Investment Avenue Analysis

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

1. Data Overview

```
df = pd.read_csv("Financial Dataset.csv")
df.head(5)
```

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

   Government_Bonds  Fixed_Deposits  PPF  Gold  ...           Duration
\
0                 3               7    6     4  ...          1-3 years

1                 1               5    6     7  ...  More than 5 years

2                 2               5    1     7  ...          3-5 years

3                 7               6    4     5  ...   Less than 1 year

4                 6               4    5     7  ...   Less than 1 year


  Invest_Monitor   Expect       Avenue What are your savings
objectives?  \
0        Monthly  20%-30%  Mutual Fund                  Retirement
Plan
1         Weekly  20%-30%  Mutual Fund                      Health
Care
2          Daily  20%-30%       Equity                  Retirement
Plan
3          Daily  10%-20%       Equity                  Retirement
Plan
```

```
4          Daily  20%-30%       Equity                     Retirement
Plan

          Reason_Equity         Reason_Mutual     Reason_Bonds  \
0  Capital Appreciation        Better Returns  Safe Investment
1             Dividend        Better Returns  Safe Investment
2  Capital Appreciation          Tax Benefits  Assured Returns
3             Dividend  Fund Diversification   Tax Incentives
4  Capital Appreciation        Better Returns  Safe Investment

             Reason_FD                      Source
0        Fixed Returns  Newspapers and Magazines
1  High Interest Rates     Financial Consultants
2        Fixed Returns                Television
3  High Interest Rates                  Internet
4            Risk Free                  Internet

[5 rows x 24 columns]
```

```
print(df.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')
```

```
df.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
```

---

2. Gender Distribution

To extract gender column

```
gender_data = df['gender']
gender_data
```

```
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
```

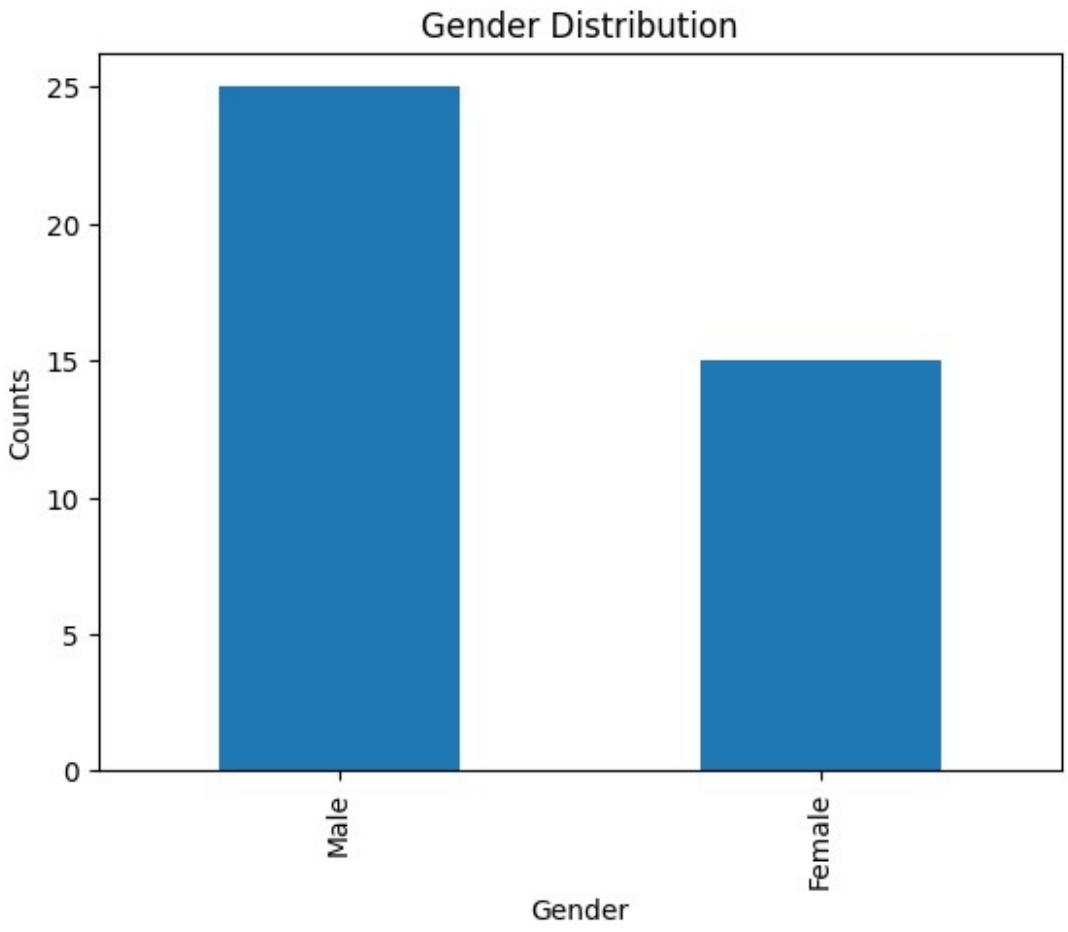
```
24    Female
25    Female
26      Male
27      Male
28      Male
29    Female
30      Male
31    Female
32      Male
33      Male
34      Male
35      Male
36      Male
37      Male
38      Male
39      Male
Name: gender, dtype: object
```

No. of patrticipants based on gender

