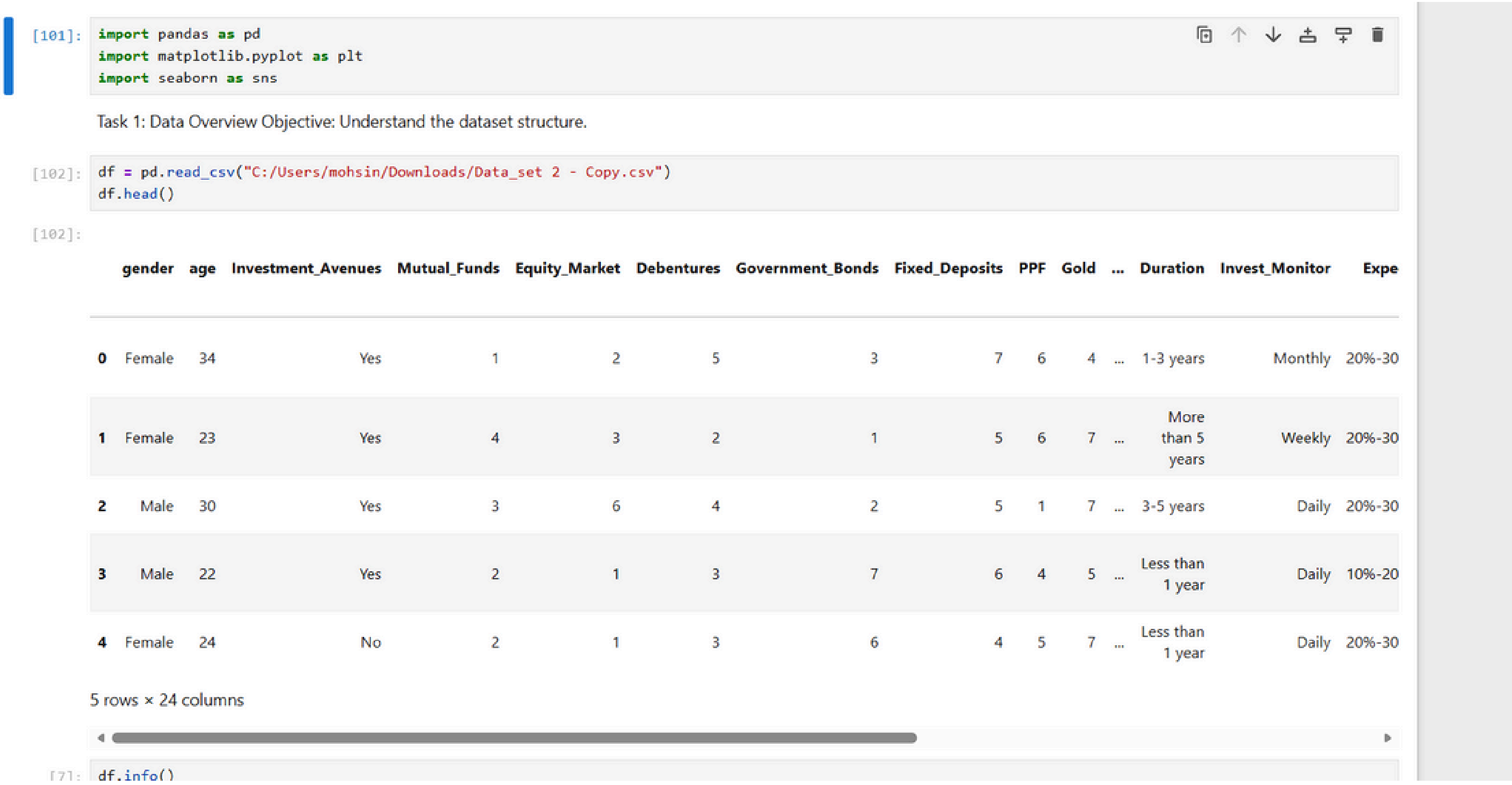
link of jupyter notebook project

http://localhost:8888/notebooks/internship%20task-Copy1.ipynb



```
[101]: import pandas as pd
       import matplotlib.pyplot as plt
       import seaborn as sns
```

Task 1: Data Overview Objective: Understand the dataset structure.

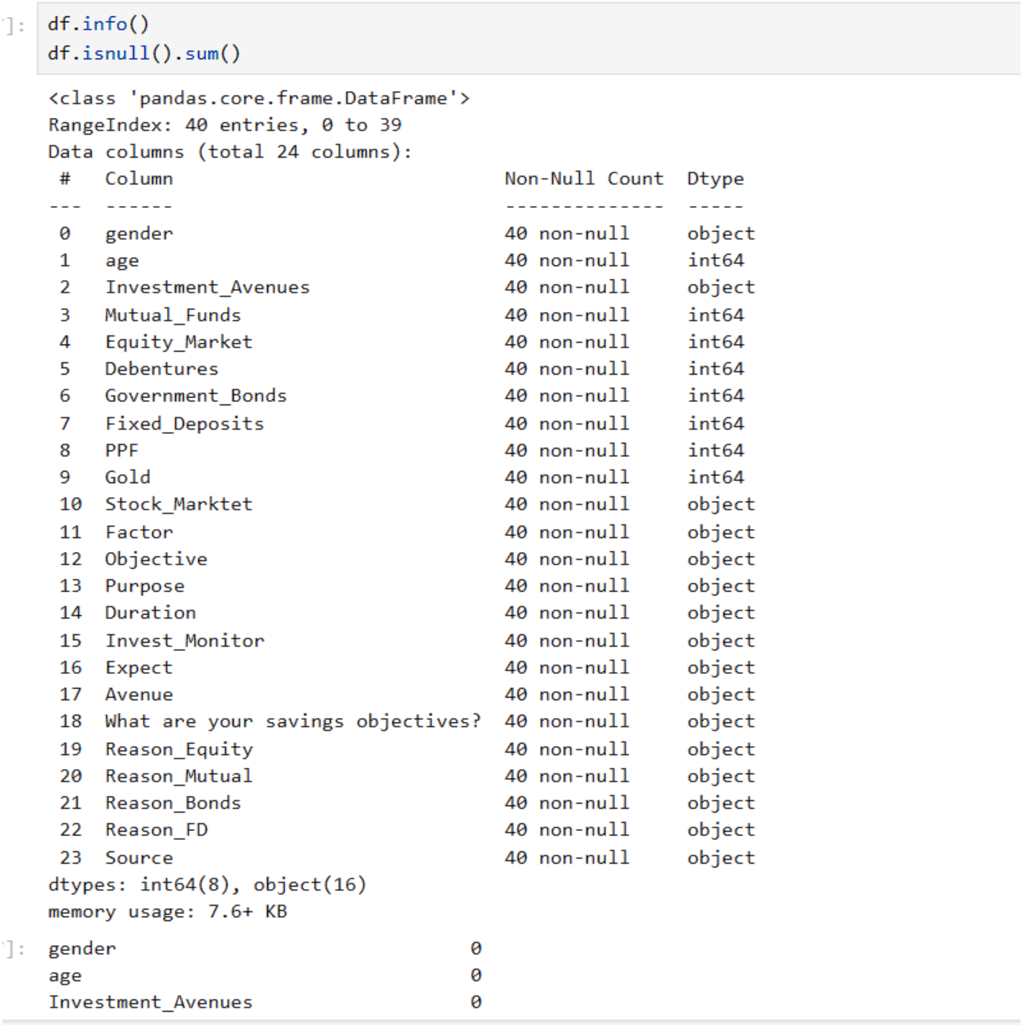
```
[102]: df = pd.read_csv("C:/Users/mohsin/Downloads/Data_set 2 - Copy.csv")
       df.head()
```

[102]:

	gender	age	Investment_Avenues	Mutual_Funds	Equity_Market	Debentures	Government_Bonds	Fixed_Deposits	PPF	Gold	...	Duration	Invest_Monitor	Expe
0	Female	34	Yes	1	2	5	3	7	6	4	...	1-3 years	Monthly	20%-30
1	Female	23	Yes	4	3	2	1	5	6	7	...	More than 5 years	Weekly	20%-30
2	Male	30	Yes	3	6	4	2	5	1	7	...	3-5 years	Daily	20%-30
3	Male	22	Yes	2	1	3	7	6	4	5	...	Less than 1 year	Daily	10%-20
4	Female	24	No	2	1	3	6	4	5	7	...	Less than 1 year	Daily	20%-30

5 rows × 24 columns

```
[7]: df.info()
```



```
]: df.info()
   df.isnull().sum()
```

```
<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
```

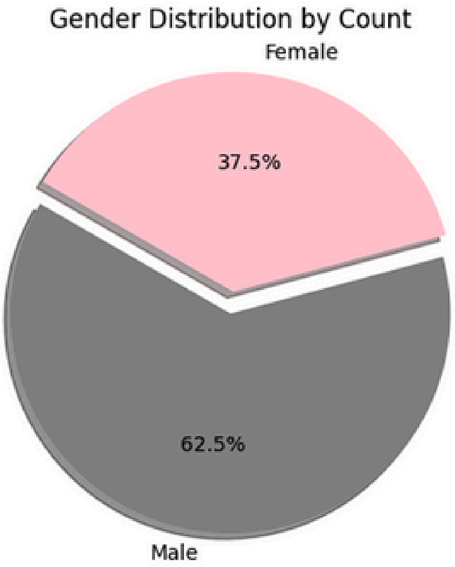
```
]: gender                               0
   age                                  0
   Investment_Avenues                   0
```

```
     memory usage: 7.6+ KB

[7]: gender                               0
     age                                  0
     Investment_Avenues                   0
     Mutual_Funds                         0
     Equity_Market                        0
     Debentures                           0
     Government_Bonds                     0
     Fixed_Deposits                       0
     PPF                                  0
     Gold                                 0
     Stock_Marktet                        0
     Factor                               0
     Objective                            0
     Purpose                              0
     Duration                             0
     Invest_Monitor                       0
     Expect                               0
     Avenue                               0
     What are your savings objectives?    0
     Reason_Equity                        0
     Reason_Mutual                        0
     Reason_Bonds                         0
     Reason_FD                            0
     Source                               0
     dtype: int64
```

Task 2: Gender Distribution Objective: Visualize gender distribution in the dataset.

