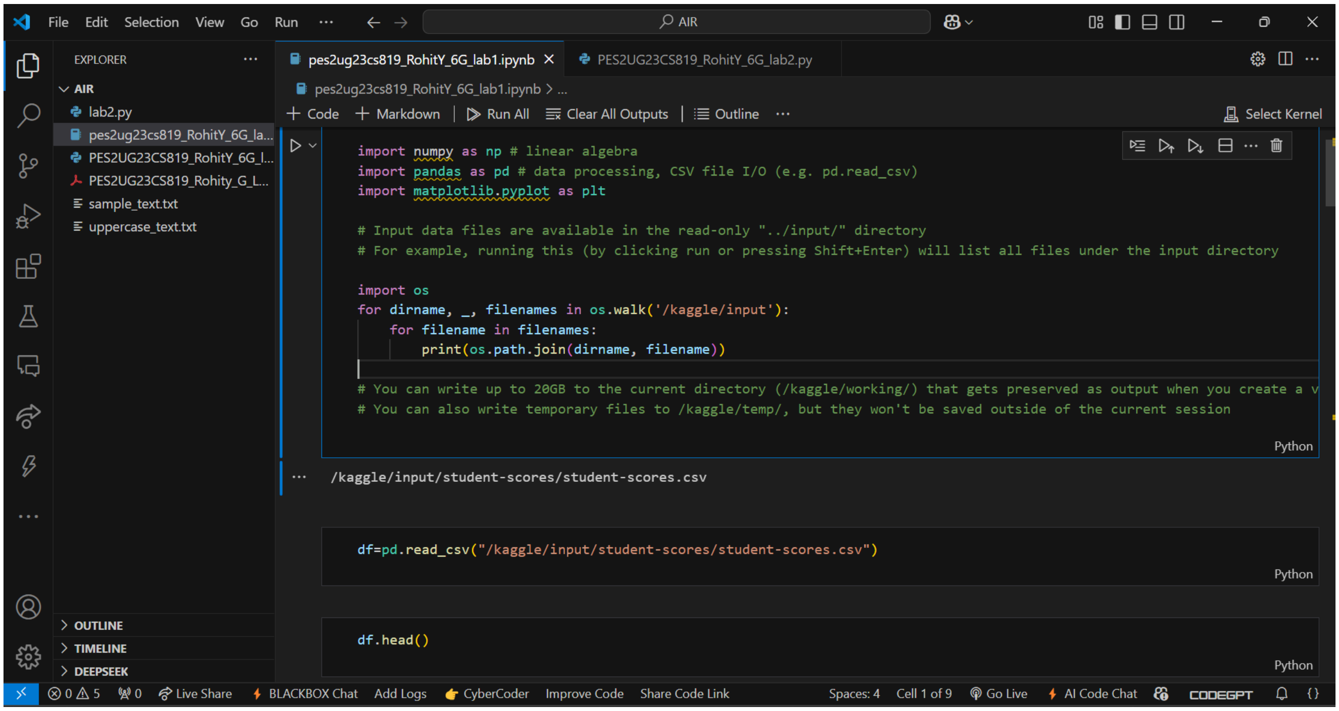
LAB 1

1. Retrieve minimum & maximum marks for each subject [Bar-graph reqd.]
2. Extract Gender Diversity [Pie Chart reqd.]
3. Retrieve the number of students who have taken part in extracurricular activities [Pie Chart reqd.]
4. Sort the names in alphabetical order and store it in a new csv file