```
gender_count = gender_data.value_counts()
gender_count
```

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

```
gender_count.plot(kind="bar")
plt.title('Gender Distribution')
plt.xlabel('Gender')
plt.ylabel('Counts')
plt.show()
```



3.Descriptive Statistics

```
numerical_columns = df.select_dtypes('int64').columns
print("Numerical Columns:", numerical_columns)
```

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

```
df[numerical_columns].describe()
```

```
             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
```

---

4. Most Preferred Investment Avenue

```
investment_cloumns = ['Mutual_Funds', 'Equity_Market', 'Debentures',
       'Government_Bonds', 'Fixed_Deposits', 'PPF', 'Gold']

avenue_counts = {}

for col in investment_cloumns:
    avenue_counts[col] = df[col].sum()

print("Each investment avenue")
for avenue, count in avenue_counts.items():
    print(f"{avenue}: {count}")
```

```
Each investment avenue
Mutual_Funds: 102
Equity_Market: 139
Debentures: 230
Government_Bonds: 186
Fixed_Deposits: 143
PPF: 81
Gold: 239
```

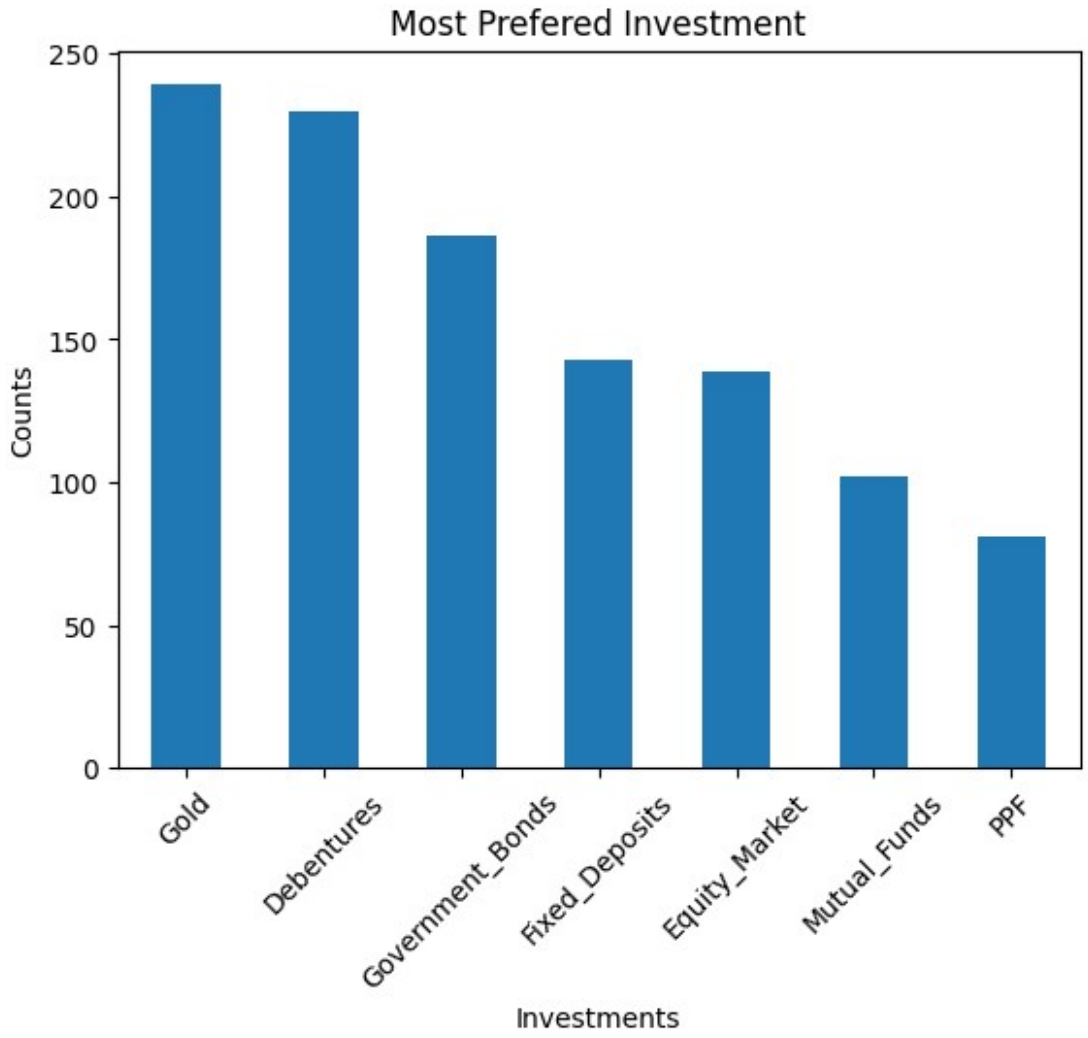
```
most_prefered = max(avenue_counts,key=avenue_counts.get)
print(f"Most Prefered Investment Avenue: {most_prefered}
({avenue_counts[most_prefered]})")

