the number of entries, columns, and data types.

```
# Importing the libraries
import pandas as pd
```

```
#Loading Dataset
data=pd.read_csv("/content/Data_set 2 - Copy.csv")
data.head()
```

	gender	age	Investment_Avenues	Mutual_Funds	Equity_Market	Debentures	Government_Bonds	Fixed_De
0	Female	34	Yes	1	2	5	3	
1	Female	23	Yes	4	3	2	1	
2	Male	30	Yes	3	6	4	2	
3	Male	22	Yes	2	1	3	7	
4	Female	24	No	2	1	3	6	

5 rows × 24 columns

```
#Descriptive Statistics
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 40 entries, 0 to 39
Data columns (total 24 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   gender                          40 non-null     object
 1   age                             40 non-null     int64
 2   Investment_Avenues              40 non-null     object
 3   Mutual_Funds                    40 non-null     int64
 4   Equity_Market                   40 non-null     int64
 5   Debentures                      40 non-null     int64
 6   Government_Bonds                40 non-null     int64
 7   Fixed_Deposits                  40 non-null     int64
 8   PPF                             40 non-null     int64
 9   Gold                            40 non-null     int64
 10  Stock_Marktet                   40 non-null     object
 11  Factor                          40 non-null     object
 12  Objective                       40 non-null     object
```

```
 13  Purpose                          40 non-null     object
 14  Duration                         40 non-null     object
 15  Invest_Monitor                   40 non-null     object
 16  Expect                           40 non-null     object
 17  Avenue                           40 non-null     object
 18  What are your savings objectives?  40 non-null     object
 19  Reason_Equity                    40 non-null     object
 20  Reason_Mutual                    40 non-null     object
 21  Reason_Bonds                     40 non-null     object
 22  Reason_FD                        40 non-null     object
 23  Source                           40 non-null     object
dtypes: int64(8), object(16)
memory usage: 7.6+ KB
```

```
data.shape
```

```
(40, 24)
```

```
data.describe()
```

	age	Mutual_Funds	Equity_Market	Debentures	Government_Bonds	Fixed_Deposits	PPF
count	40.000000	40.000000	40.000000	40.000000	40.000000	40.000000	40.000000
mean	27.800000	2.550000	3.475000	5.750000	4.650000	3.575000	2.025000
std	3.560467	1.197219	1.131994	1.675617	1.369072	1.795828	1.609069
min	21.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
25%	25.750000	2.000000	3.000000	5.000000	4.000000	2.750000	1.000000
50%	27.000000	2.000000	4.000000	6.500000	5.000000	3.500000	1.000000
75%	30.000000	3.000000	4.000000	7.000000	5.000000	5.000000	2.250000
max	35.000000	7.000000	6.000000	7.000000	7.000000	7.000000	6.000000

```
data.columns
```

```
Index(['gender', 'age', 'Investment_Avenues', 'Mutual_Funds', 'Equity_Market',
       'Debentures', 'Government_Bonds', 'Fixed_Deposits', 'PPF', 'Gold',
       'Stock_Marktet', 'Factor', 'Objective', 'Purpose', 'Duration',
       'Invest_Monitor', 'Expect', 'Avenue',
       'What are your savings objectives?', 'Reason_Equity', 'Reason_Mutual',
       'Reason_Bonds', 'Reason_FD', 'Source'],
      dtype='object')
```

Task 2: Gender Distribution

Objective: Visualize gender distribution in the dataset.

Steps:

1.Extract Gender Information: Identify and extract the gender column from the dataset.

2.Visualization: Create a simple visualization, such as a bar chart or pie chart, to represent the distribution of genders in the dataset.

```
x=data[['gender']]
print(x)
```

```
    gender
0   Female
1   Female
2     Male
```

```
3     Male
4   Female
5   Female
6   Female
7     Male
8     Male
9     Male
10  Female
11    Male
12  Female
13  Female
14  Female
15    Male
16  Female
17    Male
18    Male
19    Male
20    Male
21  Female
22    Male
23    Male
24  Female
25  Female
26    Male
27    Male
28    Male
29  Female
30    Male
31  Female
32    Male
33    Male
34    Male
35    Male
36    Male
37    Male
38    Male
39    Male
```

```
print(data['gender'].value_counts())
```

