Pandas

Pandas is a powerful and open-source Python library. The Pandas library is used for data manipulation and analysis. Pandas consist of data structures and functions to perform efficient operations on data.

Pandas is well-suited for working with tabular data, such as spreadsheets or SQL tables.

The name "Pandas" has a reference to both "Panel Data", and "Python Data Analysis" and was created by Wes McKinney in 2008.

Pandas allows us to analyze big data and make conclusions based on statistical theories.
Pandas can clean messy data sets, and make them readable and relevant.
Relevant data is very important in data science.

Steps

1. Open Anaconda Prompt
2. Type ‘jupyter notebook’
3. It will redirect to Home page of Jupyter notebook on browser
4. In Files tab open Desktop folder
5. Create new folder using New button on top right and name it
6. Open the folder and click on Upload button
7. Upload the dataset file (.csv , .xlsv) on which you have to perform tasks
8. Create a Python file by clicking on new button and choosing ‘Python 3’ file
9. On creating, file will open in new tab of the browser
10. Import pandas
11. Store dataset file in a variable

Perform the following operations on dataset

Before performing tasks on dataset we need to

1. Import pandas

   import pandas as pd

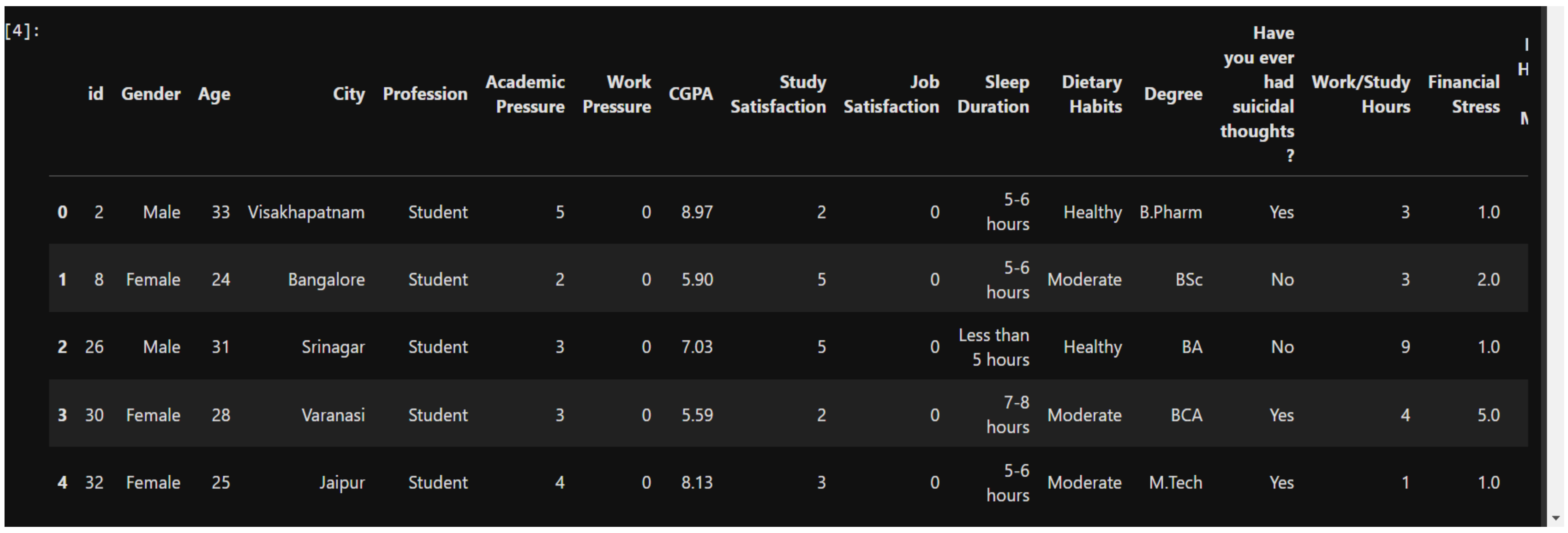
2. Store dataset file in a variable

   ds = pd.read_csv("Student Depression Dataset.csv")

1. **What are the first five rows of the dataset?**

   df.head()

   OUTPUT:



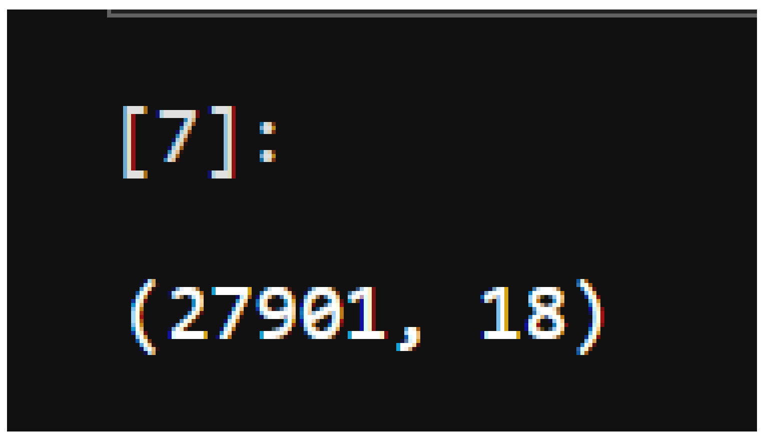
```
[4]:
```

	id	Gender	Age	City	Profession	Academic Pressure	Work Pressure	CGPA	Study Satisfaction	Job Satisfaction	Sleep Duration	Dietary Habits	Degree	Have you ever had suicidal thoughts ?	Work/Study Hours	Financial Stress
0	2	Male	33	Visakhapatnam	Student	5	0	8.97	2	0	5-6 hours	Healthy	B.Pharm	Yes	3	1.0
1	8	Female	24	Bangalore	Student	2	0	5.90	5	0	5-6 hours	Moderate	BSc	No	3	2.0
2	26	Male	31	Srinagar	Student	3	0	7.03	5	0	Less than 5 hours	Healthy	BA	No	9	1.0
3	30	Female	28	Varanasi	Student	3	0	5.59	2	0	7-8 hours	Moderate	BCA	Yes	4	5.0
4	32	Female	25	Jaipur	Student	4	0	8.13	3	0	5-6 hours	Moderate	M.Tech	Yes	1	1.0

2. **What is the total number of rows and columns in the dataset?**

   df.shape # Returns (number of rows, number of columns)

OUTPUT:

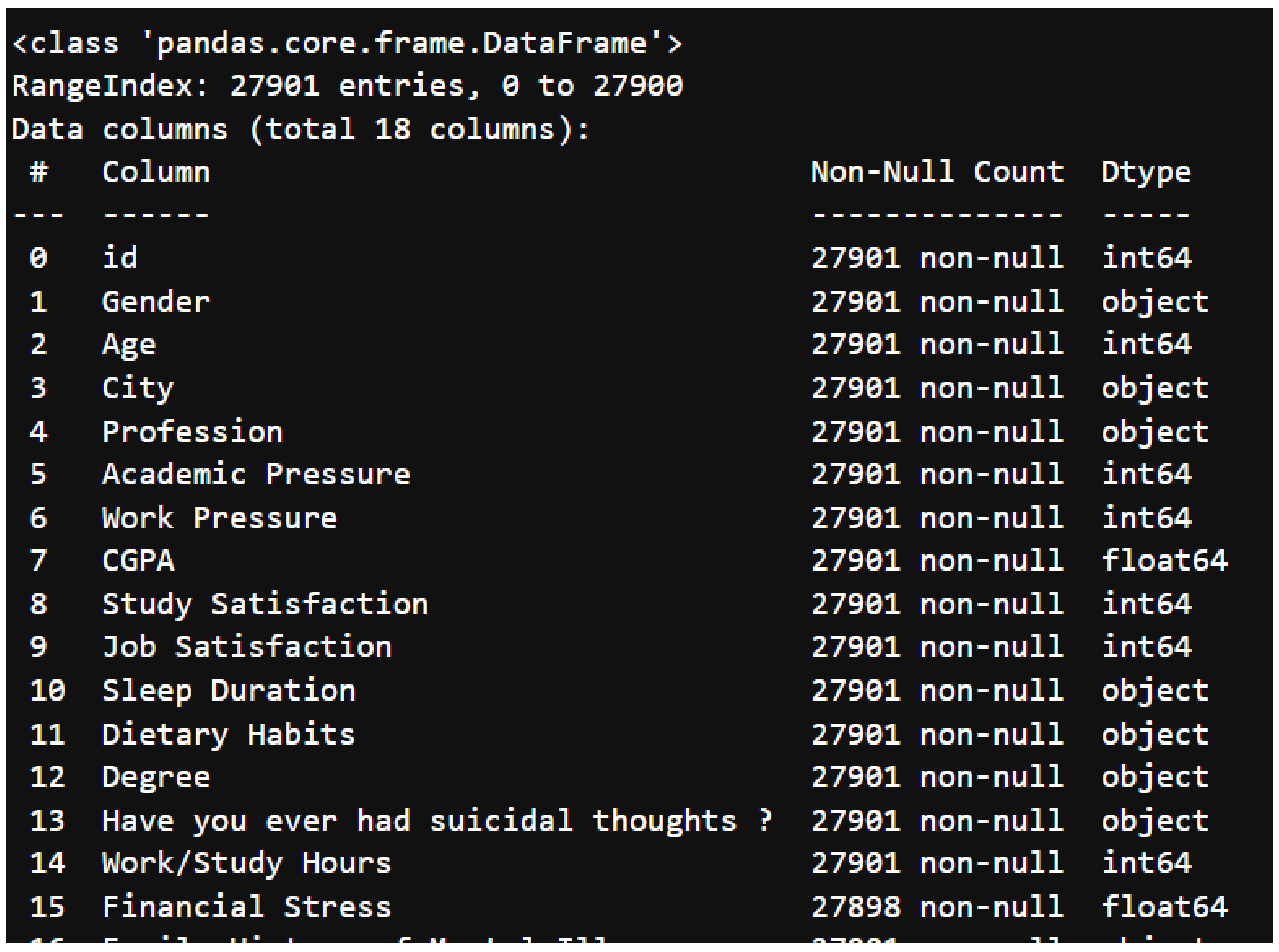


```
[7]:

(27901, 18)
```

3.How to Get Detailed Information About the Dataset?

   df.info() # Displays column names, data types, and non-null values

OUTPUT:

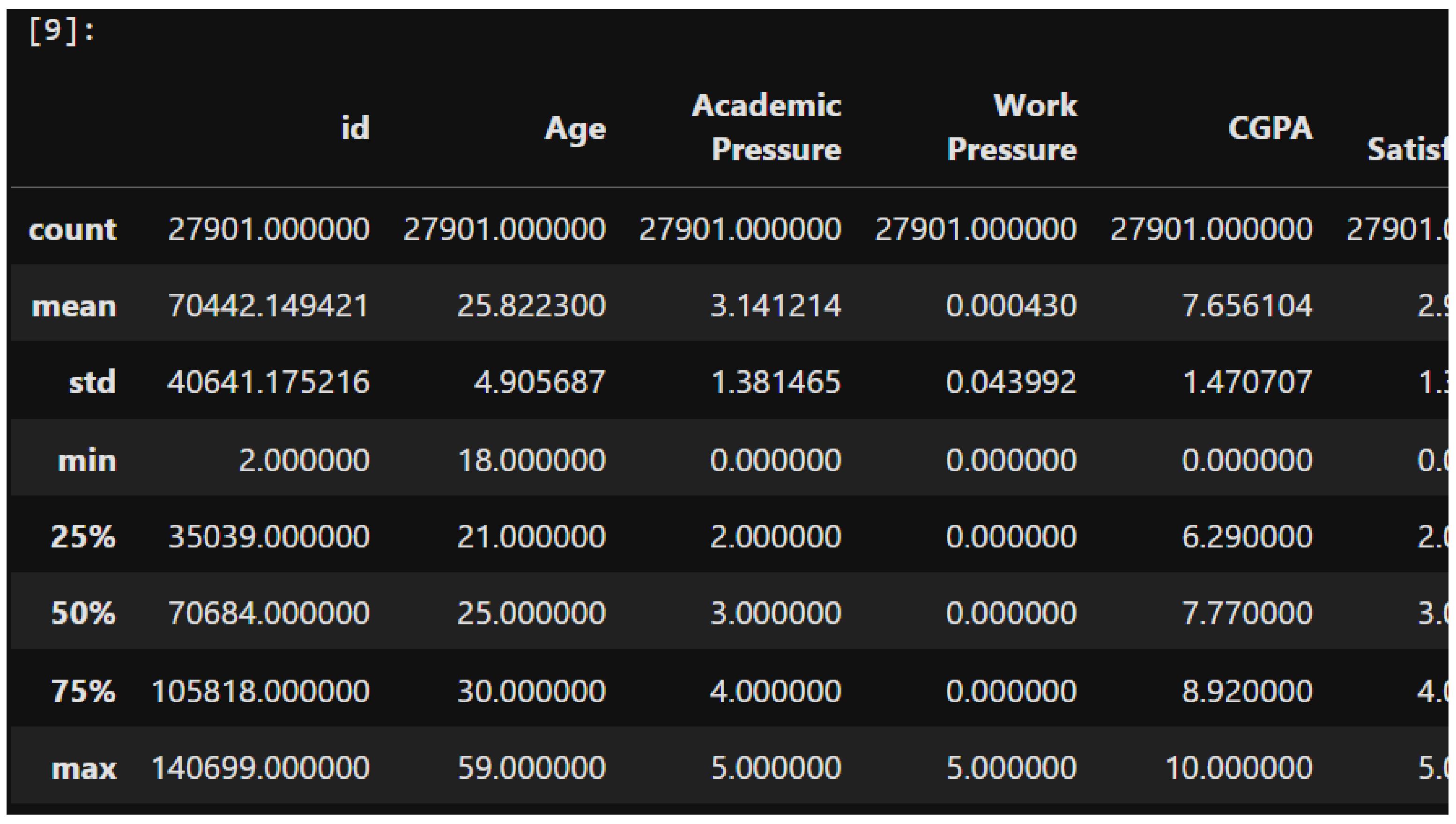


```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 27901 entries, 0 to 27900
Data columns (total 18 columns):
 #   Column                                 Non-Null Count  Dtype
---  ------                                 --------------  -----
 0   id                                     27901 non-null  int64
 1   Gender                                 27901 non-null  object
 2   Age                                    27901 non-null  int64
 3   City                                   27901 non-null  object
 4   Profession                             27901 non-null  object
 5   Academic Pressure                      27901 non-null  int64
 6   Work Pressure                          27901 non-null  int64
 7   CGPA                                   27901 non-null  float64
 8   Study Satisfaction                     27901 non-null  int64
 9   Job Satisfaction                       27901 non-null  int64
 10  Sleep Duration                         27901 non-null  object
 11  Dietary Habits                         27901 non-null  object
 12  Degree                                 27901 non-null  object
 13  Have you ever had suicidal thoughts ?  27901 non-null  object
 14  Work/Study Hours                       27901 non-null  int64
 15  Financial Stress                       27898 non-null  float64
```

5. How to Get Summary Statistics of the Dataset?

df.describe() # Provides mean, median, std, min, max, etc.