[8]:

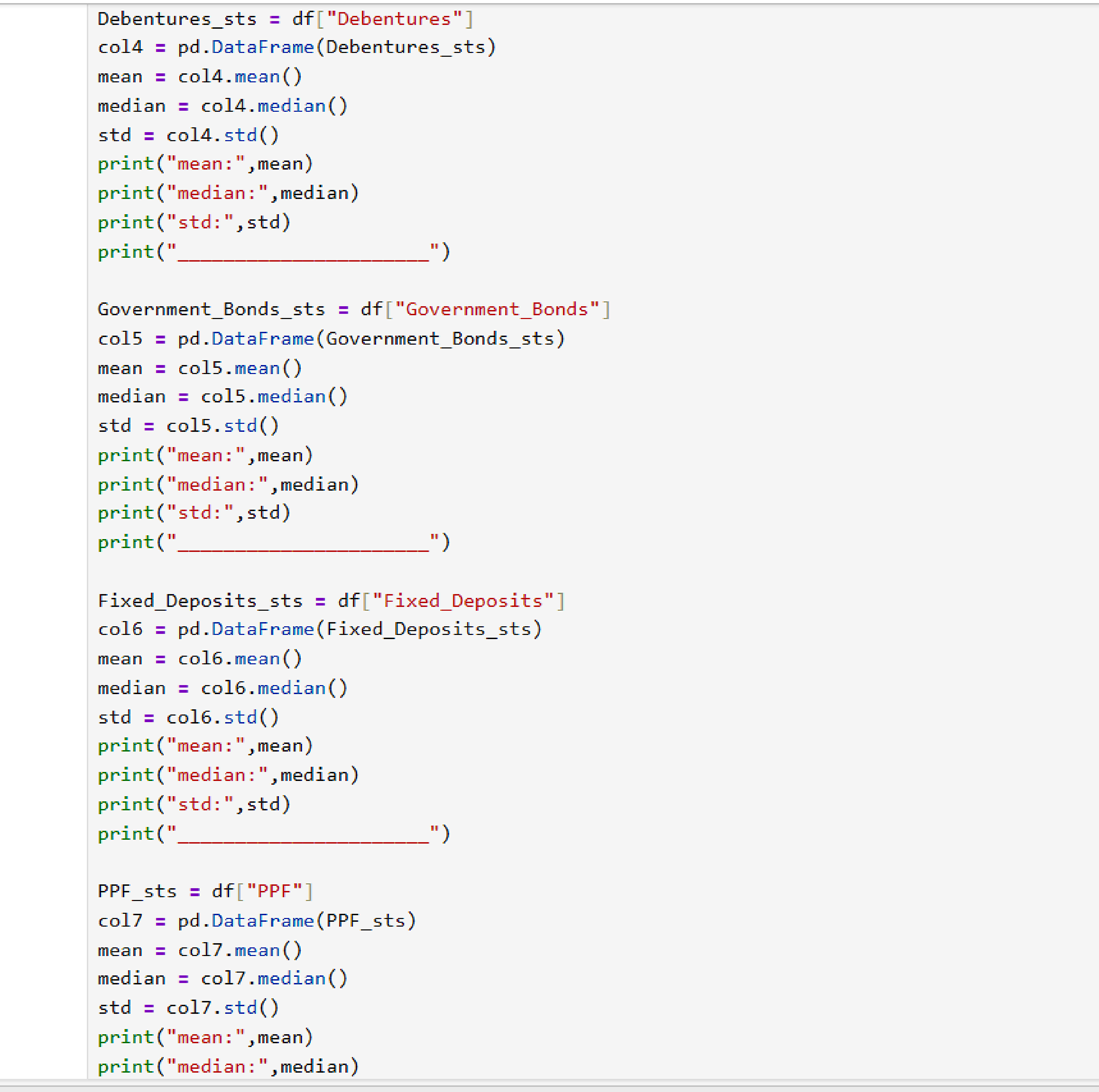
```
gb = df["gender"].value_counts()
color = ["gray","pink"]
explode = (0,00.1)
plt.pie(gb.values,labels = gb.index,startangle = 150 , autopct = "%1.1f%%",colors = color,shadow = True,explode = explode )
plt.title("Gender Distribution by Count")
plt.figure(figsize = (6,6))
plt.show()
```



```
<Figure size 600x600 with 0 Axes>
```

```
Task 3: Descriptive Statistics
Objective: Present basic statistics for
numerical columns.
```

```
age_mean = df["age"]
col = pd.DataFrame(age_mean)
mean = col.mean()
median = col.median()
std = col.std()
print("mean:",mean)
print("median:",median)
print("std:",std)
print(f"______________________")

mutual_sts = df["Mutual_Funds"]
col2 = pd.DataFrame(mutual_sts)
mean = col2.mean()
median = col2.median()
std = col2.std()
print("mean:",mean)
print("median:",median)
print("std:",std)
print("______________________")

Equity_Market_sts = df["Equity_Market"]
col3 = pd.DataFrame(Equity_Market_sts)
mean = col3.mean()
median = col3.median()
std = col3.std()
print("mean:",mean)
print("median:",median)
print("std:",std)
print("______________________")

Debentures_sts = df["Debentures"]
col4 = pd.DataFrame(Debentures_sts)
```



```
Debentures_sts = df["Debentures"]
col4 = pd.DataFrame(Debentures_sts)
mean = col4.mean()
median = col4.median()
std = col4.std()
print("mean:",mean)
print("median:",median)
print("std:",std)
print("______________________")

Government_Bonds_sts = df["Government_Bonds"]
col5 = pd.DataFrame(Government_Bonds_sts)
mean = col5.mean()
median = col5.median()
std = col5.std()
print("mean:",mean)
print("median:",median)
print("std:",std)
print("______________________")

Fixed_Deposits_sts = df["Fixed_Deposits"]
col6 = pd.DataFrame(Fixed_Deposits_sts)
mean = col6.mean()
median = col6.median()
std = col6.std()
print("mean:",mean)
print("median:",median)
print("std:",std)
print("______________________")

PPF_sts = df["PPF"]
col7 = pd.DataFrame(PPF_sts)
mean = col7.mean()
median = col7.median()
std = col7.std()
print("mean:",mean)
print("median:",median)
print("median:",median)
print("std:",std)
print("______________________")

Gold_sts = df["Gold"]
col8 = pd.DataFrame(Gold_sts)
mean = col8.mean()
median = col8.median()
std = col8.std()
print("mean:",mean)
print("median:",median)
print("std:",std)
print("______________________")
```

```
mean: age    27.8
dtype: float64
median: age    27.0
dtype: float64
std: age    3.560467
dtype: float64
______________________
mean: Mutual_Funds    2.55
dtype: float64
median: Mutual_Funds    2.0
dtype: float64
std: Mutual_Funds    1.197219
dtype: float64
______________________
mean: Equity_Market    3.475
dtype: float64
median: Equity_Market    4.0
dtype: float64
std: Equity_Market    1.131994
dtype: float64
______________________
mean: Debentures    5.75
dtype: float64
median: Debentures    6.5
dtype: float64
```

```
std: Equity_Market    1.131994
dtype: float64

______________________
mean: Debentures    5.75
dtype: float64
median: Debentures    6.5
dtype: float64
std: Debentures    1.675617
dtype: float64

______________________
mean: Government_Bonds    4.65
dtype: float64
median: Government_Bonds    5.0
dtype: float64
std: Government_Bonds    1.369072
dtype: float64

______________________
mean: Fixed_Deposits    3.575
dtype: float64
median: Fixed_Deposits    3.5
dtype: float64
std: Fixed_Deposits    1.795828
dtype: float64

______________________
mean: PPF    2.025
dtype: float64
median: PPF    1.0
dtype: float64
std: PPF    1.609069
dtype: float64

______________________
mean: Gold    5.975
dtype: float64
median: Gold    6.0
dtype: float64
std: Gold    1.143263
dtype: float64
```

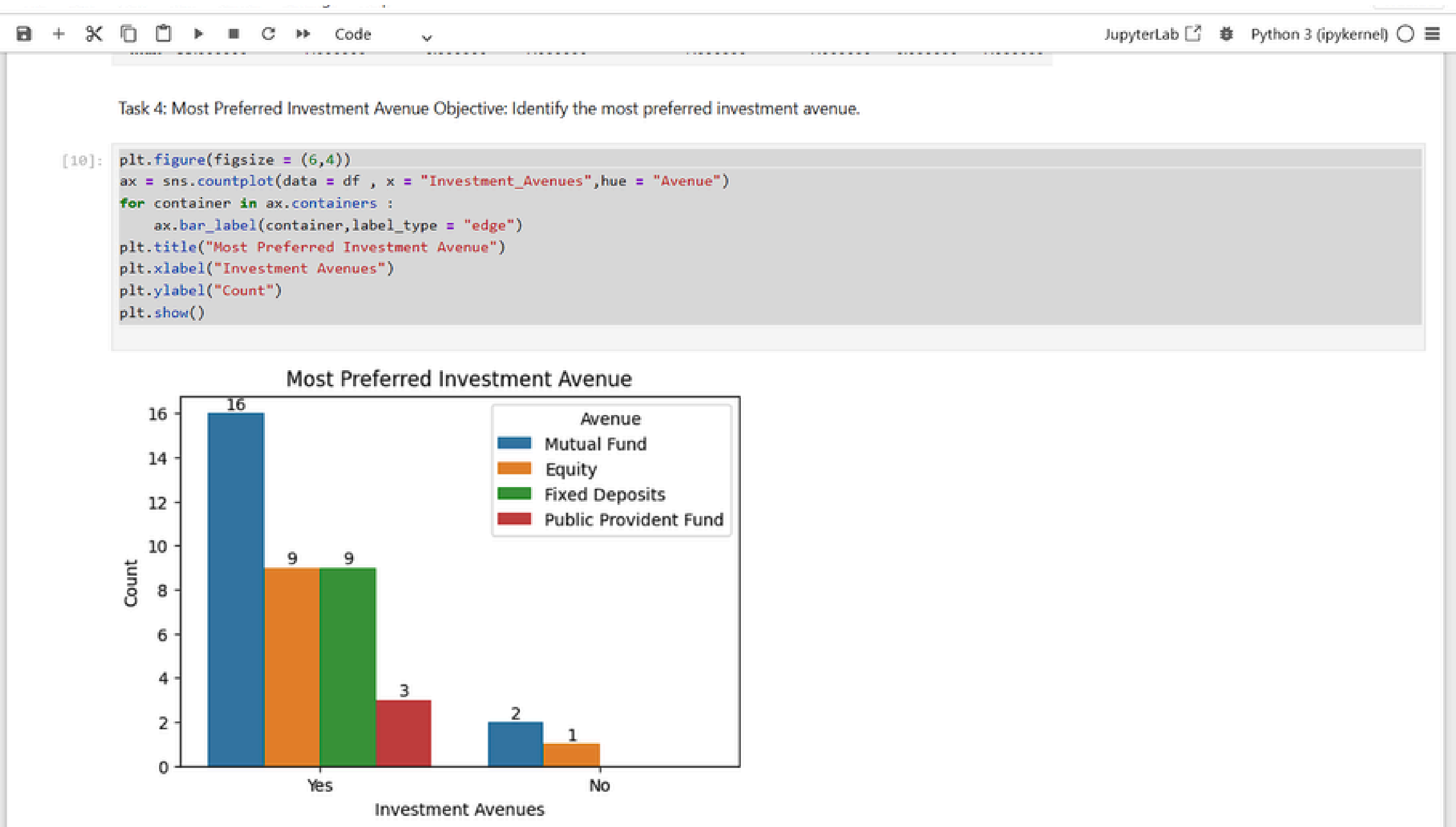
[58]:

```
df.describe()
```

[58]:

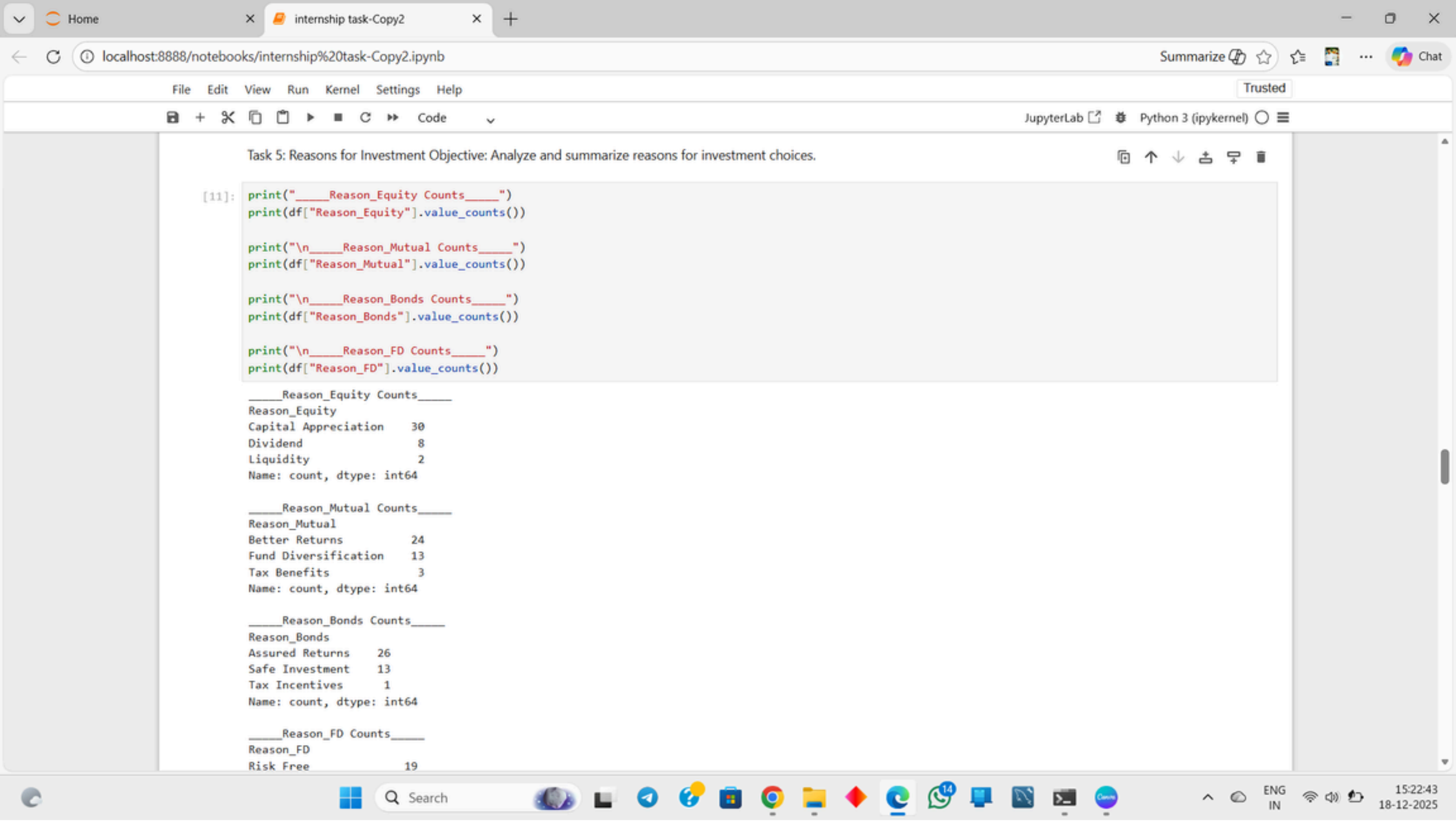
	age	Mutual_Funds	Equity_Market	Debentures	Government_Bonds	Fixed_Deposits	PPF	Gold
count	40.000000	40.000000	40.000000	40.000000	40.000000	40.000000	40.000000	40.000000
mean	27.800000	2.550000	3.475000	5.750000	4.650000	3.575000	2.025000	5.975000
std	3.560467	1.197219	1.131994	1.675617	1.369072	1.795828	1.609069	1.143263
min	21.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	2.000000
25%	25.750000	2.000000	3.000000	5.000000	4.000000	2.750000	1.000000	6.000000
50%	27.000000	2.000000	4.000000	6.500000	5.000000	3.500000	1.000000	6.000000
75%	30.000000	3.000000	4.000000	7.000000	5.000000	5.000000	2.250000	7.000000
max	35.000000	7.000000	6.000000	7.000000	7.000000	7.000000	6.000000	7.000000

Task 4: Most Preferred Investment Avenue Objective: Identify the most preferred investment avenue.



Task 4: Most Preferred Investment Avenue Objective: Identify the most preferred investment avenue.

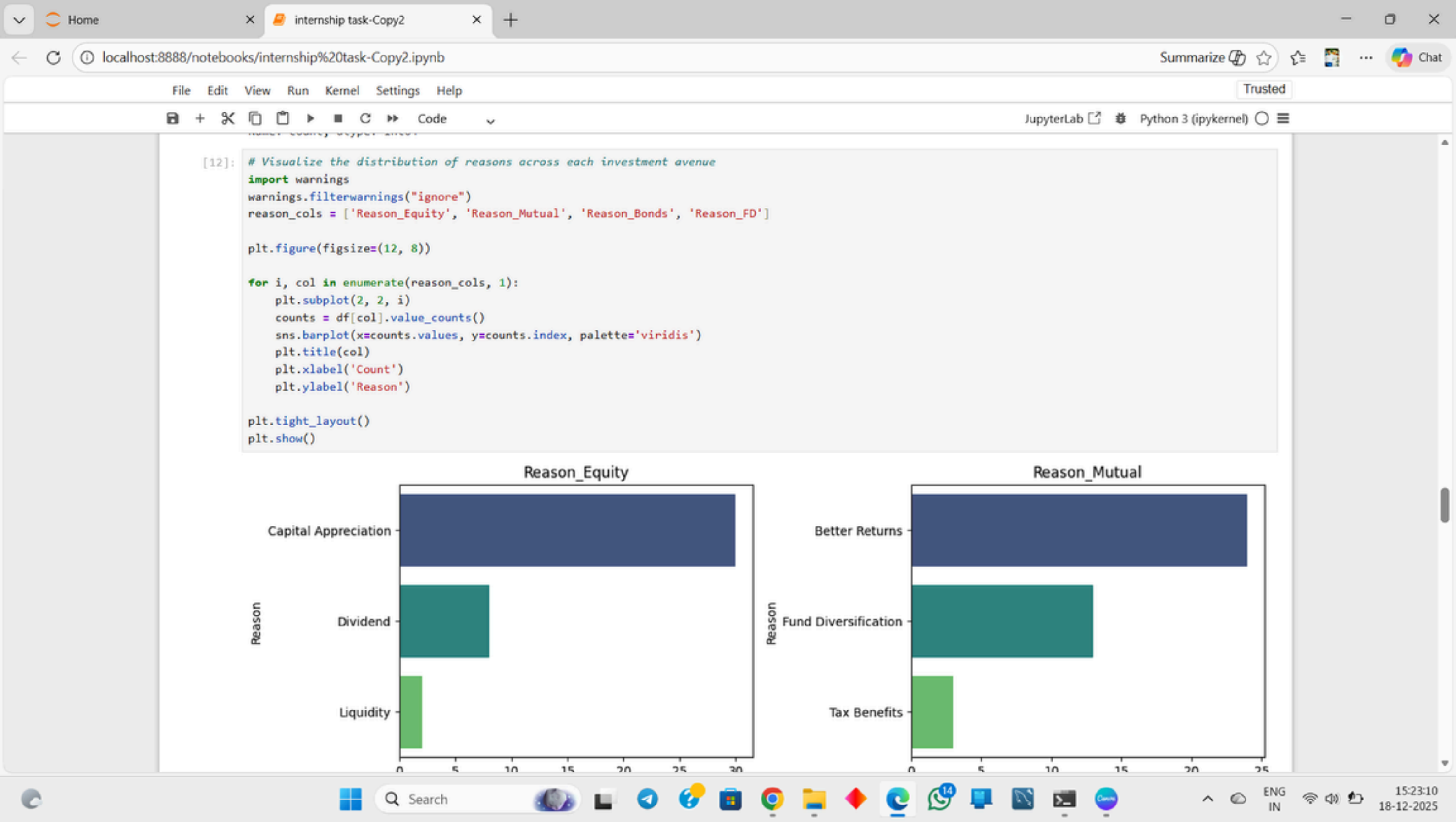
```
plt.figure(figsize = (6,4))
ax = sns.countplot(data = df , x = "Investment_Avenues",hue = "Avenue")
for container in ax.containers :
    ax.bar_label(container,label_type = "edge")
plt.title("Most Preferred Investment Avenue")
plt.xlabel("Investment Avenues")
plt.ylabel("Count")
plt.show()
```



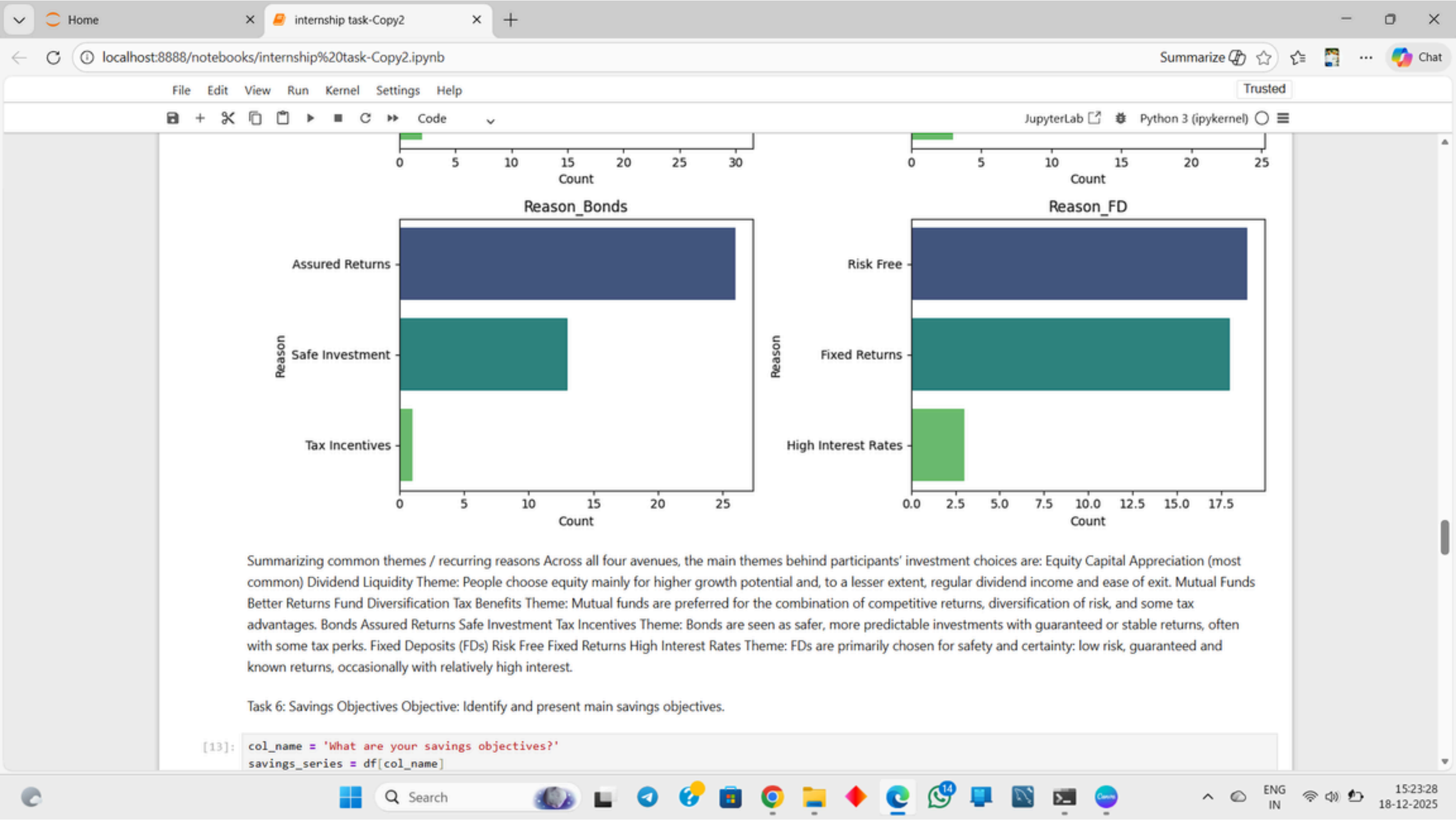
Task 5: Reasons for Investment Objective: Analyze and summarize reasons for investment choices.

```
print("_____Reason_Equity Counts_____")
print(df["Reason_Equity"].value_counts())

print("\n_____Reason_Mutual Counts_____")
print(df["Reason_Mutual"].value_counts())

print("\n_____Reason_Bonds Counts_____")
print(df["Reason_Bonds"].value_counts())

print("\n_____Reason_FD Counts_____")
print(df["Reason_FD"].value_counts())
```

```
_____Reason_Equity Counts_____
Reason_Equity
Capital Appreciation    30
Dividend                 8
Liquidity                2
Name: count, dtype: int64

_____Reason_Mutual Counts_____
Reason_Mutual
Better Returns          24
Fund Diversification    13
Tax Benefits             3
Name: count, dtype: int64

_____Reason_Bonds Counts_____
Reason_Bonds
Assured Returns    26
Safe Investment    13
Tax Incentives      1
Name: count, dtype: int64

_____Reason_FD Counts_____
Reason_FD
Risk Free          19
```