Most Prefered Investment Avenue: Gold (239)
```

Convert dictionery to series to show bar chart

```
avenue_series = pd.Series(avenue_counts)
avenue_series_sorted = avenue_series.sort_values(ascending= False)

avenue_series_sorted.plot(kind="bar")
plt.title("Most Prefered Investment")
plt.xlabel("Investments")
plt.ylabel("Counts")
plt.xticks(rotation=45)
plt.show()
```



---

5. Reasons for Investment

```
print(df[['Reason_Equity', 'Reason_Mutual',
       'Reason_Bonds', 'Reason_FD']].head())
```

```
          Reason_Equity         Reason_Mutual     Reason_Bonds  \
0  Capital Appreciation        Better Returns  Safe Investment
1             Dividend        Better Returns  Safe Investment
2  Capital Appreciation          Tax Benefits  Assured Returns
3             Dividend  Fund Diversification   Tax Incentives
4  Capital Appreciation        Better Returns  Safe Investment

             Reason_FD
0        Fixed Returns
1  High Interest Rates
2        Fixed Returns
3  High Interest Rates
4            Risk Free
```

For exploring reasons columns and to see unique reasons

```
for columns in ['Reason_Equity', 'Reason_Mutual', 'Reason_Bonds',
'Reason_FD']:
    print(f"Unique reasons for {columns}:")
    print(df[columns].unique())
```