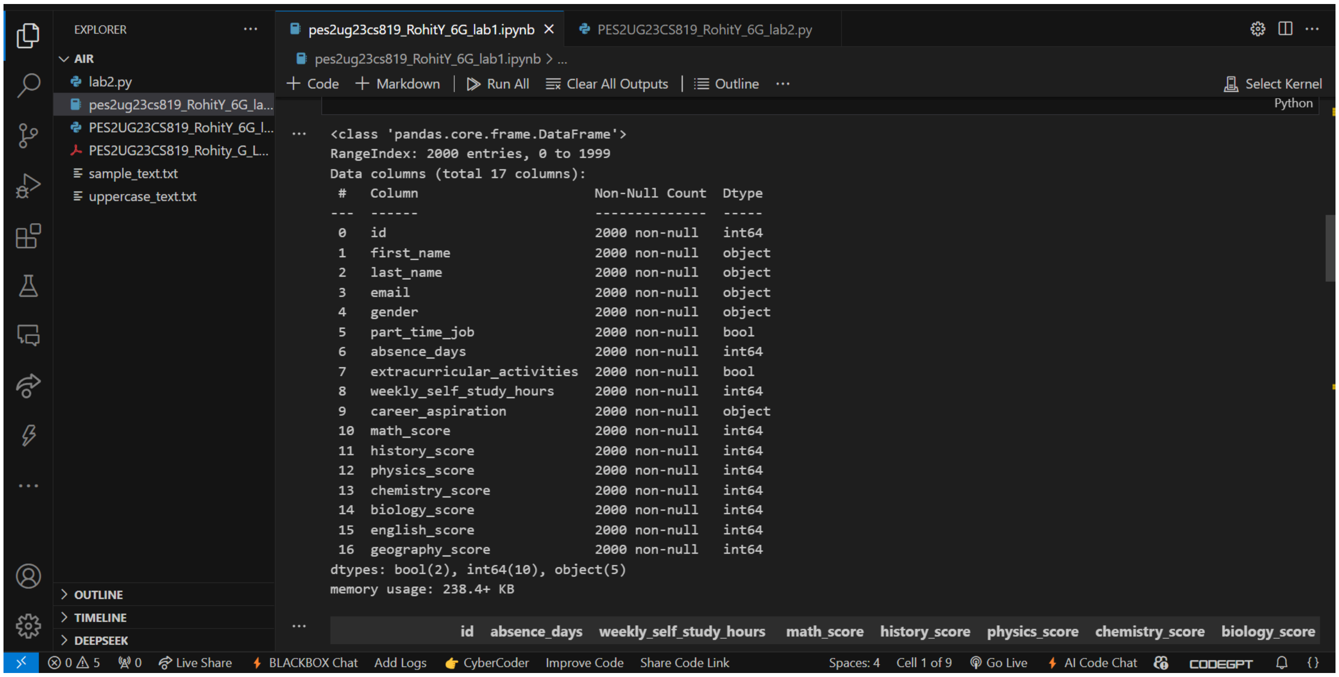


```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import matplotlib.pyplot as plt

# Input data files are available in the read-only "../input/" directory
# For example, running this (by clicking run or pressing Shift+Enter) will list all files under the input directory

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

# You can write up to 20GB to the current directory (/kaggle/working/) that gets preserved as output when you create a v
# You can also write temporary files to /kaggle/temp/, but they won't be saved outside of the current session
```

```
/kaggle/input/student-scores/student-scores.csv
```