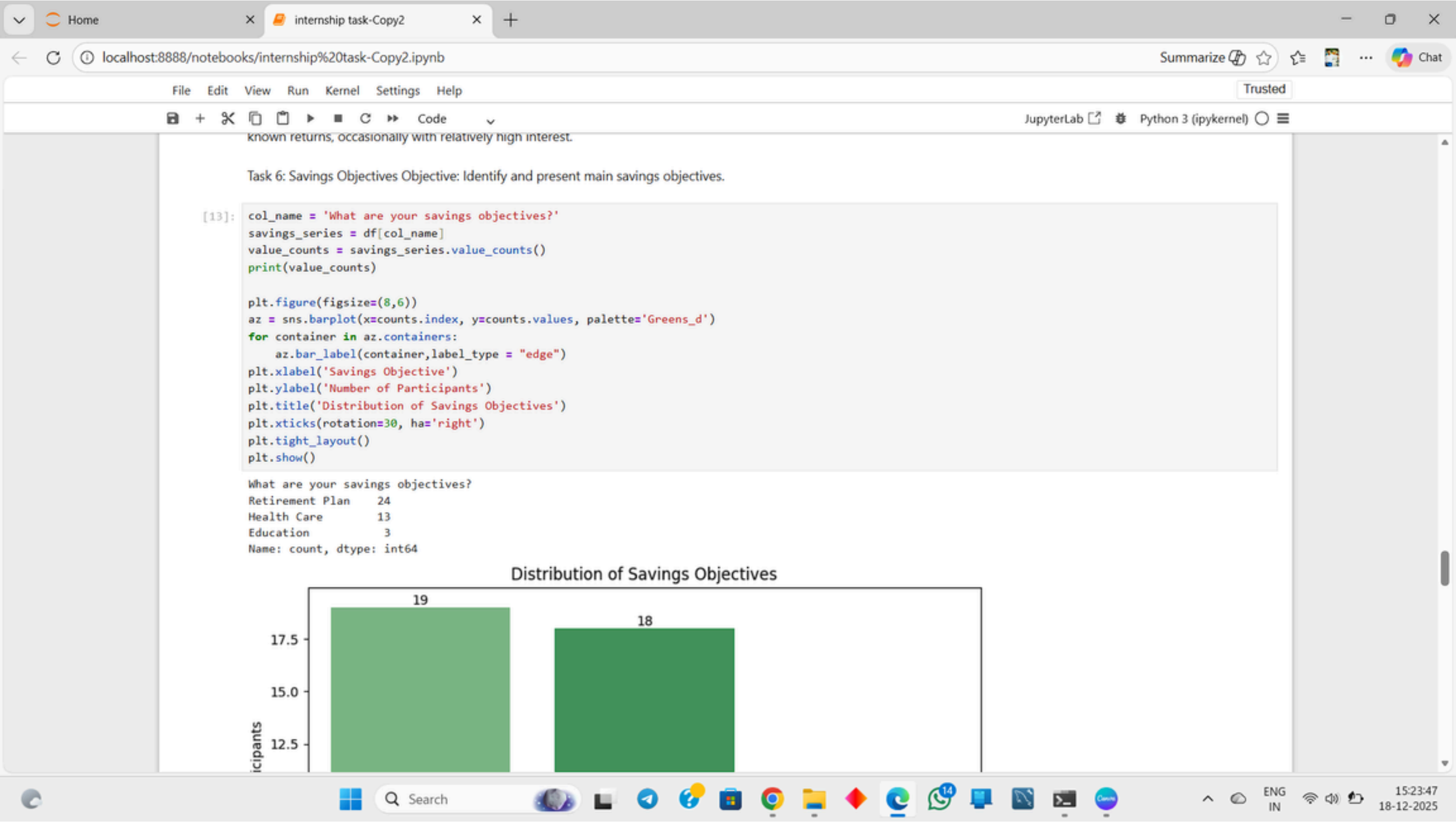
```
[12]: # Visualize the distribution of reasons across each investment avenue
import warnings
warnings.filterwarnings("ignore")
reason_cols = ['Reason_Equity', 'Reason_Mutual', 'Reason_Bonds', 'Reason_FD']

plt.figure(figsize=(12, 8))

for i, col in enumerate(reason_cols, 1):
    plt.subplot(2, 2, i)
    counts = df[col].value_counts()
    sns.barplot(x=counts.values, y=counts.index, palette='viridis')
    plt.title(col)
    plt.xlabel('Count')
    plt.ylabel('Reason')

plt.tight_layout()
plt.show()
```



Summarizing common themes / recurring reasons Across all four avenues, the main themes behind participants' investment choices are: Equity Capital Appreciation (most common) Dividend Liquidity Theme: People choose equity mainly for higher growth potential and, to a lesser extent, regular dividend income and ease of exit. Mutual Funds Better Returns Fund Diversification Tax Benefits Theme: Mutual funds are preferred for the combination of competitive returns, diversification of risk, and some tax advantages. Bonds Assured Returns Safe Investment Tax Incentives Theme: Bonds are seen as safer, more predictable investments with guaranteed or stable returns, often with some tax perks. Fixed Deposits (FDs) Risk Free Fixed Returns High Interest Rates Theme: FDs are primarily chosen for safety and certainty: low risk, guaranteed and known returns, occasionally with relatively high interest.

Task 6: Savings Objectives Objective: Identify and present main savings objectives.

```
[13]: col_name = 'What are your savings objectives?'
savings_series = df[col_name]
```

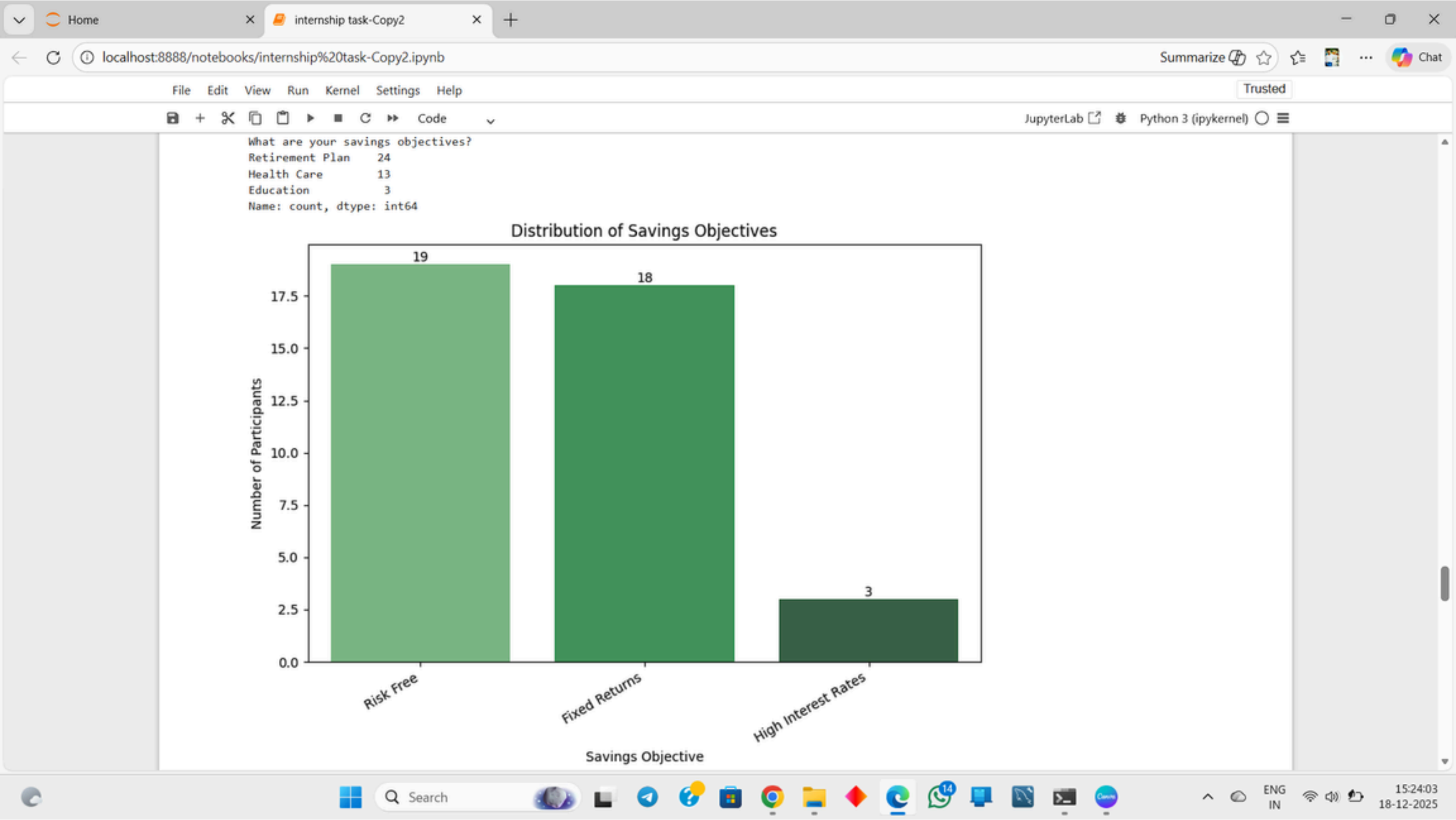


known returns, occasionally with relatively high interest.