```
Unique reasons for Reason_Equity:
['Capital Appreciation' 'Dividend' 'Liquidity']
Unique reasons for Reason_Mutual:
['Better Returns' 'Tax Benefits' 'Fund Diversification']
Unique reasons for Reason_Bonds:
['Safe Investment' 'Assured Returns' 'Tax Incentives']
Unique reasons for Reason_FD:
['Fixed Returns' 'High Interest Rates' 'Risk Free']
```

To see the purpose of investment

```
for columns in ['Reason_Equity', 'Reason_Mutual', 'Reason_Bonds',
'Reason_FD']:

    print(f"Top reasons for {columns}:")
    print(df[columns].value_counts())
```

```
Top reasons for Reason_Equity:
Reason_Equity
Capital Appreciation    30
Dividend                 8
Liquidity                2
```

```
Name: count, dtype: int64
Top reasons for Reason_Mutual:
Reason_Mutual
Better Returns          24
Fund Diversification    13
Tax Benefits             3
Name: count, dtype: int64
Top reasons for Reason_Bonds:
Reason_Bonds
Assured Returns    26
Safe Investment    13
Tax Incentives      1
Name: count, dtype: int64
Top reasons for Reason_FD:
Reason_FD
Risk Free              19
Fixed Returns          18
High Interest Rates     3
Name: count, dtype: int64
```

Top Reasons for Investment Choices

Equity:

Investors primarily choose equity markets for capital appreciation, aiming to grow their wealth over the long term through higher returns.

Mutual Funds:

Mutual funds are favored for delivering better returns while offering diversification.

Bonds:

Bonds are mainly selected for their assured returns, providing a sense of safety.

Fixed Deposits:

Fixed deposits attract investors who prefer risk-free options and fixed returns, ensuring capital protection and guaranteed interest.

---

6. Savings Objectives

```
print(df["What are your savings objectives?"].tail())
```

```
35        Health Care
36    Retirement Plan
37        Health Care
38        Health Care
39    Retirement Plan
Name: What are your savings objectives?, dtype: object
```

```
objectives_counts = df["What are your savings objectives?"].
value_counts()
print(objectives_counts)
```

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

Main Savings Objective

Retirement Plan:

The majority of participants are saving to ensure a financially secure and independent life after retirement.

---

7. Common Information Sources

```
print(df["Source"].head())
```

```
0    Newspapers and Magazines
1       Financial Consultants
2                  Television
3                    Internet
4                    Internet
Name: Source, dtype: object
```

```
Source_counts = df["Source"].value_counts()
print(Source_counts)
```

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

The most common sources participants rely on for investment information are financial consultants, who provide expert advice, and newspapers and magazines, which keep them updated on market trends and financial news.

***

8. Investment Duration

```
print(df["Duration"].unique())
print(df["Duration"].dtype)
```

```
['1-3 years' 'More than 5 years' '3-5 years' 'Less than 1 year']
object
```

The "Duration" column contains text datatype, so it is converted to numeric values to calculate the average duration accurately.

```
Duration_mapping = {
    '1-3 years': 2.0,
    'More than 5 years':6.0,
    '3-5 years':4.0,
    'Less than 1 year':0.5
}

df['Duration_num'] = df['Duration'].map(Duration_mapping)
print(df[['Duration','Duration_num']].head())
```

```
            Duration  Duration_num
0          1-3 years           2.0
1  More than 5 years           6.0
2          3-5 years           4.0
3   Less than 1 year           0.5
4   Less than 1 year           0.5
```

```
Duration_average = df['Duration_num'].mean()
print(f"The average investment duration is approximately
{Duration_average:.1f} years.")
```

```
The average investment duration is approximately 3.0 years.
```

---

9. Expectations from Investments