OUTPUT:



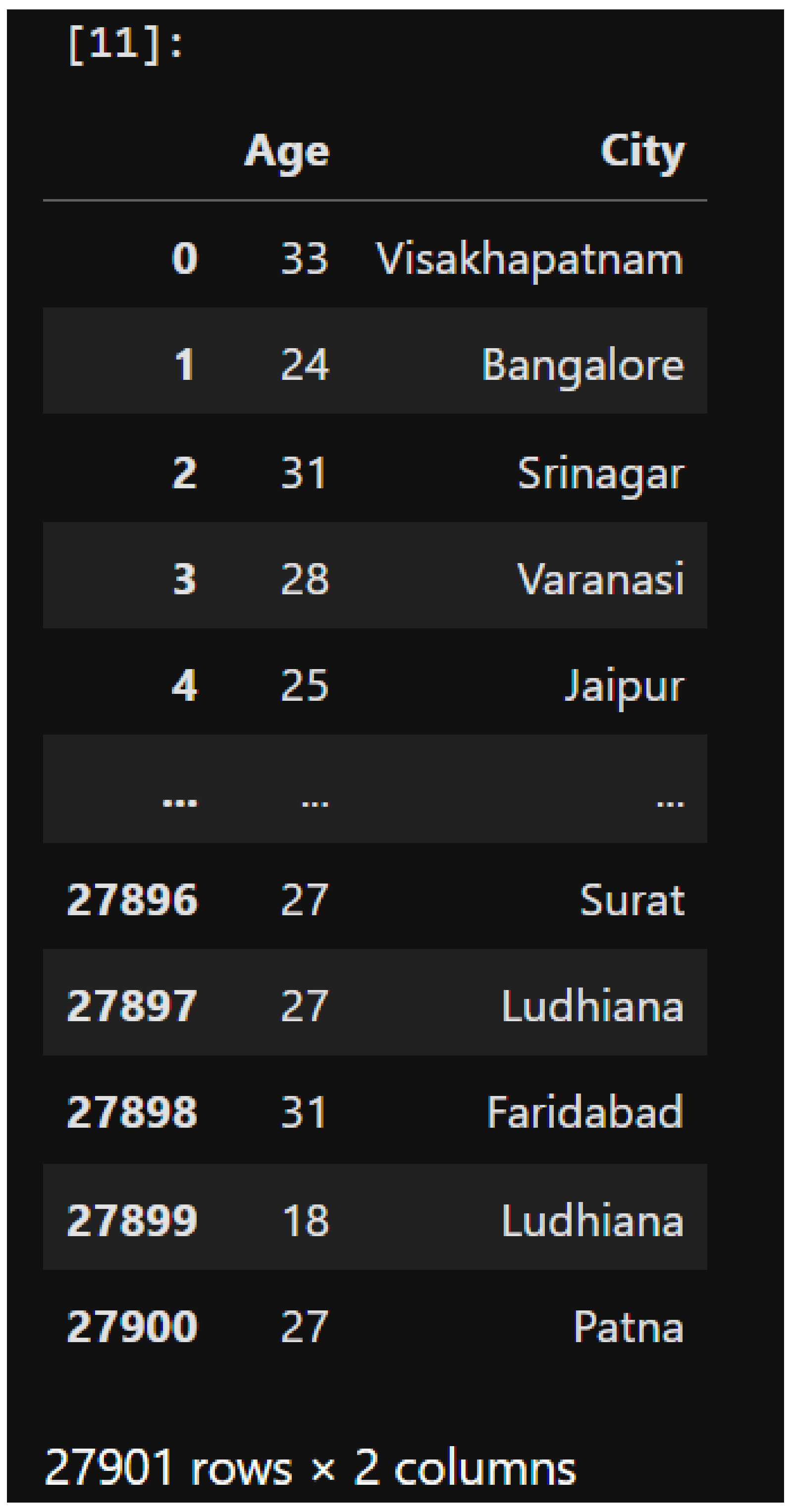
[9]:

	id	Age	Academic Pressure	Work Pressure	CGPA	Satisf
count	27901.000000	27901.000000	27901.000000	27901.000000	27901.000000	27901.
mean	70442.149421	25.822300	3.141214	0.000430	7.656104	2.9
std	40641.175216	4.905687	1.381465	0.043992	1.470707	1.3
min	2.000000	18.000000	0.000000	0.000000	0.000000	0.0
25%	35039.000000	21.000000	2.000000	0.000000	6.290000	2.0
50%	70684.000000	25.000000	3.000000	0.000000	7.770000	3.0
75%	105818.000000	30.000000	4.000000	0.000000	8.920000	4.0
max	140699.000000	59.000000	5.000000	5.000000	10.000000	5.0

6. How to Select Specific Columns (e.g., Age and City)?

df[['Age', 'City']]

OUTPUT:


[11]:

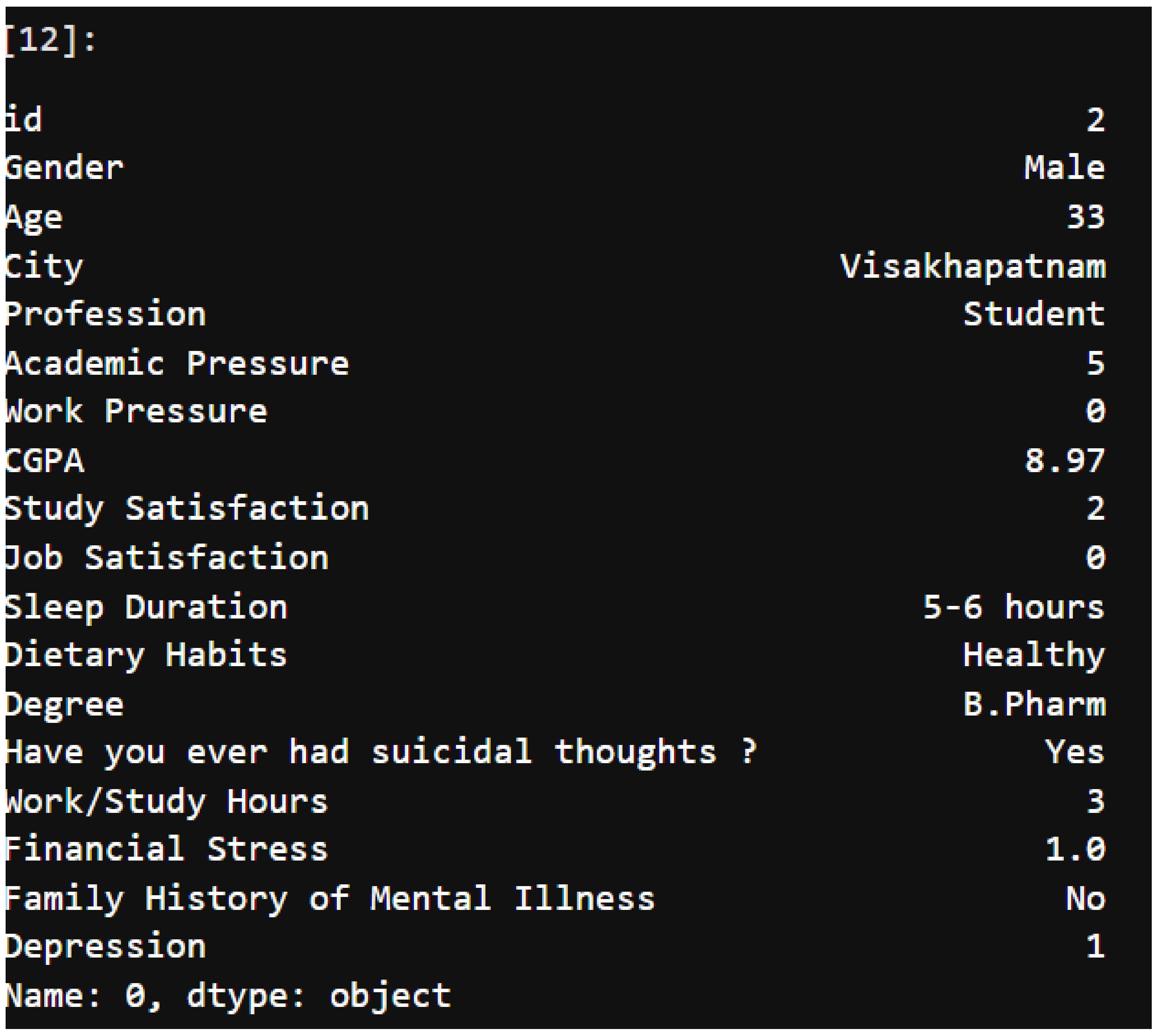
	Age	City
0	33	Visakhapatnam
1	24	Bangalore
2	31	Srinagar
3	28	Varanasi
4	25	Jaipur
...	...	...
27896	27	Surat
27897	27	Ludhiana
27898	31	Faridabad
27899	18	Ludhiana
27900	27	Patna

27901 rows × 2 columns

7. How to Select a Specific Row by Index?

df.iloc[0] # First row

OUTPUT :

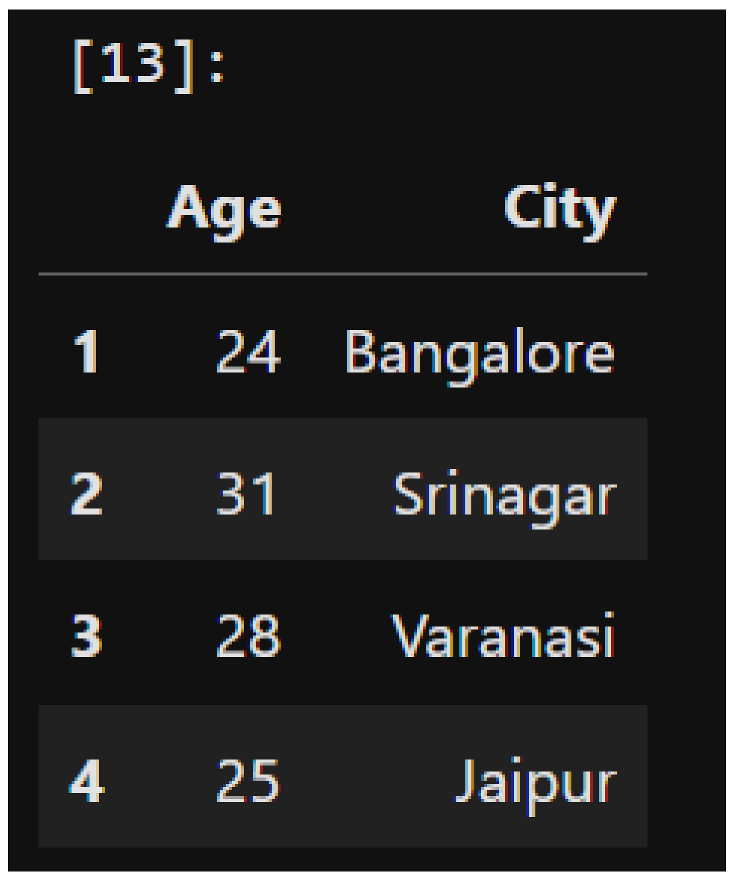


```
[12]:

id                                                 2
Gender                                          Male
Age                                               33
City                                   Visakhapatnam
Profession                                   Student
Academic Pressure                                  5
Work Pressure                                      0
CGPA                                            8.97
Study Satisfaction                                 2
Job Satisfaction                                   0
Sleep Duration                             5-6 hours
Dietary Habits                               Healthy
Degree                                       B.Pharm
Have you ever had suicidal thoughts ?            Yes
Work/Study Hours                                   3
Financial Stress                                 1.0
Family History of Mental Illness                  No
Depression                                         1
Name: 0, dtype: object
```

8. How to Select Specific Rows and Columns?

df.iloc[1:5, 2:4] # Select rows 1-4 and columns 2-3

OUTPUT:



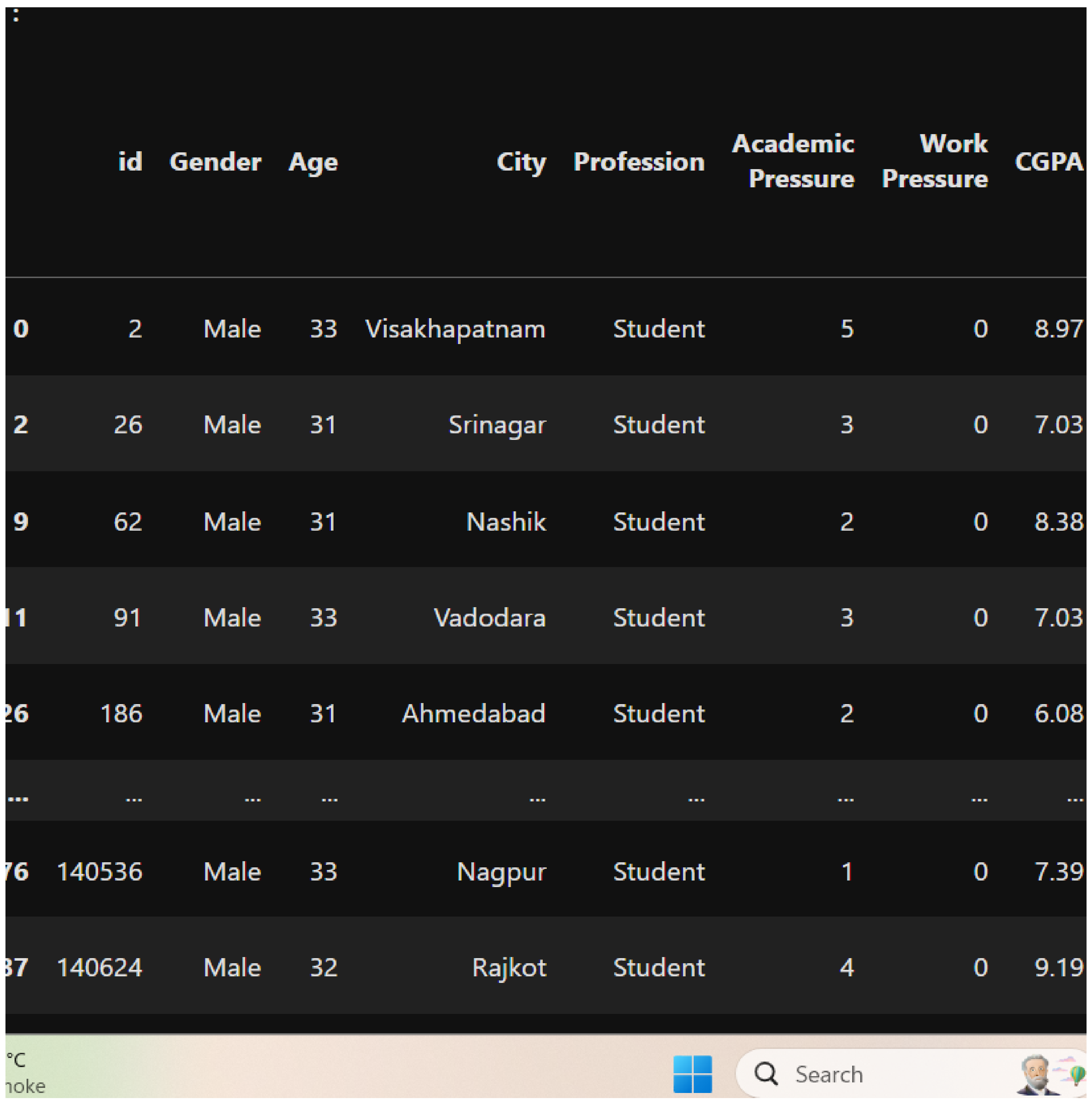
[13]:

	Age	City
1	24	Bangalore
2	31	Srinagar
3	28	Varanasi
4	25	Jaipur

9. How to Filter Students Older Than 30?

```
df[df['Age'] > 30]
```

OUTPUT:

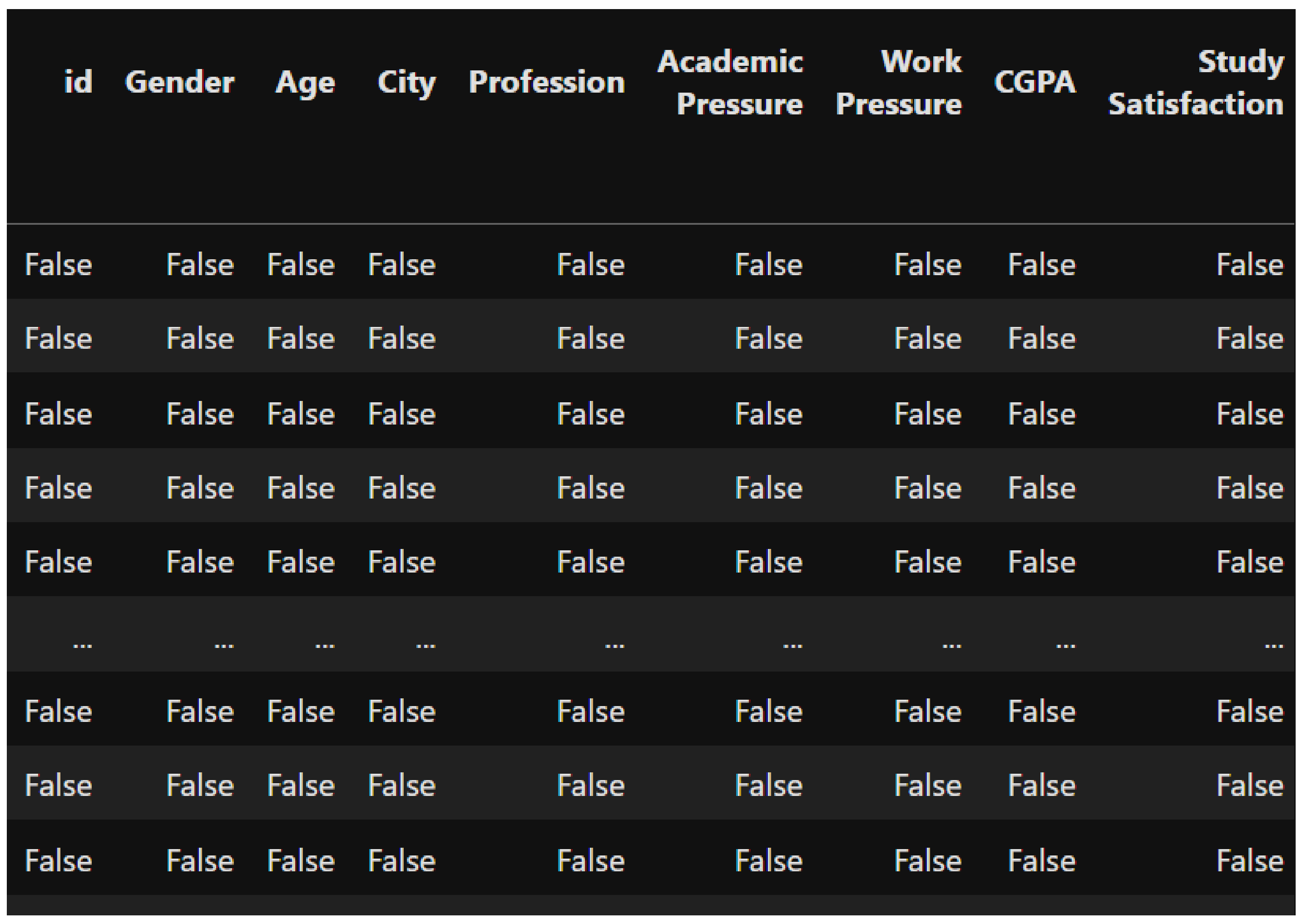


	id	Gender	Age	City	Profession	Academic Pressure	Work Pressure	CGPA
0	2	Male	33	Visakhapatnam	Student	5	0	8.97
2	26	Male	31	Srinagar	Student	3	0	7.03
9	62	Male	31	Nashik	Student	2	0	8.38
11	91	Male	33	Vadodara	Student	3	0	7.03
26	186	Male	31	Ahmedabad	Student	2	0	6.08
...	...	...	...	...	...	...	...	...
76	140536	Male	33	Nagpur	Student	1	0	7.39
37	140624	Male	32	Rajkot	Student	4	0	9.19

10. How to Check for Missing Values?

```
df.isnull().sum()  # Counts NaN values per column
```

OUTPUT:

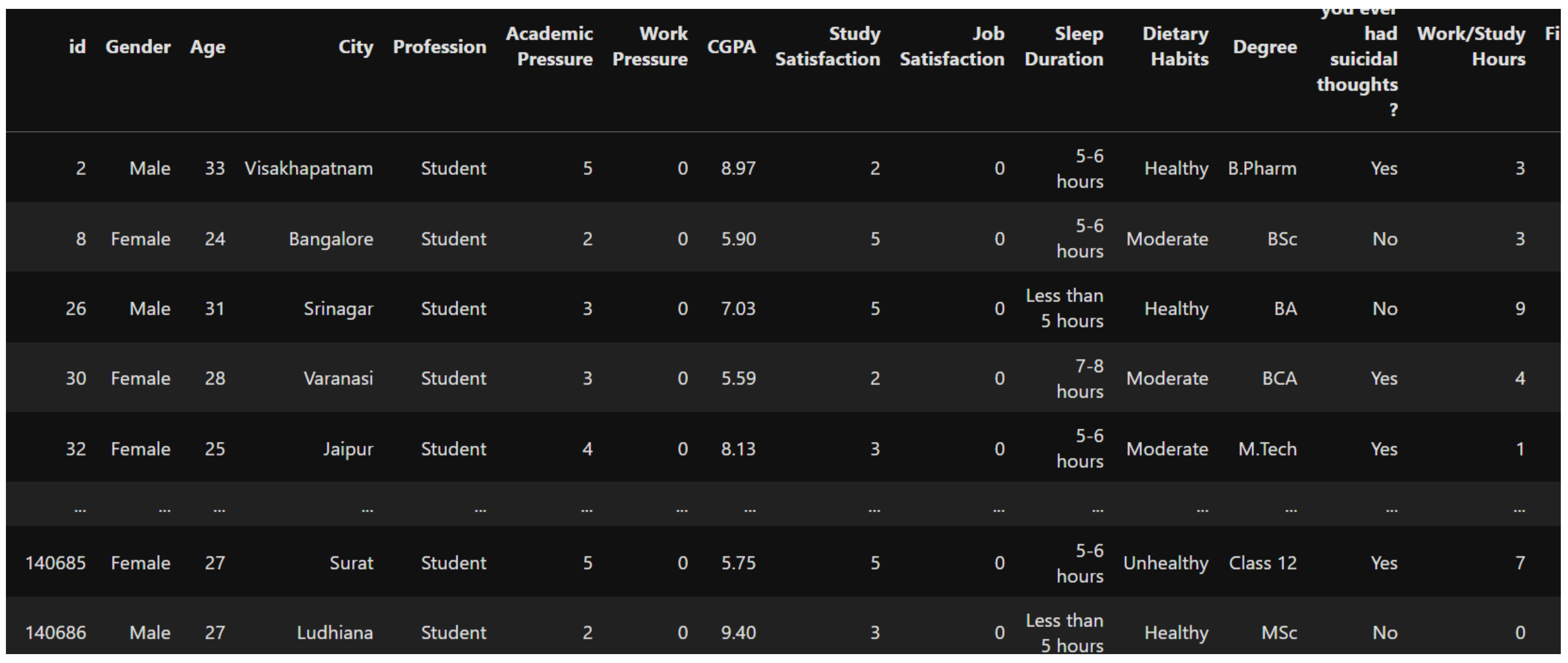


id	Gender	Age	City	Profession	Academic Pressure	Work Pressure	CGPA	Study Satisfaction
False	False	False	False	False	False	False	False	False
False	False	False	False	False	False	False	False	False
False	False	False	False	False	False	False	False	False
False	False	False	False	False	False	False	False	False
False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...
False	False	False	False	False	False	False	False	False
False	False	False	False	False	False	False	False	False
False	False	False	False	False	False	False	False	False

11.How to Drop Rows with Missing Values?

```
df.dropna(inplace=True)
```

OUTPUT:

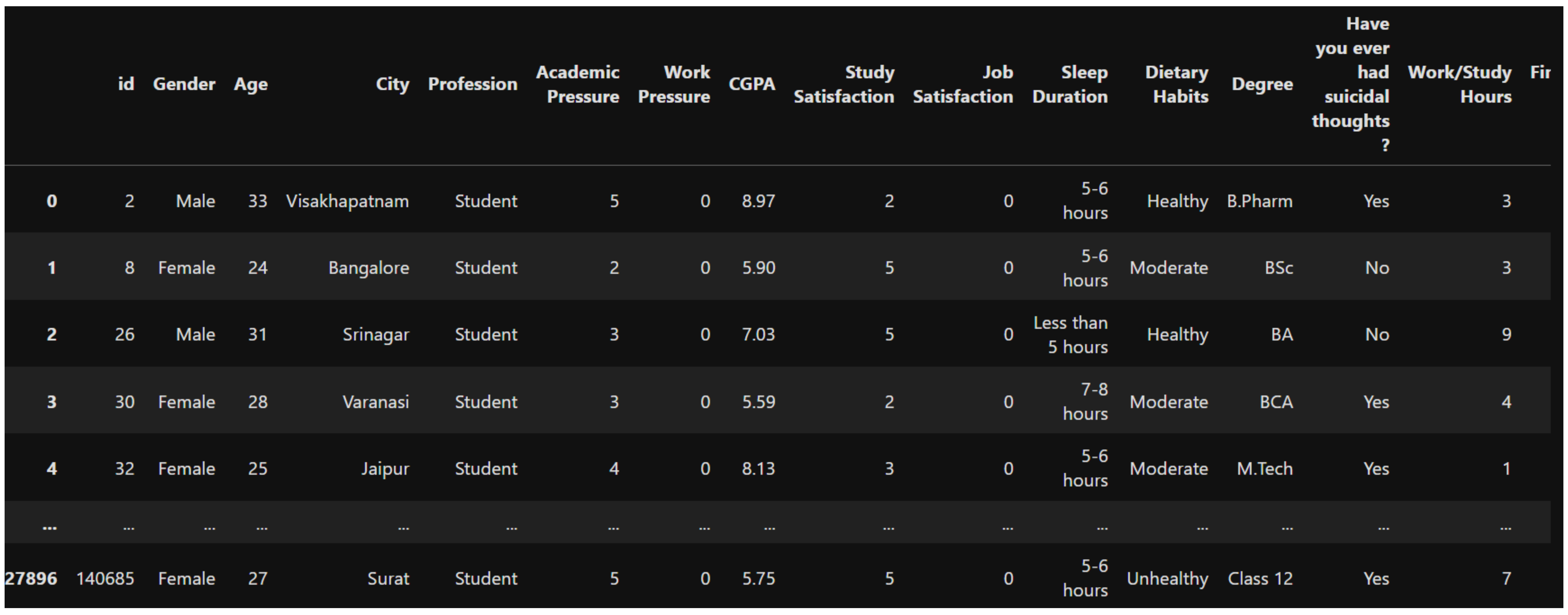


id	Gender	Age	City	Profession	Academic Pressure	Work Pressure	CGPA	Study Satisfaction	Job Satisfaction	Sleep Duration	Dietary Habits	Degree	you ever had suicidal thoughts ?	Work/Study Hours	Fi
2	Male	33	Visakhapatnam	Student	5	0	8.97	2	0	5-6 hours	Healthy	B.Pharm	Yes	3	
8	Female	24	Bangalore	Student	2	0	5.90	5	0	5-6 hours	Moderate	BSc	No	3	
26	Male	31	Srinagar	Student	3	0	7.03	5	0	Less than 5 hours	Healthy	BA	No	9	
30	Female	28	Varanasi	Student	3	0	5.59	2	0	7-8 hours	Moderate	BCA	Yes	4	
32	Female	25	Jaipur	Student	4	0	8.13	3	0	5-6 hours	Moderate	M.Tech	Yes	1	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
140685	Female	27	Surat	Student	5	0	5.75	5	0	5-6 hours	Unhealthy	Class 12	Yes	7	
140686	Male	27	Ludhiana	Student	2	0	9.40	3	0	Less than 5 hours	Healthy	MSc	No	0	