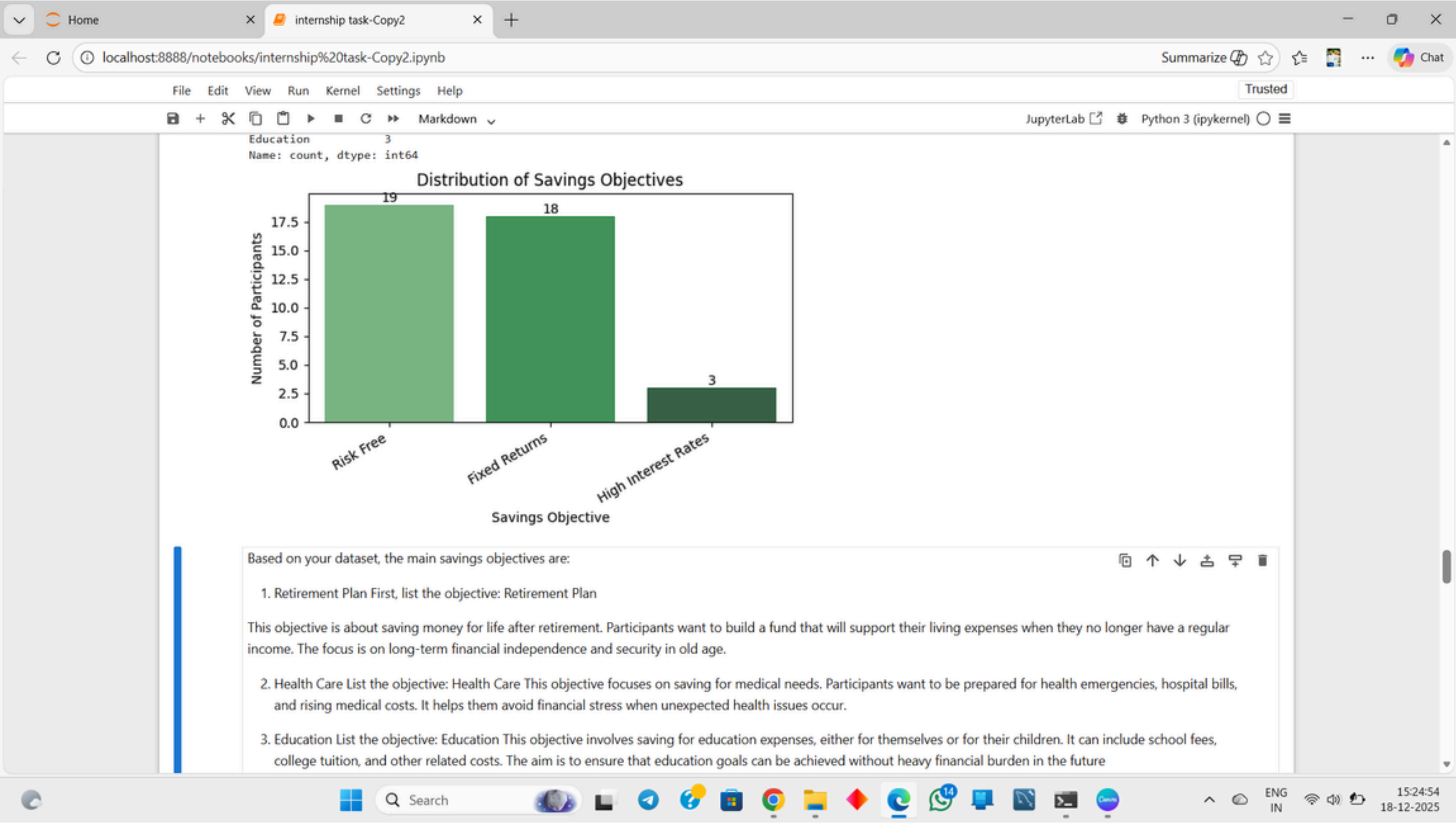
Task 6: Savings Objectives Objective: Identify and present main savings objectives.

```
[13]: col_name = 'What are your savings objectives?'
savings_series = df[col_name]
value_counts = savings_series.value_counts()
print(value_counts)

plt.figure(figsize=(8,6))
az = sns.barplot(x=counts.index, y=counts.values, palette='Greens_d')
for container in az.containers:
    az.bar_label(container,label_type = "edge")
plt.xlabel('Savings Objective')
plt.ylabel('Number of Participants')
plt.title('Distribution of Savings Objectives')
plt.xticks(rotation=30, ha='right')
plt.tight_layout()
plt.show()
```

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



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



```
Education          3
Name: count, dtype: int64
```

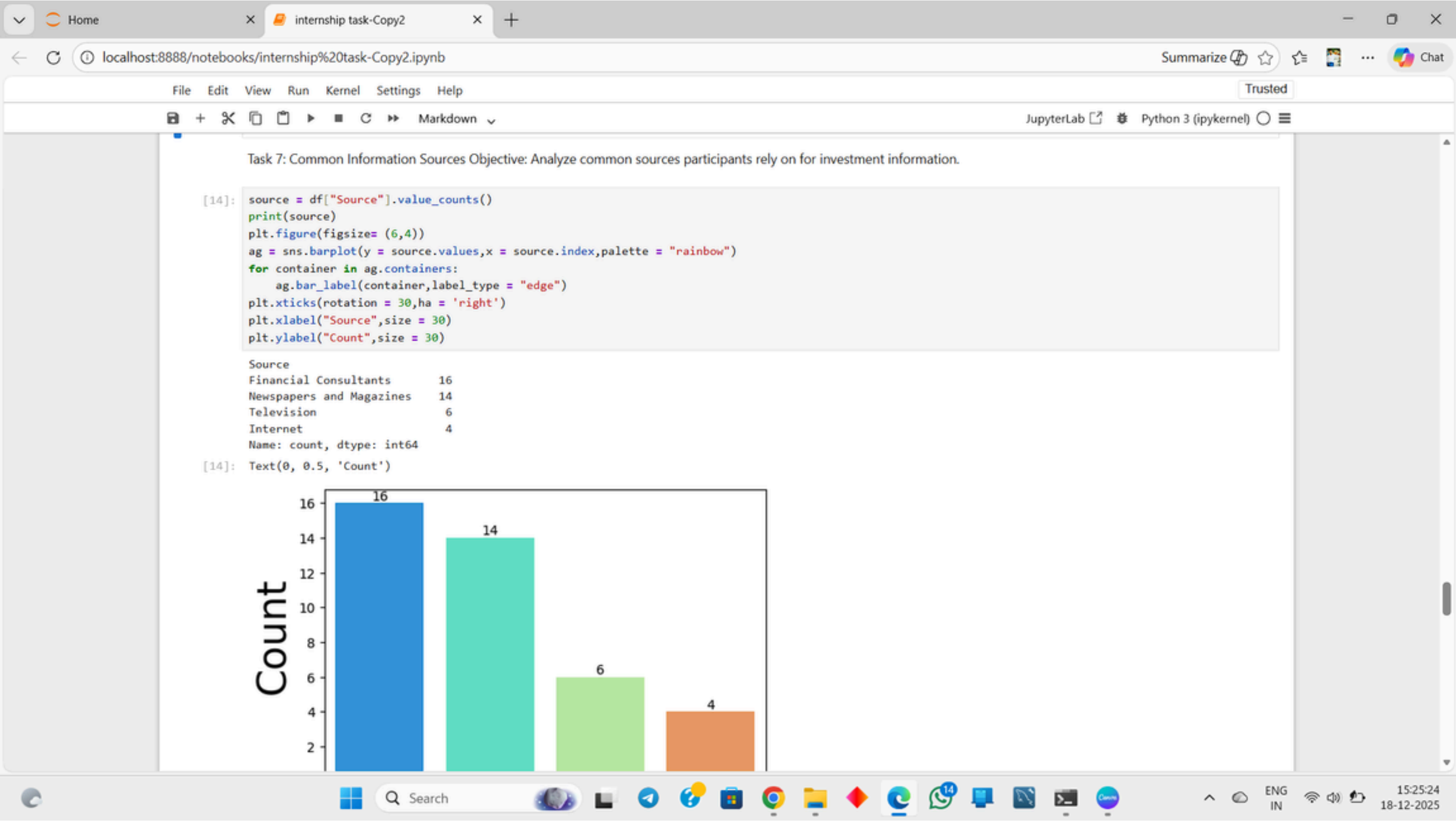
Based on your dataset, the main savings objectives are:

1. Retirement Plan First, list the objective: Retirement Plan

This objective is about saving money for life after retirement. Participants want to build a fund that will support their living expenses when they no longer have a regular income. The focus is on long-term financial independence and security in old age.

2. Health Care List the objective: Health Care This objective focuses on saving for medical needs. Participants want to be prepared for health emergencies, hospital bills, and rising medical costs. It helps them avoid financial stress when unexpected health issues occur.

3. Education List the objective: Education This objective involves saving for education expenses, either for themselves or for their children. It can include school fees, college tuition, and other related costs. The aim is to ensure that education goals can be achieved without heavy financial burden in the future

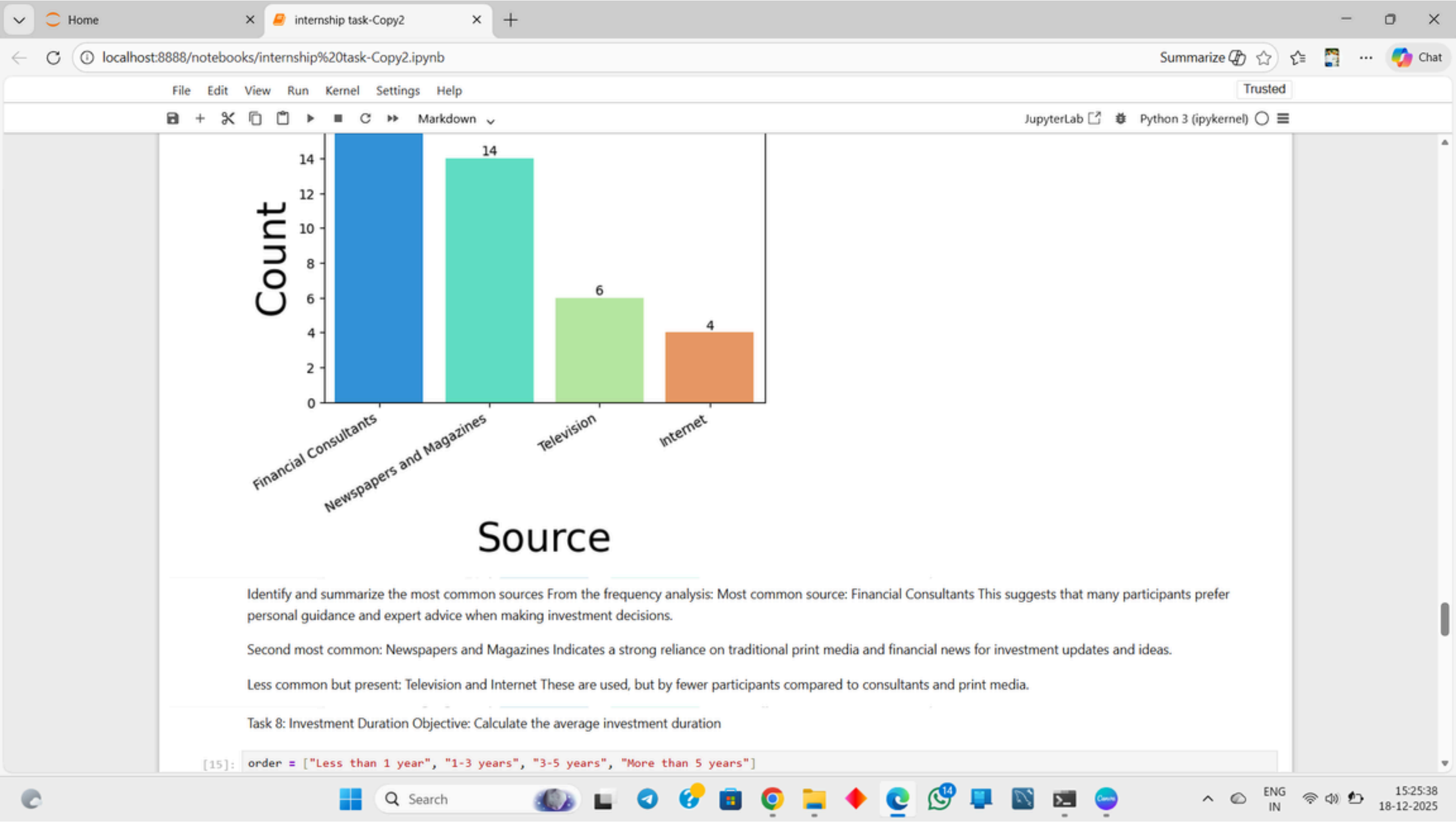


Task 7: Common Information Sources Objective: Analyze common sources participants rely on for investment information.

```
[14]: source = df["Source"].value_counts()
      print(source)
      plt.figure(figsize= (6,4))
      ag = sns.barplot(y = source.values,x = source.index,palette = "rainbow")
      for container in ag.containers:
          ag.bar_label(container,label_type = "edge")
      plt.xticks(rotation = 30,ha = 'right')
      plt.xlabel("Source",size = 30)
      plt.ylabel("Count",size = 30)
```

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

```
[14]: Text(0, 0.5, 'Count')
```