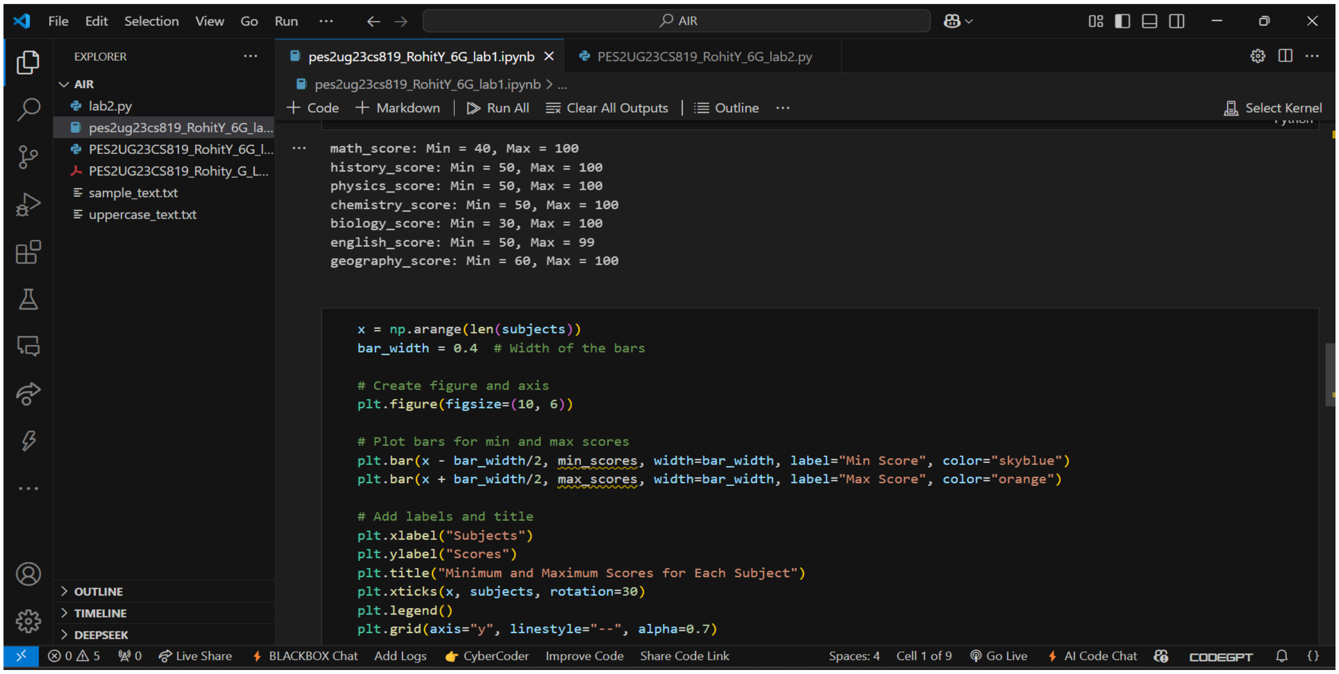
```
df=pd.read_csv("/kaggle/input/student-scores/student-scores.csv")
```

```
df.head()
```