12. How to Create a New Column Using Existing Columns?

```
df['mine'] = df['id'] + df['Age']
```

OUTPUT:

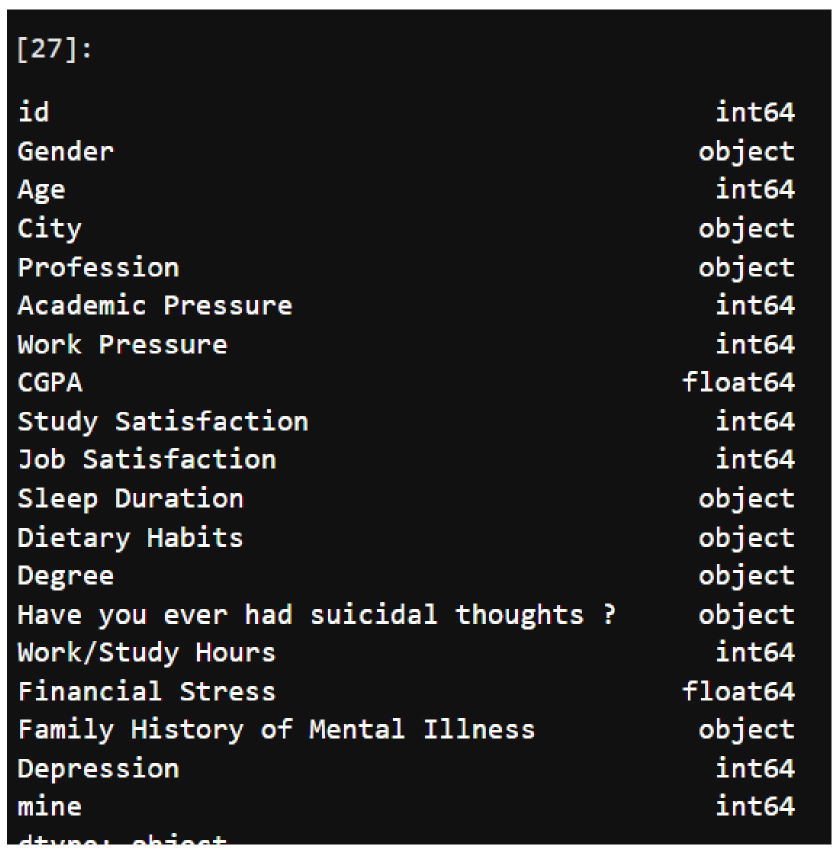


	id	Gender	Age	City	Profession	Academic Pressure	Work Pressure	CGPA	Study Satisfaction	Job Satisfaction	Sleep Duration	Dietary Habits	Degree	Have you ever had suicidal thoughts ?	Work/Study Hours	Fir
0	2	Male	33	Visakhapatnam	Student	5	0	8.97	2	0	5-6 hours	Healthy	B.Pharm	Yes	3	
1	8	Female	24	Bangalore	Student	2	0	5.90	5	0	5-6 hours	Moderate	BSc	No	3	
2	26	Male	31	Srinagar	Student	3	0	7.03	5	0	Less than 5 hours	Healthy	BA	No	9	
3	30	Female	28	Varanasi	Student	3	0	5.59	2	0	7-8 hours	Moderate	BCA	Yes	4	
4	32	Female	25	Jaipur	Student	4	0	8.13	3	0	5-6 hours	Moderate	M.Tech	Yes	1	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
27896	140685	Female	27	Surat	Student	5	0	5.75	5	0	5-6 hours	Unhealthy	Class 12	Yes	7	

13. How to Get Data Types of Each Column?

df.dtypes

OUTPUT:

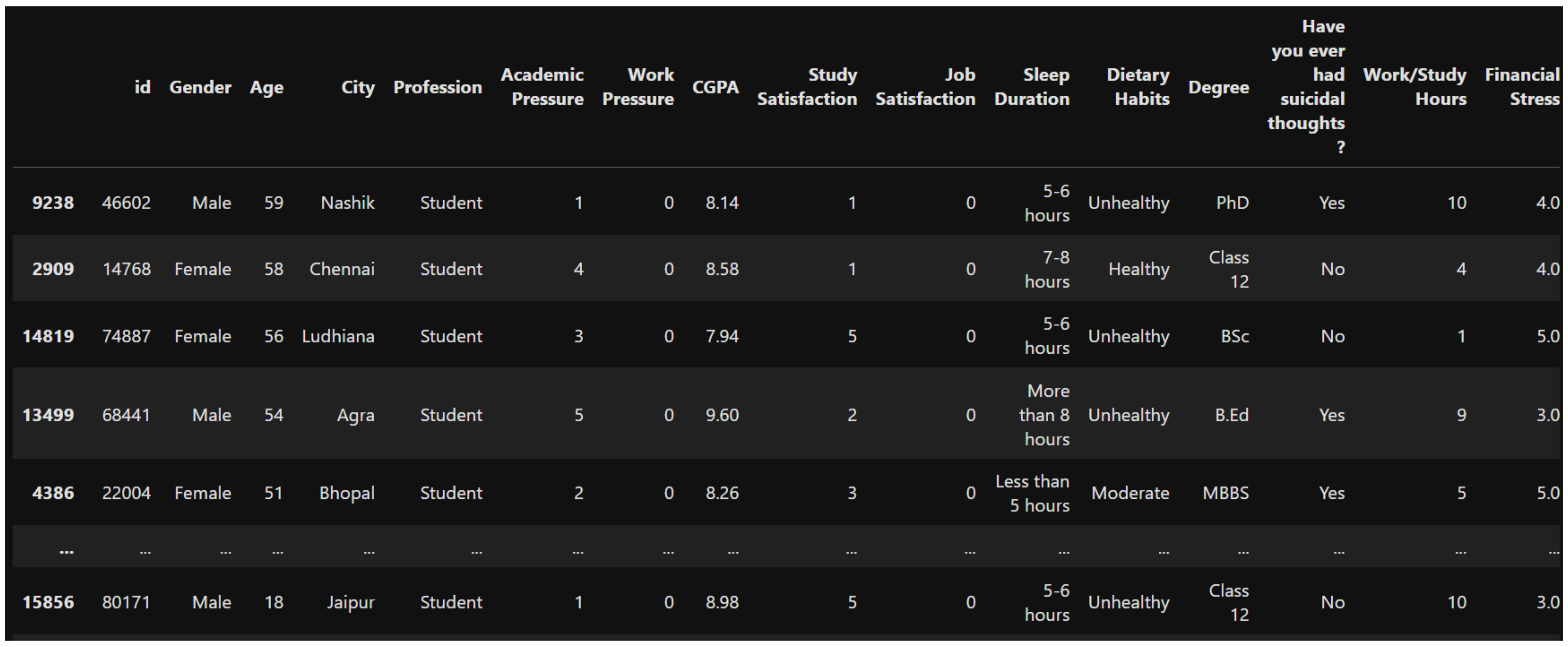


```
[27]:

id                                    int64
Gender                               object
Age                                   int64
City                                 object
Profession                           object
Academic Pressure                     int64
Work Pressure                         int64
CGPA                                float64
Study Satisfaction                    int64
Job Satisfaction                      int64
Sleep Duration                       object
Dietary Habits                       object
Degree                               object
Have you ever had suicidal thoughts ?  object
Work/Study Hours                      int64
Financial Stress                    float64
Family History of Mental Illness     object
Depression                            int64
mine                                  int64
dtype: object
```