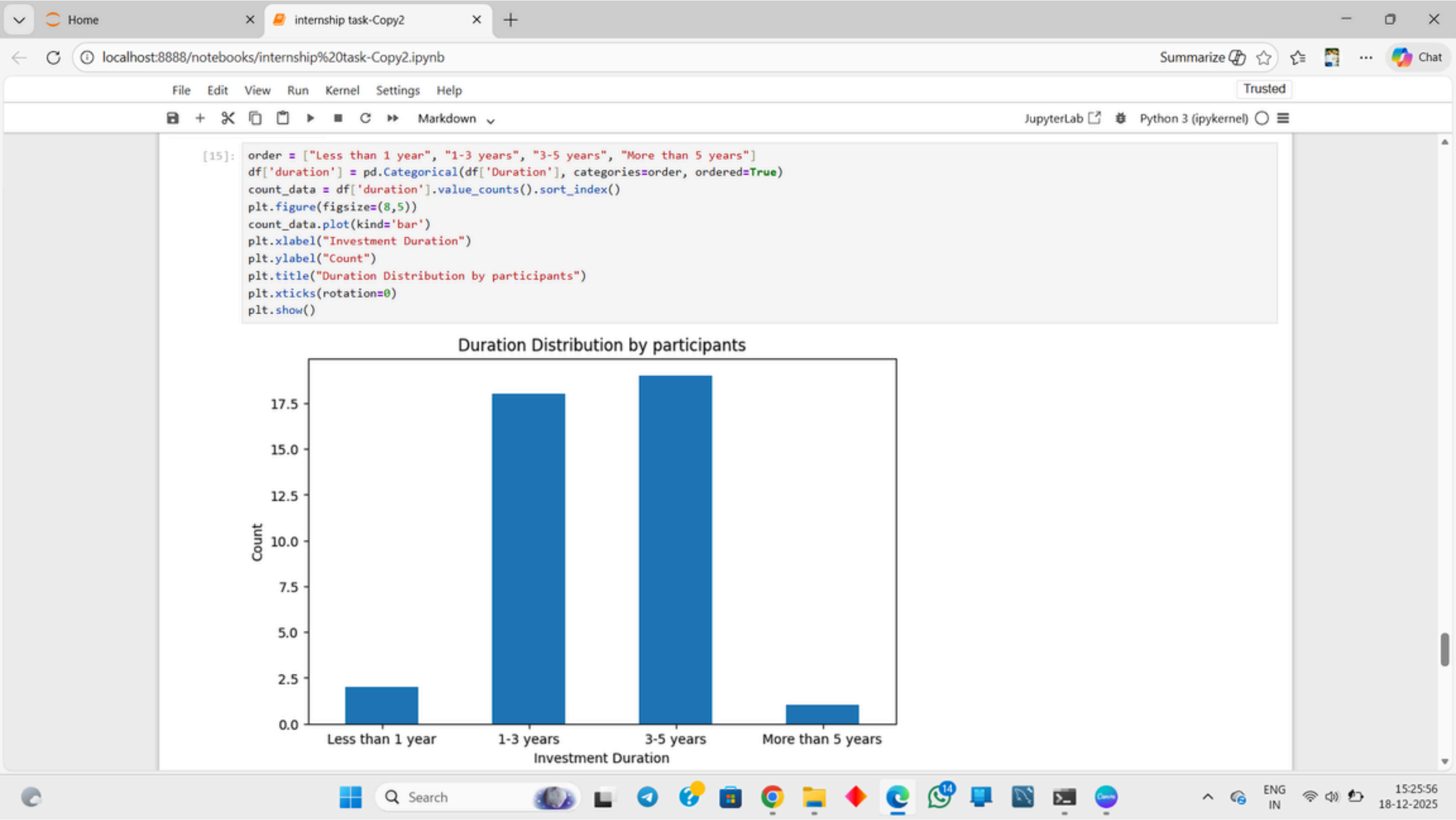
Identify and summarize the most common sources From the frequency analysis: Most common source: Financial Consultants This suggests that many participants prefer personal guidance and expert advice when making investment decisions.

Second most common: Newspapers and Magazines Indicates a strong reliance on traditional print media and financial news for investment updates and ideas.

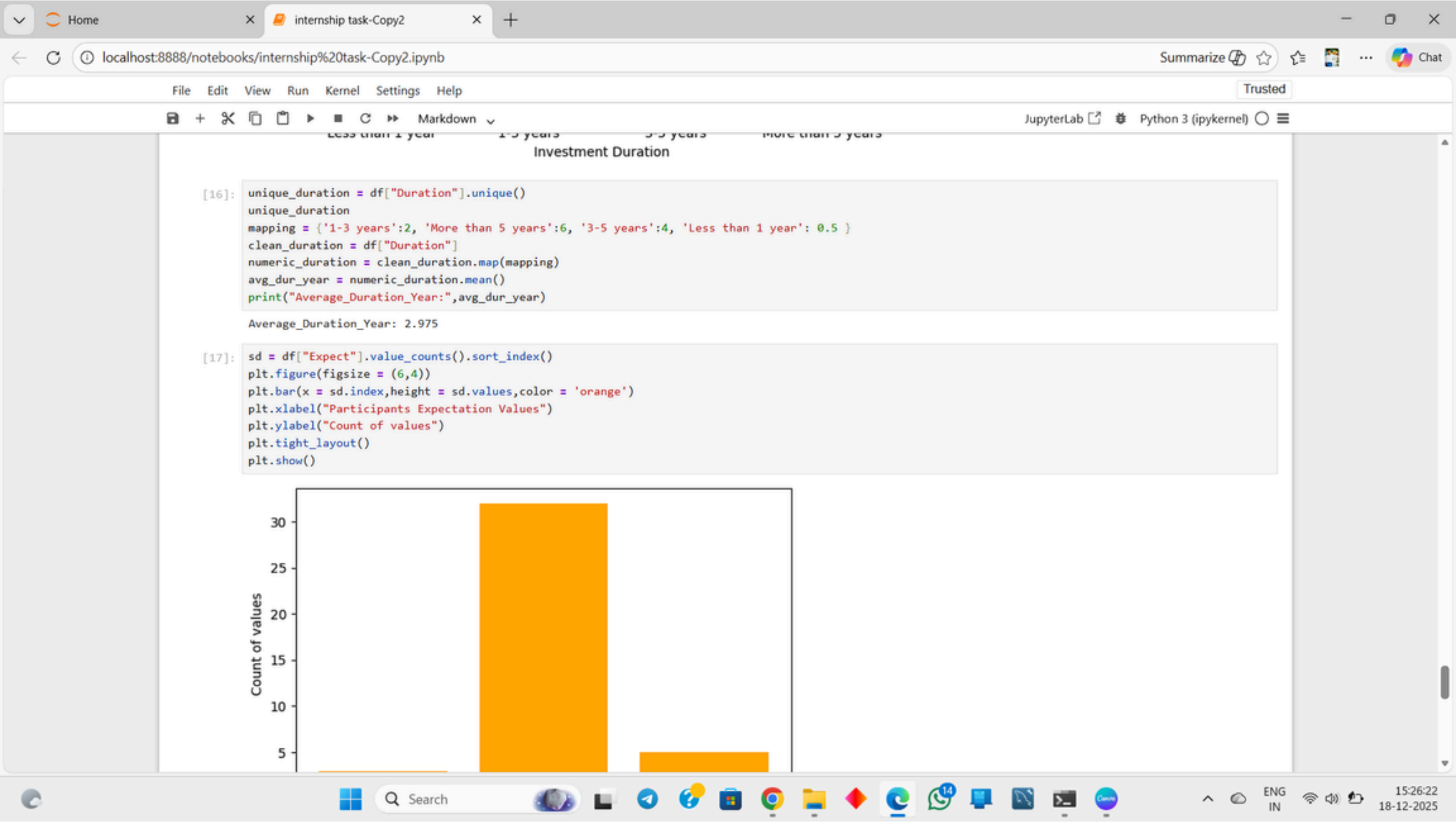
Less common but present: Television and Internet These are used, but by fewer participants compared to consultants and print media.

Task 8: Investment Duration Objective: Calculate the average investment duration

```
[15]: order = ["Less than 1 year", "1-3 years", "3-5 years", "More than 5 years"]
```

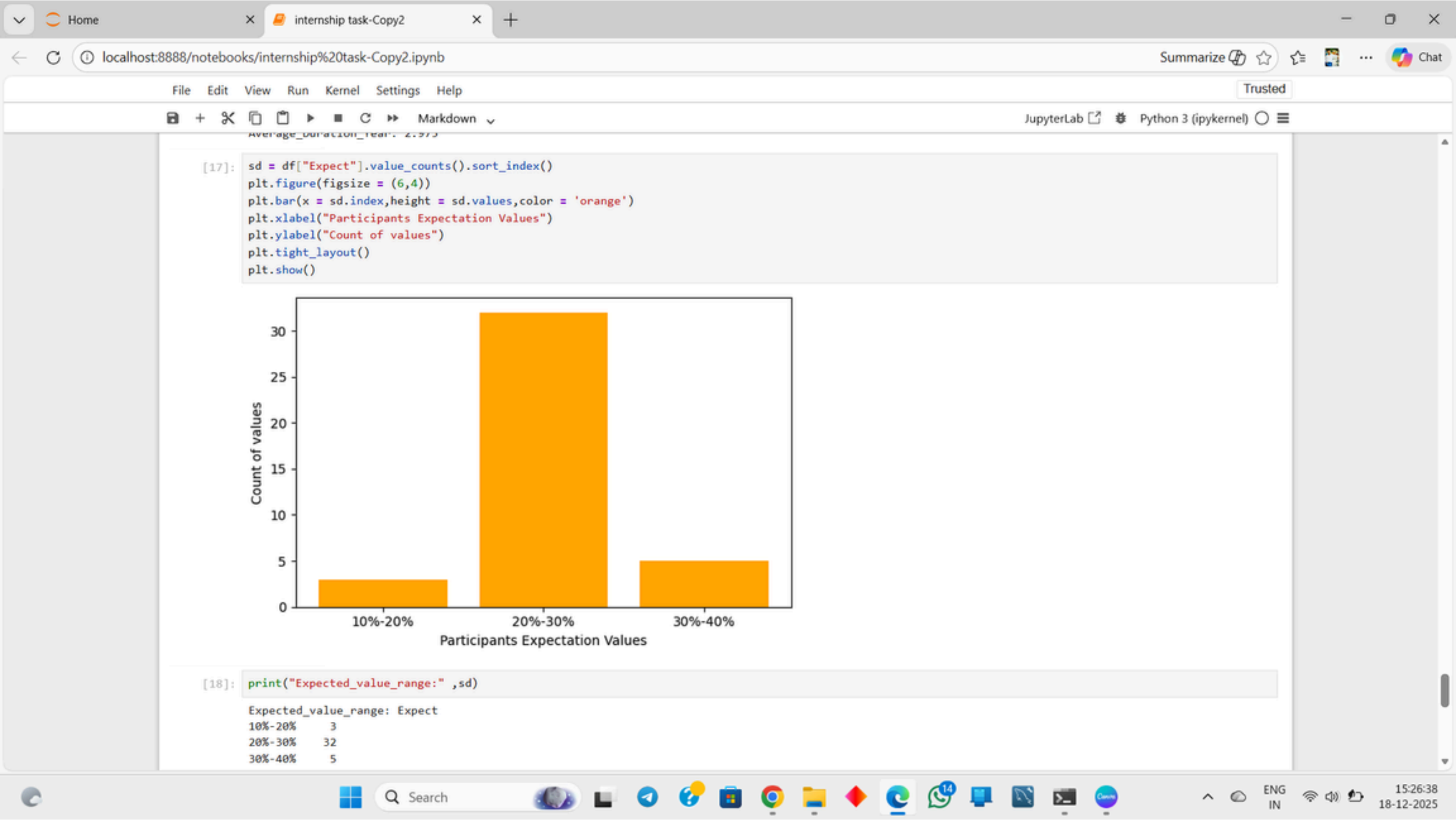

Home
internship task-Copy2
localhost:8888/notebooks/internship%20task-Copy2.ipynb
Summarize
Chat
File Edit View Run Kernel Settings Help
Trusted
Markdown
JupyterLab
Python 3 (ipykernel)

```
[15]: order = ["Less than 1 year", "1-3 years", "3-5 years", "More than 5 years"]
      df['duration'] = pd.Categorical(df['Duration'], categories=order, ordered=True)
      count_data = df['duration'].value_counts().sort_index()
      plt.figure(figsize=(8,5))
      count_data.plot(kind='bar')
      plt.xlabel("Investment Duration")
      plt.ylabel("Count")
      plt.title("Duration Distribution by participants")
      plt.xticks(rotation=0)
      plt.show()
```