```
df['Expect'].head()
```

```
0    20%-30%
1    20%-30%
2    20%-30%
3    10%-20%
4    20%-30%
Name: Expect, dtype: object
```

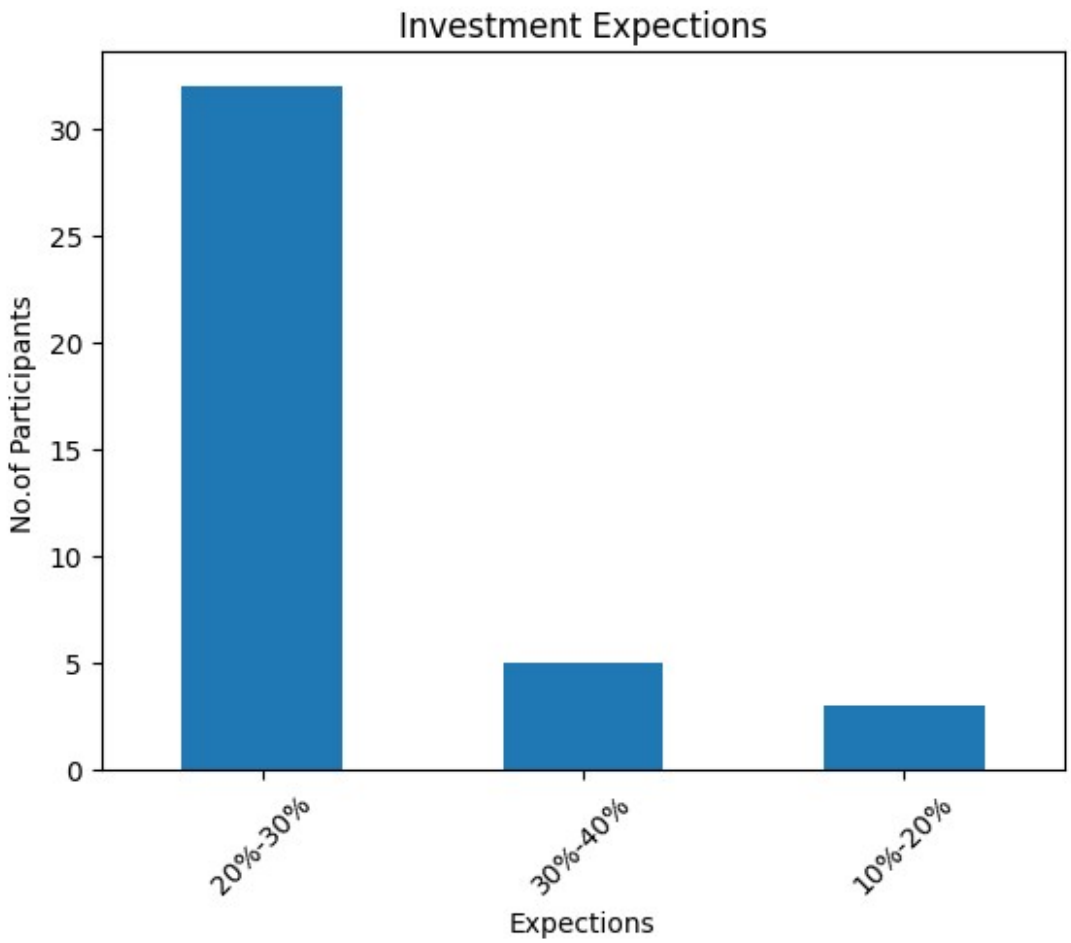
```
df['Expect'].unique()
```

```
array(['20%-30%', '10%-20%', '30%-40%'], dtype=object)
```

```
Expections_counts = df['Expect'].value_counts()
print(Expections_counts)
```

```
Expect
20%-30%    32
30%-40%     5
10%-20%     3
Name: count, dtype: int64
```

```
Expections_counts.plot(kind="bar")
plt.title("Investment Expections")
plt.xlabel("Expections")
plt.ylabel("No.of Participants")
plt.xticks(rotation=45)
plt.show()
```



---

10. Correlation Analysis

```
print(df[['age','Duration_num','Expect']].head())
```

```
   age  Duration_num   Expect
0   34           2.0  20%-30%
1   23           6.0  20%-30%
2   30           4.0  20%-30%
3   22           0.5  10%-20%
4   24           0.5  20%-30%
```