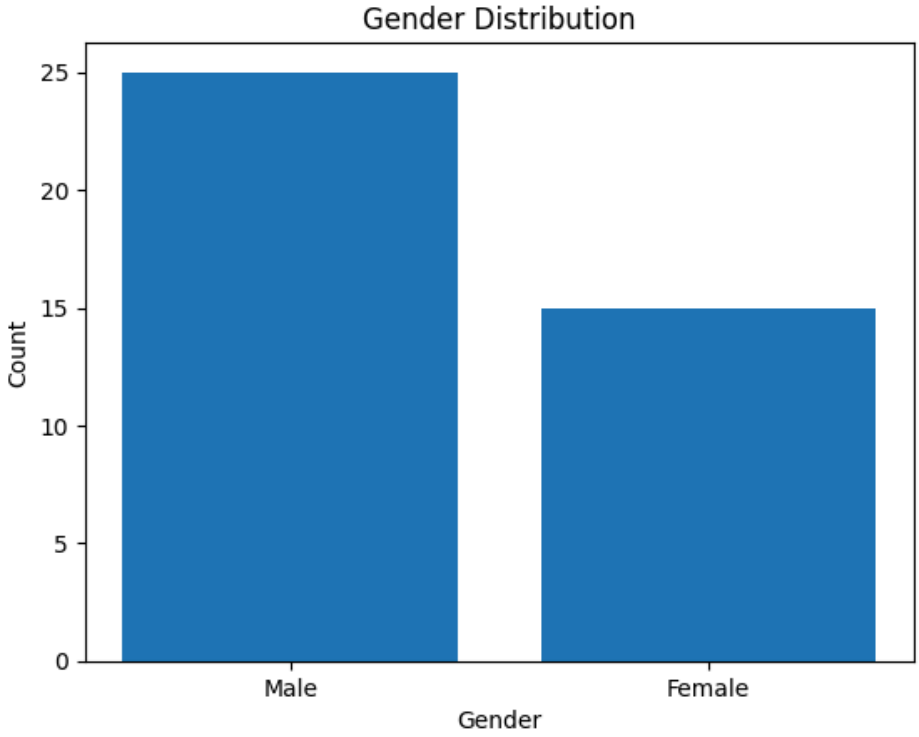
```
gender
Male      25
Female    15
Name: count, dtype: int64
```

```
# Assuming 'Gender' is the name of the column containing gender information
gender_column = data['gender']

import matplotlib.pyplot as plt

# Count the occurrences of each gender
gender_counts = gender_column.value_counts()

# Plot the bar chart
plt.bar(gender_counts.index, gender_counts.values)
plt.xlabel('Gender')
plt.ylabel('Count')
plt.title('Gender Distribution')
plt.show()
```



Task 3: Descriptive Statistics

Objective: Present basic statistics for numerical columns.

Steps:

1.Identify Numerical Columns: Review the dataset to identify columns containing numerical data (e.g., age, income).

2.Calculations: Use statistical functions (e.g., mean(), median(), std()) to calculate the mean, median, and standard deviation for each numerical colum

```
# Identify numerical columns
numerical_columns = data.select_dtypes(include=['number']).columns
numerical_columns
```

```
Index(['age', 'Mutual_Funds', 'Equity_Market', 'Debentures',
       'Government_Bonds', 'Fixed_Deposits', 'PPF', 'Gold'],
      dtype='object')
```

```
# Calculate statistics for each numerical column
statistics = data[numerical_columns].describe()

# Display the statistics
print(statistics)
```

```
              age  Mutual_Funds  Equity_Market  Debentures  Government_Bonds  \
count   40.000000     40.000000      40.000000   40.000000         40.000000
mean    27.800000      2.550000       3.475000    5.750000          4.650000
std      3.560467      1.197219       1.131994    1.675617          1.369072
min     21.000000      1.000000       1.000000    1.000000          1.000000
25%     25.750000      2.000000       3.000000    5.000000          4.000000
50%     27.000000      2.000000       4.000000    6.500000          5.000000
75%     30.000000      3.000000       4.000000    7.000000          5.000000
max     35.000000      7.000000       6.000000    7.000000          7.000000

       Fixed_Deposits        PPF       Gold
count       40.000000  40.000000  40.000000
mean         3.575000   2.025000   5.975000
std          1.795828   1.609069   1.143263
```

```
min          1.000000   1.000000   2.000000
25%          2.750000   1.000000   6.000000
50%          3.500000   1.000000   6.000000
75%          5.000000   2.250000   7.000000
max          7.000000   6.000000   7.000000
```

Task 4: Most Preferred Investment Avenue

Objective: Identify the most preferred investment avenue.

Steps:

1.Analyze Investment Avenues: Examine the column containing information about different investment avenues (e.g., equity, mutual funds).

2. Frequency Analysis: Determine the investment avenue with the highest frequency or occurrence.

```
#details pf investment avenue
investment_avenue_counts = data['Avenue'].value_counts()
investment_avenue_counts
```

```
Avenue
Mutual Fund              18
Equity                   10
Fixed Deposits            9
Public Provident Fund     3
Name: count, dtype: int64
```

```
# Get the most preferred investment avenue (the one with the highest frequency)
most_preferred_avenue = investment_avenue_counts.idxmax()
frequency_of_most_preferred = investment_avenue_counts.max()

print("Most preferred investment avenue:", most_preferred_avenue)
print("Frequency of most preferred investment avenue:", frequency_of_most_preferred)
```

```
Most preferred investment avenue: Mutual Fund
Frequency of most preferred investment avenue: 18
```

Task 6: Savings Objectives

Objective: Identify and present main savings objectives.