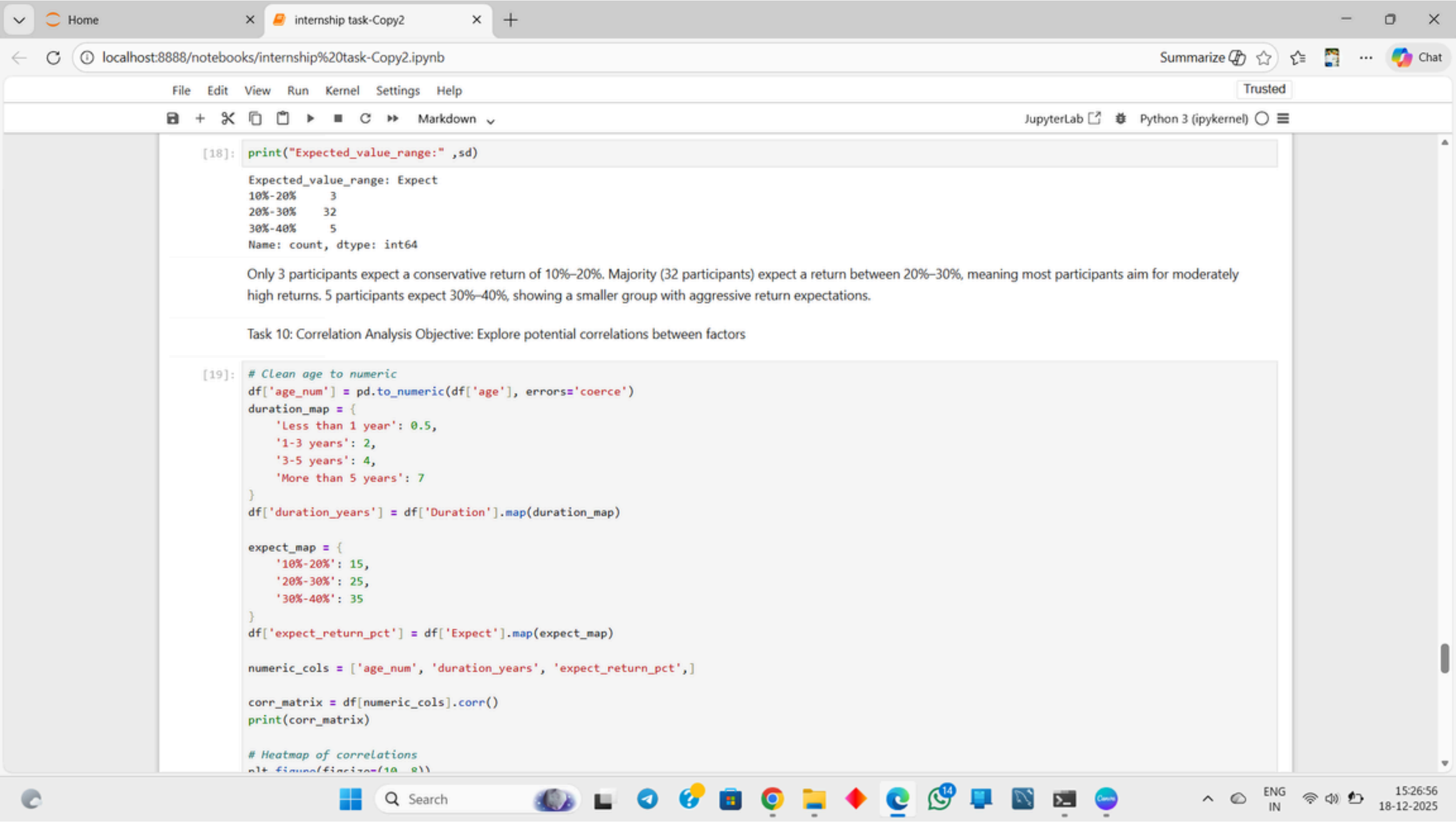
```
[16]: unique_duration = df["Duration"].unique()
      unique_duration
      mapping = {'1-3 years':2, 'More than 5 years':6, '3-5 years':4, 'Less than 1 year': 0.5 }
      clean_duration = df["Duration"]
      numeric_duration = clean_duration.map(mapping)
      avg_dur_year = numeric_duration.mean()
      print("Average_Duration_Year:",avg_dur_year)

      Average_Duration_Year: 2.975
```

```
[17]: sd = df["Expect"].value_counts().sort_index()
      plt.figure(figsize = (6,4))
      plt.bar(x = sd.index,height = sd.values,color = 'orange')
      plt.xlabel("Participants Expectation Values")
      plt.ylabel("Count of values")
      plt.tight_layout()
      plt.show()
```


Home
internship task-Copy2
localhost:8888/notebooks/internship%20task-Copy2.ipynb
File Edit View Run Kernel Settings Help
Trusted
Markdown
JupyterLab
Python 3 (ipykernel)
Average_Duration_Year: 2.975
30
25
20
15
10
5
0
Count of values
10%-20%
20%-30%
30%-40%
Participants Expectation Values
Search
ENG
IN
15:26:38
18-12-2025

```
[17]: sd = df["Expect"].value_counts().sort_index()
      plt.figure(figsize = (6,4))
      plt.bar(x = sd.index,height = sd.values,color = 'orange')
      plt.xlabel("Participants Expectation Values")
      plt.ylabel("Count of values")
      plt.tight_layout()
      plt.show()
```

```
[18]: print("Expected_value_range:" ,sd)

      Expected_value_range: Expect
      10%-20%     3
      20%-30%    32
      30%-40%     5
```

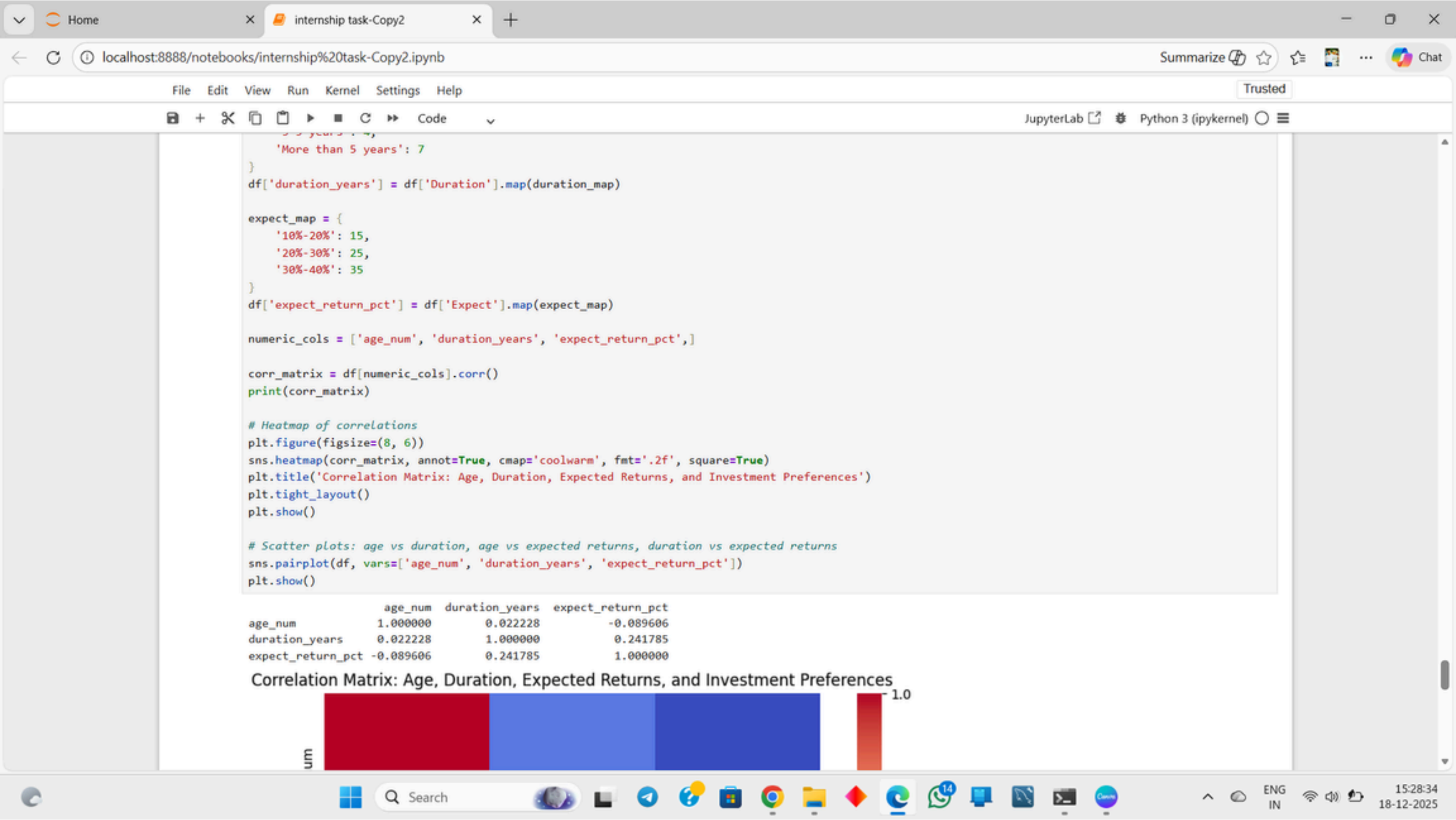


```
[18]: print("Expected_value_range:" ,sd)

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

Only 3 participants expect a conservative return of 10%–20%. Majority (32 participants) expect a return between 20%–30%, meaning most participants aim for moderately high returns. 5 participants expect 30%–40%, showing a smaller group with aggressive return expectations.

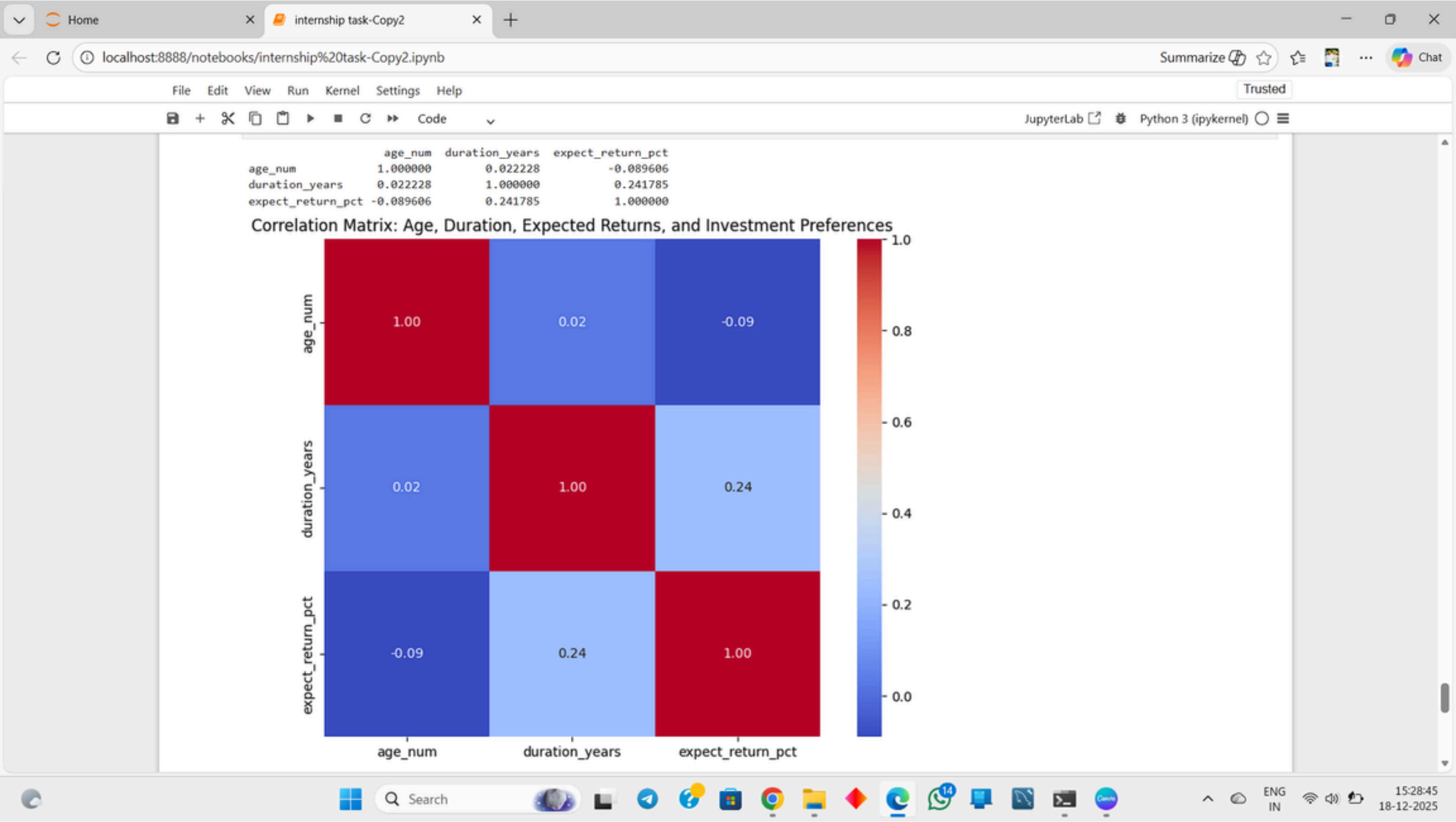
Task 10: Correlation Analysis Objective: Explore potential correlations between factors