```
print(df[['age','Duration_num','Expect']].dtypes)
```

```
age               int64
Duration_num    float64
Expect           object
dtype: object
```

Convert "Expect" column datatype for correlation analysis

```
expect_mapping = {
    '10%-20%': 15,
    '20%-30%': 25,
    '30%-40%': 35
}
df['Expect_Num'] = df['Expect'].map(expect_mapping)

print(df[['age','Duration_num','Expect_Num']].dtypes)
```

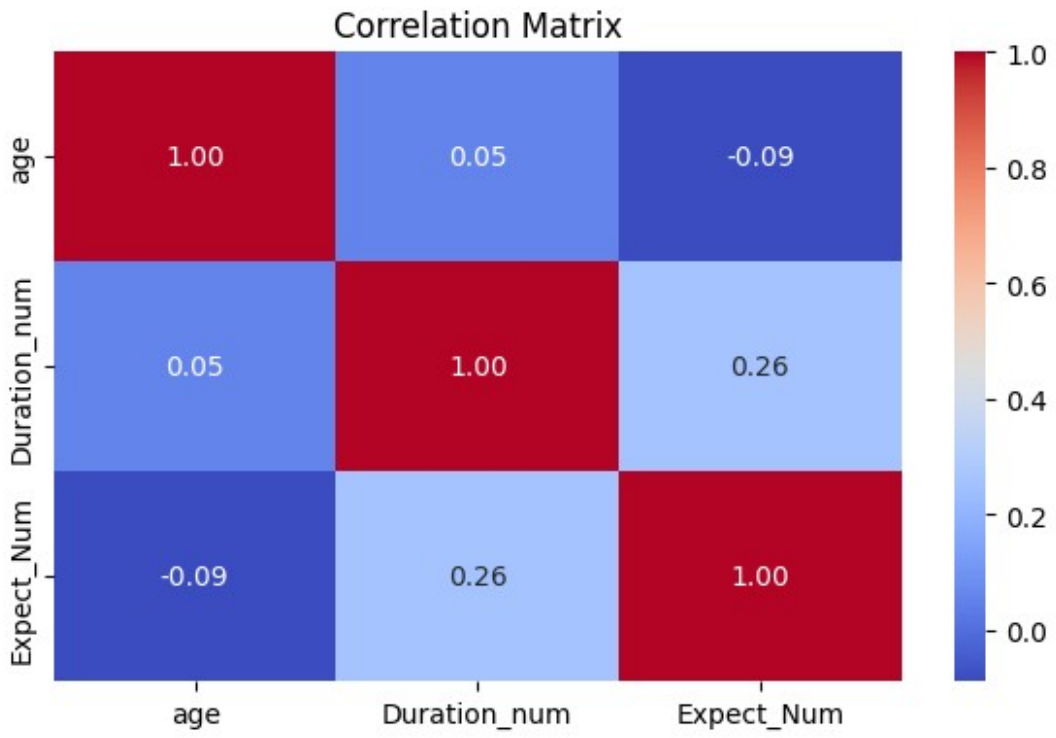
```
age               int64
Duration_num    float64
Expect_Num        int64
dtype: object
```

```
correlation_matrix = df[['age','Duration_num','Expect_Num']].corr()
print("Correlation Matrix:")
print(correlation_matrix)
```

```
Correlation Matrix:
                   age  Duration_num  Expect_Num
age           1.000000      0.051756   -0.089606
Duration_num  0.051756      1.000000    0.258223
Expect_Num   -0.089606      0.258223    1.000000
```

Heatmap

```
plt.figure(figsize=(6, 4))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm',
fmt=".2f")
plt.title("Correlation Matrix")
plt.tight_layout()
plt.show()
```


Correlation Matrix
age
Duration_num
Expect_Num
1.00
0.05
-0.09
0.05
1.00
0.26
-0.09
0.26
1.00
age
Duration_num
Expect_Num
1.0
0.8
0.6
0.4
0.2
0.0