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2000 entries, 0 to 1999
Data columns (total 17 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   id                          2000 non-null   int64
 1   first_name                  2000 non-null   object
 2   last_name                   2000 non-null   object
 3   email                       2000 non-null   object
 4   gender                      2000 non-null   object
 5   part_time_job               2000 non-null   bool
 6   absence_days                2000 non-null   int64
 7   extracurricular_activities  2000 non-null   bool
 8   weekly_self_study_hours     2000 non-null   int64
 9   career_aspiration           2000 non-null   object
 10  math_score                  2000 non-null   int64
 11  history_score               2000 non-null   int64
 12  physics_score               2000 non-null   int64
 13  chemistry_score             2000 non-null   int64
 14  biology_score               2000 non-null   int64
 15  english_score               2000 non-null   int64
 16  geography_score             2000 non-null   int64
dtypes: bool(2), int64(10), object(5)
memory usage: 238.4+ KB
```

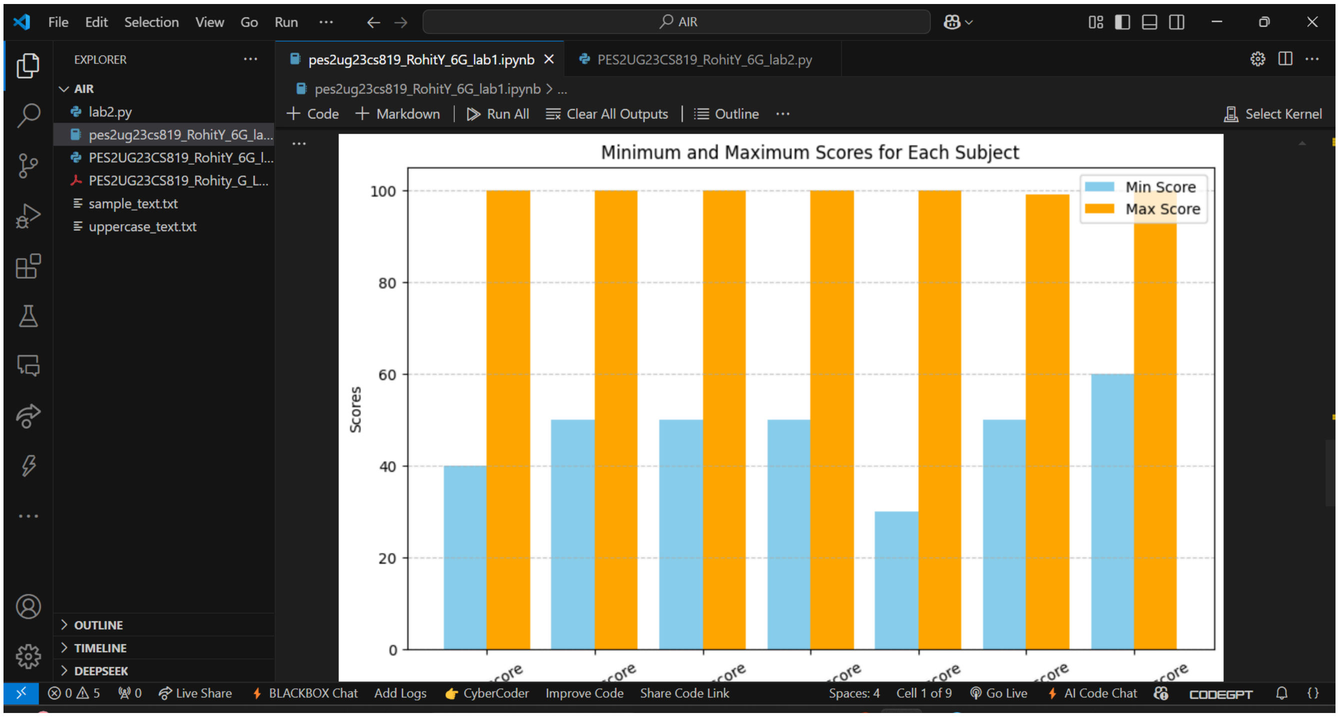
```
id  absence_days  weekly_self_study_hours  math_score  history_score  physics_score  chemistry_score  biology_score
```

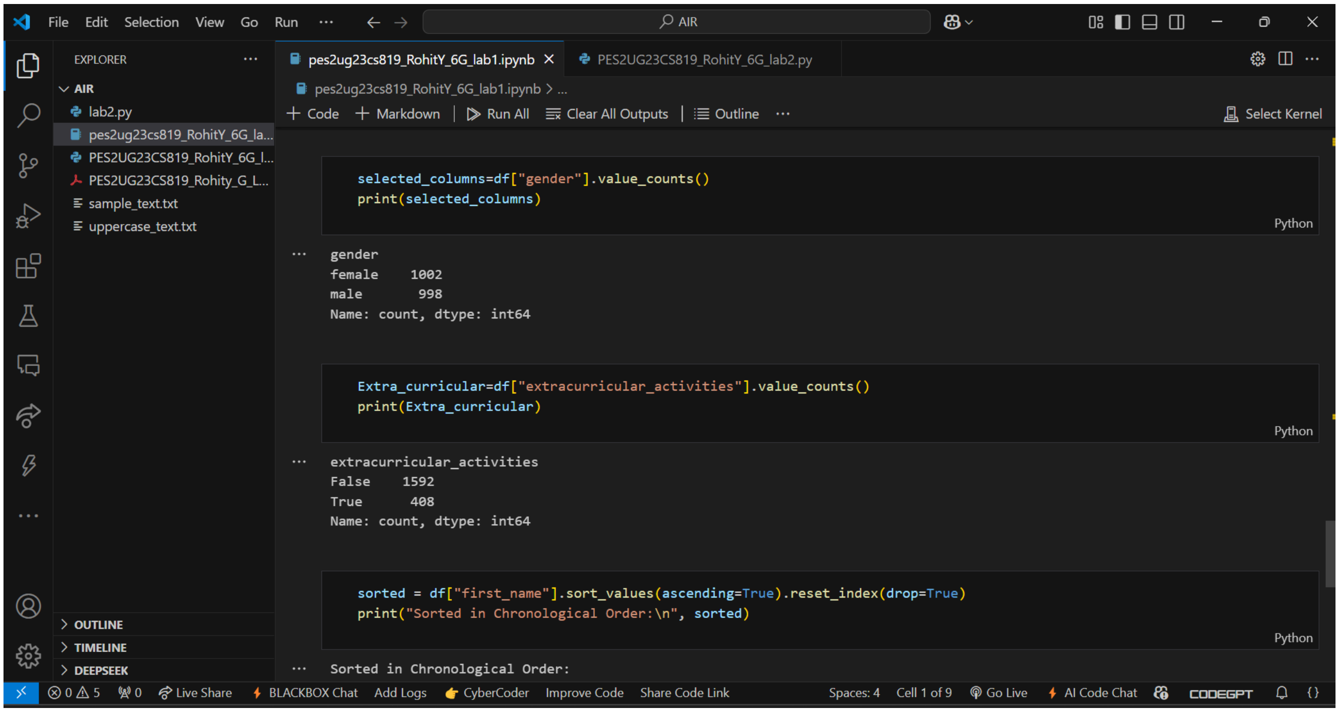


File Edit Selection View Go Run
AIR
EXPLORER
lab2.py
pes2ug23cs819_RohitY_6G_la...
PES2UG23CS819_RohitY_6G_l...
PES2UG23CS819_Rohity_G_L...
sample_text.txt
uppercase_text.txt
pes2ug23cs819_RohitY_6G_lab1.ipynb
PES2UG23CS819_RohitY_6G_lab2.py
+ Code + Markdown Run All Clear All Outputs Outline
Select Kernel
OUTLINE
TIMELINE
DEEPSEEK
Live Share BLACKBOX Chat Add Logs CyberCoder Improve Code Share Code Link Spaces: 4 Cell 1 of 9 Go Live AI Code Chat CODEGPT