```
[19]: # Clean age to numeric
df['age_num'] = pd.to_numeric(df['age'], errors='coerce')
duration_map = {
    'Less than 1 year': 0.5,
    '1-3 years': 2,
    '3-5 years': 4,
    'More than 5 years': 7
}
df['duration_years'] = df['Duration'].map(duration_map)

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

numeric_cols = ['age_num', 'duration_years', 'expect_return_pct',]

corr_matrix = df[numeric_cols].corr()
print(corr_matrix)

# Heatmap of correlations
plt.figure(figsize=(10, 8))
```

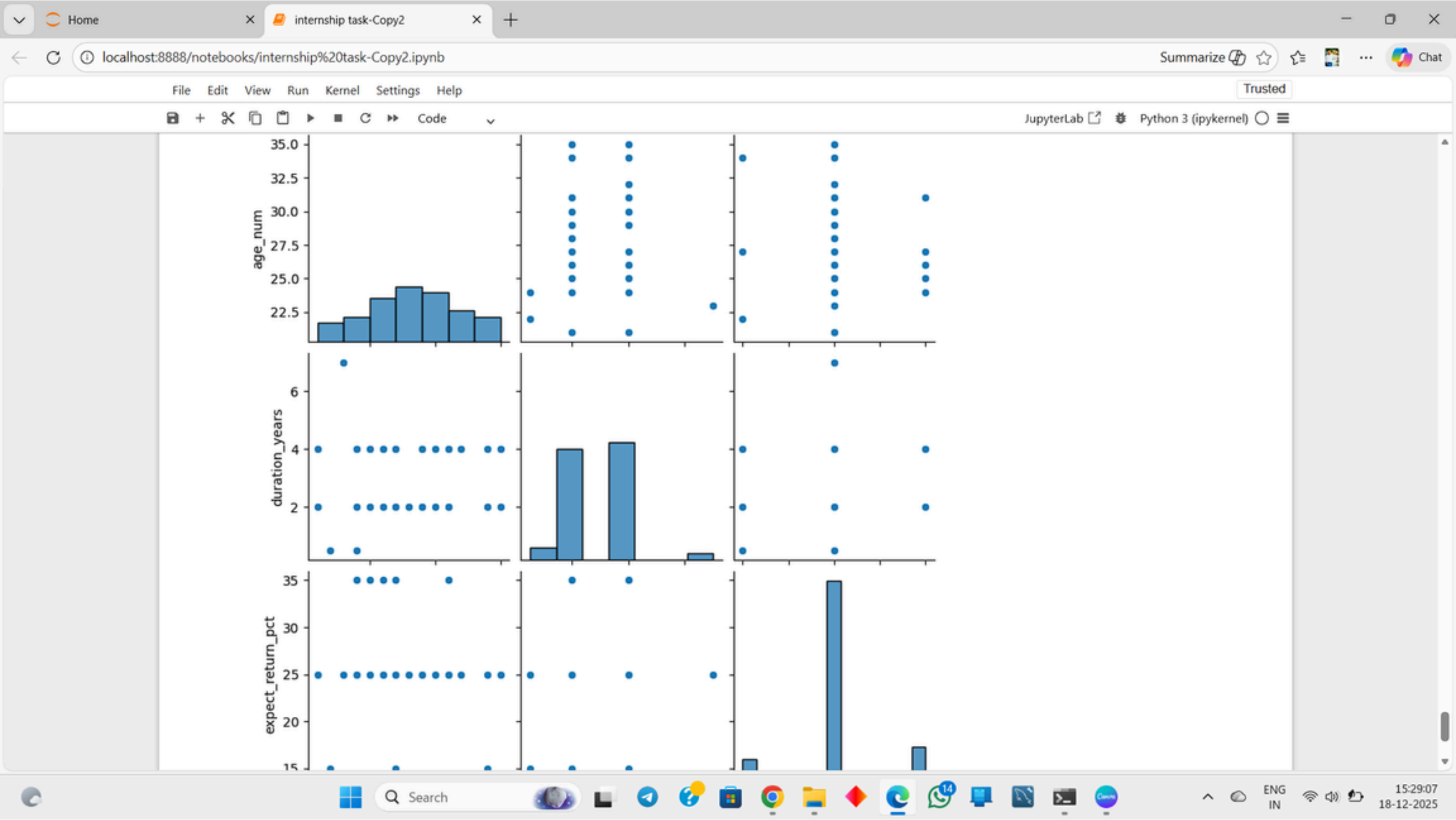


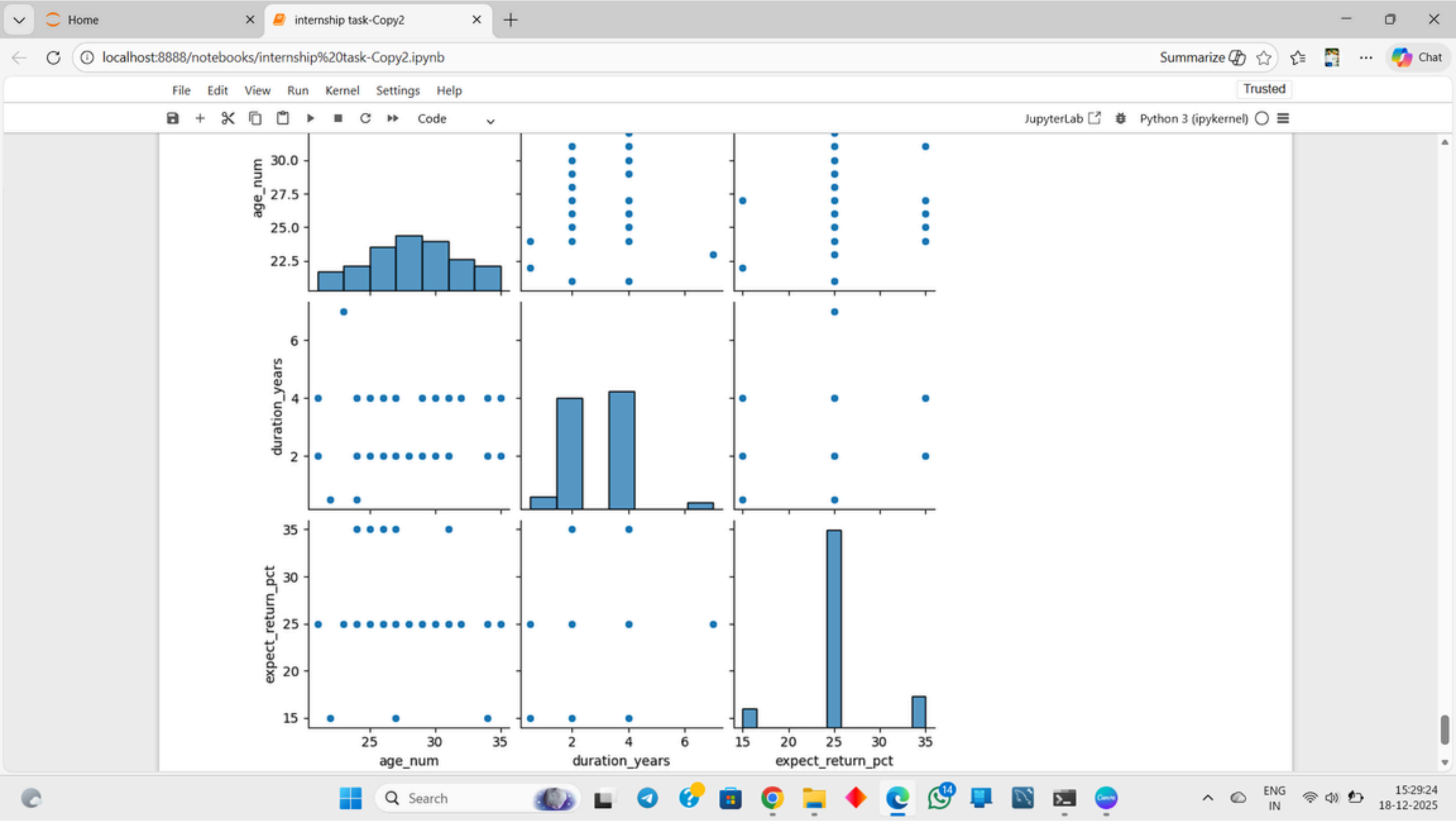
```
    'More than 5 years': 7
}
df['duration_years'] = df['Duration'].map(duration_map)

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

numeric_cols = ['age_num', 'duration_years', 'expect_return_pct',]

corr_matrix = df[numeric_cols].corr()
print(corr_matrix)

# Heatmap of correlations
plt.figure(figsize=(8, 6))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', fmt='.2f', square=True)
plt.title('Correlation Matrix: Age, Duration, Expected Returns, and Investment Preferences')
plt.tight_layout()
plt.show()

# Scatter plots: age vs duration, age vs expected returns, duration vs expected returns
sns.pairplot(df, vars=['age_num', 'duration_years', 'expect_return_pct'])
plt.show()

                    age_num  duration_years  expect_return_pct
age_num            1.000000        0.022228          -0.089606
duration_years     0.022228        1.000000           0.241785
expect_return_pct -0.089606        0.241785           1.000000
```

Correlation Matrix: Age, Duration, Expected Returns, and Investment Preferences



```
                 age_num  duration_years  expect_return_pct
age_num          1.000000        0.022228          -0.089606
duration_years   0.022228        1.000000           0.241785
expect_return_pct -0.089606      0.241785           1.000000
```




Home
internship task-Copy2
localhost:8888/notebooks/internship%20task-Copy2.ipynb
Summarize
Chat
File
Edit
View
Run
Kernel
Settings
Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)
age_num
30.0
27.5
25.0
22.5
duration_years
6
4
2
expect_return_pct
35
30
25
20
15
25
30
35
age_num
2
4
6
duration_years
15
20
25
30
35
expect_return_pct
Search
ENG
IN
15:29:24
18-12-2025