Steps:

1.Analyze Savings Objectives: Examine the column containing information about participants' savings objectives.

2.List and Describe Objectives: Create a list and describe the main savings objectives mentioned by participant

```
#details pf saving objective
savingobjective_counts = data['What are your savings objectives?'].value_counts()
savingobjective_counts
```

```
What are your savings objectives?
Retirement Plan    24
Health Care        13
Education           3
Name: count, dtype: int64
```

```
  # Define savings objectives
savings_objectives = {
    "24": {
        "objective": "Retirement Plan",
    },
    "13": {
        "objective": "Health Care",
    },
    "3": {
        "objective": "Education",
    }
}

# Function to display objectives
def display_savings_objectives(objectives):
    for key, value in objectives.items():
        print(f"Objective {key}: {value['objective']}")

# Display savings objectives
display_savings_objectives(savings_objectives)
```

```
Objective 24: Retirement Plan
Objective 13: Health Care
Objective 3: Education
```

Task 7: Common Information Sources

Objective: Analyze common sources participants rely on for investment information.

Steps:

1.Explore Information Sources Column: Review the column where participants indicated their sources of investment information.

2.Identify Common Sources: Analyze the data to identify and summarize the most common sources participants rely on.

```
infsourc = data[['Source']]
print(infsourc)
```

```
                      Source
0   Newspapers and Magazines
1      Financial Consultants
2                 Television
3                   Internet
4                   Internet
5                   Internet
6      Financial Consultants
7   Newspapers and Magazines
8                 Television
9   Newspapers and Magazines
10     Financial Consultants
11     Financial Consultants
12                  Internet
13  Newspapers and Magazines
14     Financial Consultants
15  Newspapers and Magazines
16                Television
17     Financial Consultants
18  Newspapers and Magazines
19  Newspapers and Magazines
20     Financial Consultants
21  Newspapers and Magazines
22     Financial Consultants
23     Financial Consultants
24  Newspapers and Magazines
```

```
25     Financial Consultants
26     Financial Consultants
27                Television
28                Television
29  Newspapers and Magazines
30                Television
31     Financial Consultants
32  Newspapers and Magazines
33  Newspapers and Magazines
34     Financial Consultants
35     Financial Consultants
36  Newspapers and Magazines
37     Financial Consultants
38  Newspapers and Magazines
39     Financial Consultants
```

```
z=data['Source'].value_counts()
z
```

```
Source
Financial Consultants       16
Newspapers and Magazines    14
Television                   6
Internet                     4
Name: count, dtype: int64
```

```
most_commonsource = z.idxmax()
most_commonsource
```

```
'Financial Consultants'
```

Task 9: Expectations from Investments

Objective: Summarize participants' expectations from investments.

Steps:

1.Explore Expectations Column: Review the column where participants provided information about their expectations.

2.List and Describe Expectations: Create a list and describe the common expectations mentioned by participants.

```
expect=data[['Expect']]
print(expect)
```

```
      Expect
0   20%-30%
1   20%-30%
2   20%-30%
3   10%-20%
4   20%-30%
5   30%-40%
6   20%-30%
7   20%-30%
8   20%-30%
9   30%-40%
10  20%-30%
11  20%-30%
12  20%-30%
13  20%-30%
14  20%-30%
15  20%-30%
16  20%-30%
17  20%-30%
18  20%-30%
19  20%-30%
20  20%-30%
```

```
21  20%-30%
22  20%-30%
23  20%-30%
24  30%-40%
25  20%-30%
26  20%-30%
27  20%-30%
28  20%-30%
29  10%-20%
30  10%-20%
31  20%-30%
32  30%-40%
33  20%-30%
34  20%-30%
35  20%-30%
36  20%-30%
37  30%-40%
38  20%-30%
39  20%-30%
```

```
import pandas as pd
data = pd.DataFrame({
    'Expectations': [
        '20%-30%', '20%-30%', '20%-30%', '10%-20%', '20%-30%', '30%-40%', '20%-30%', '20%-30%', '20%-30%', '
        '20%-30%', '20%-30%', '20%-30%', '20%-30%', '20%-30%', '20%-30%', '20%-30%', '20%-30%', '20%-30%', '
        '20%-30%', '20%-30%', '20%-30%', '20%-30%', '30%-40%', '20%-30%', '20%-30%', '20%-30%', '20%-30%', '
        '10%-20%', '20%-30%', '30%-40%', '20%-30%', '20%-30%', '20%-30%', '20%-30%', '30%-40%', '20%-30%', '
    ]
})

# Count occurrences of each expectation range
expectations_counts = data['Expectations'].value_counts()

# Display the common expectations mentioned by participants
print("Common expectations from investments:")
for expectation, count in expectations_counts.items():
    print(f"{expectation}: {count} participants")
```

```
Common expectations from investments:
20%-30%: 32 participants
30%-40%: 5 participants
10%-20%: 3 participants
```