```
math_score: Min = 40, Max = 100
history_score: Min = 50, Max = 100
physics_score: Min = 50, Max = 100
chemistry_score: Min = 50, Max = 100
biology_score: Min = 30, Max = 100
english_score: Min = 50, Max = 99
geography_score: Min = 60, Max = 100
```

```python
x = np.arange(len(subjects))
bar_width = 0.4  # Width of the bars

# Create figure and axis
plt.figure(figsize=(10, 6))

# Plot bars for min and max scores
plt.bar(x - bar_width/2, min_scores, width=bar_width, label="Min Score", color="skyblue")
plt.bar(x + bar_width/2, max_scores, width=bar_width, label="Max Score", color="orange")

# Add labels and title
plt.xlabel("Subjects")
plt.ylabel("Scores")
plt.title("Minimum and Maximum Scores for Each Subject")
plt.xticks(x, subjects, rotation=30)
plt.legend()
plt.grid(axis="y", linestyle="--", alpha=0.7)
```




File Edit Selection View Go Run ···
AIR
EXPLORER
AIR
lab2.py
pes2ug23cs819_RohitY_6G_la...
PES2UG23CS819_RohitY_6G_l...
PES2UG23CS819_Rohity_G_L...
sample_text.txt
uppercase_text.txt
OUTLINE
TIMELINE
DEEPSEEK
pes2ug23cs819_RohitY_6G_lab1.ipynb
PES2UG23CS819_RohitY_6G_lab2.py
pes2ug23cs819_RohitY_6G_lab1.ipynb > ...
+ Code + Markdown | Run All Clear All Outputs | Outline ···
Select Kernel
Python
0 5 0 Live Share BLACKBOX Chat Add Logs CyberCoder Improve Code Share Code Link Spaces: 4 Cell 1 of 9 Go Live AI Code Chat CODEGPT

```
selected_columns=df["gender"].value_counts()
print(selected_columns)
```

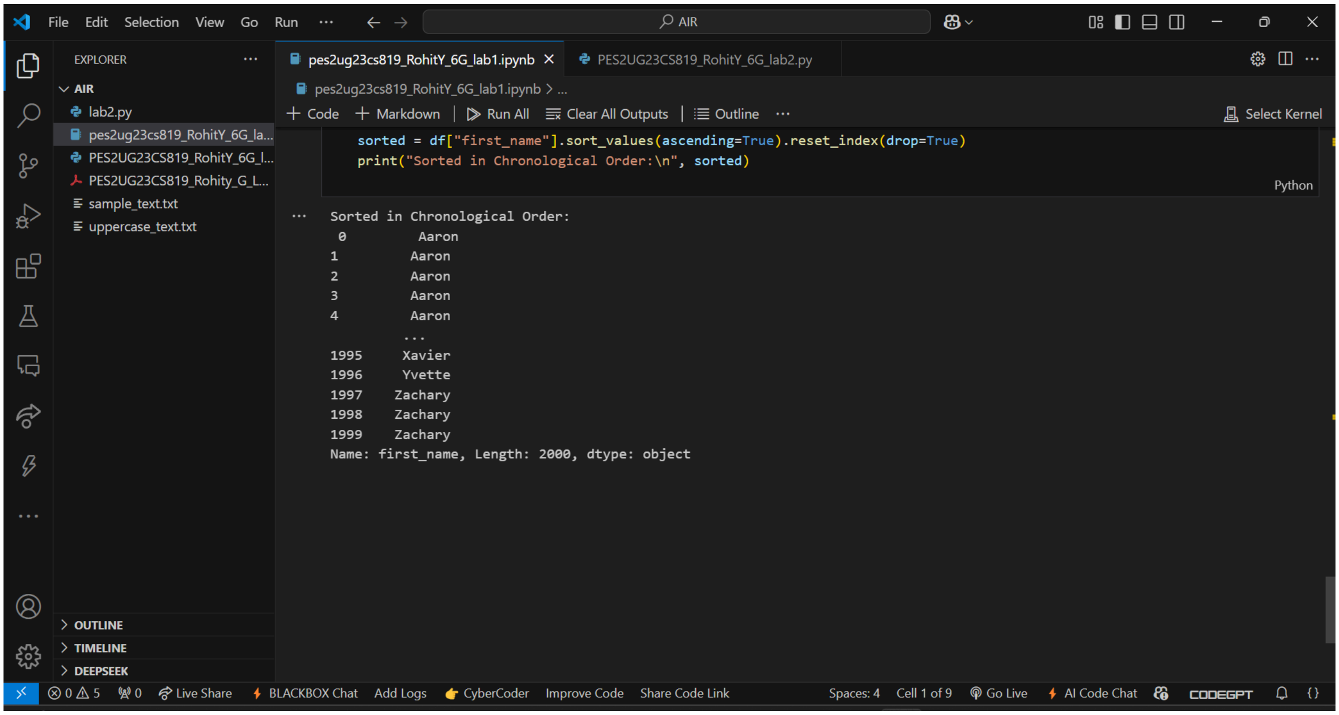
```
gender
female    1002
male       998
Name: count, dtype: int64
```

```
Extra_curricular=df["extracurricular_activities"].value_counts()
print(Extra_curricular)
```

```
extracurricular_activities
False    1592
True      408
Name: count, dtype: int64
```

```
sorted = df["first_name"].sort_values(ascending=True).reset_index(drop=True)
print("Sorted in Chronological Order:\n", sorted)
```

```
Sorted in Chronological Order:
```


File Edit Selection View Go Run ···
AIR
EXPLORER
AIR
lab2.py
pes2ug23cs819_RohitY_6G_la...
PES2UG23CS819_RohitY_6G_l...
PES2UG23CS819_Rohity_G_L...
sample_text.txt
uppercase_text.txt
OUTLINE
TIMELINE
DEEPSEEK
pes2ug23cs819_RohitY_6G_lab1.ipynb
PES2UG23CS819_RohitY_6G_lab2.py
pes2ug23cs819_RohitY_6G_lab1.ipynb > ...
+ Code + Markdown | Run All Clear All Outputs | Outline ···
Select Kernel
Python
0 5 0 Live Share BLACKBOX Chat Add Logs CyberCoder Improve Code Share Code Link Spaces: 4 Cell 1 of 9 Go Live AI Code Chat CODEGPT

```
sorted = df["first_name"].sort_values(ascending=True).reset_index(drop=True)
print("Sorted in Chronological Order:\n", sorted)
```

```
Sorted in Chronological Order:
 0         Aaron
1         Aaron
2         Aaron
3         Aaron
4         Aaron
         ...
1995     Xavier
1996     Yvette
1997    Zachary
1998    Zachary
1999    Zachary
Name: first_name, Length: 2000, dtype: object
```