14. How to Sort the Data by Age in Descending Order?

df.sort_values(by='Age', ascending=False)

OUTPUT:

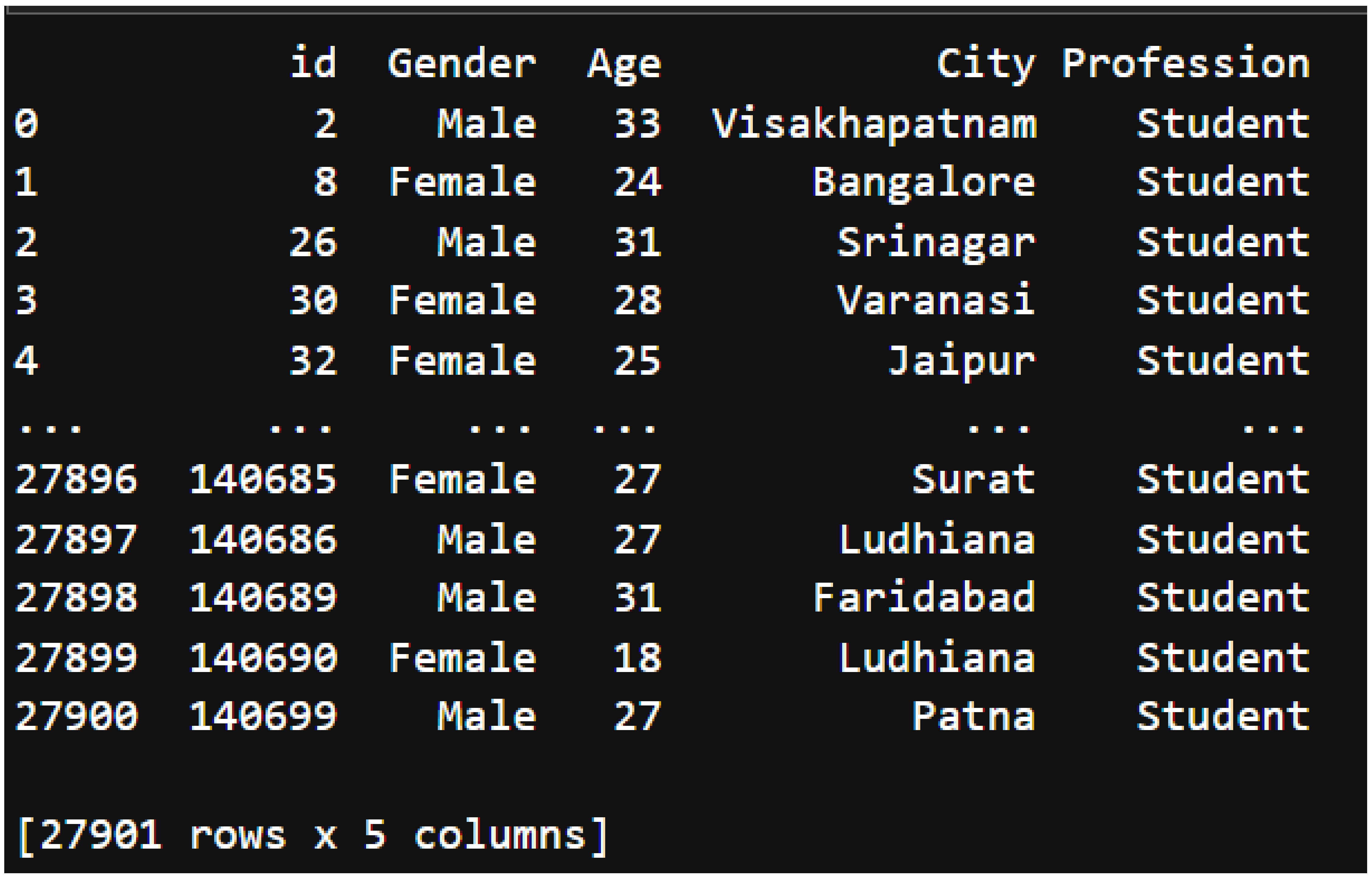


	id	Gender	Age	City	Profession	Academic Pressure	Work Pressure	CGPA	Study Satisfaction	Job Satisfaction	Sleep Duration	Dietary Habits	Degree	Have you ever had suicidal thoughts ?	Work/Study Hours	Financial Stress
9238	46602	Male	59	Nashik	Student	1	0	8.14	1	0	5-6 hours	Unhealthy	PhD	Yes	10	4.0
2909	14768	Female	58	Chennai	Student	4	0	8.58	1	0	7-8 hours	Healthy	Class 12	No	4	4.0
14819	74887	Female	56	Ludhiana	Student	3	0	7.94	5	0	5-6 hours	Unhealthy	BSc	No	1	5.0
13499	68441	Male	54	Agra	Student	5	0	9.60	2	0	More than 8 hours	Unhealthy	B.Ed	Yes	9	3.0
4386	22004	Female	51	Bhopal	Student	2	0	8.26	3	0	Less than 5 hours	Moderate	MBBS	Yes	5	5.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
15856	80171	Male	18	Jaipur	Student	1	0	8.98	5	0	5-6 hours	Unhealthy	Class 12	No	10	3.0

15. **How to Display Only the First 5 Columns?**

```
print(df.iloc[:, :5])
```

OUTPUT:

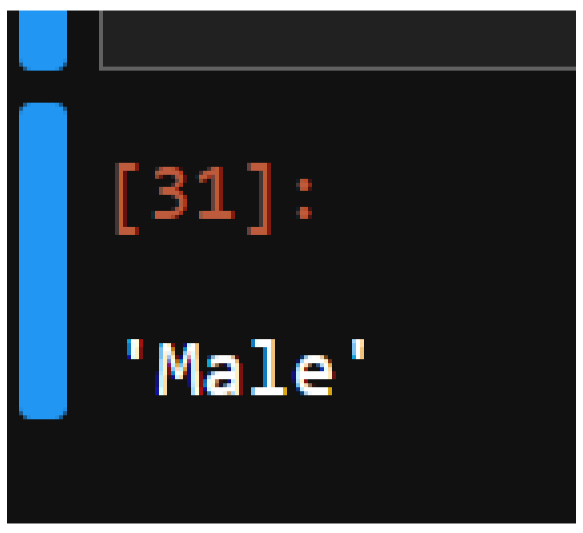


```
           id  Gender  Age           City Profession
0           2    Male   33  Visakhapatnam    Student
1           8  Female   24      Bangalore    Student
2          26    Male   31       Srinagar    Student
3          30  Female   28       Varanasi    Student
4          32  Female   25         Jaipur    Student
...       ...     ...  ...            ...        ...
27896  140685  Female   27          Surat    Student
27897  140686    Male   27       Ludhiana    Student
27898  140689    Male   31      Faridabad    Student
27899  140690  Female   18       Ludhiana    Student
27900  140699    Male   27          Patna    Student

[27901 rows x 5 columns]
```

16. **How to Select a Specific Value Using Row and Column Index?**

```
df.iloc[0, 1]  # Selects the element in first row, second column
```

OUTPUT:

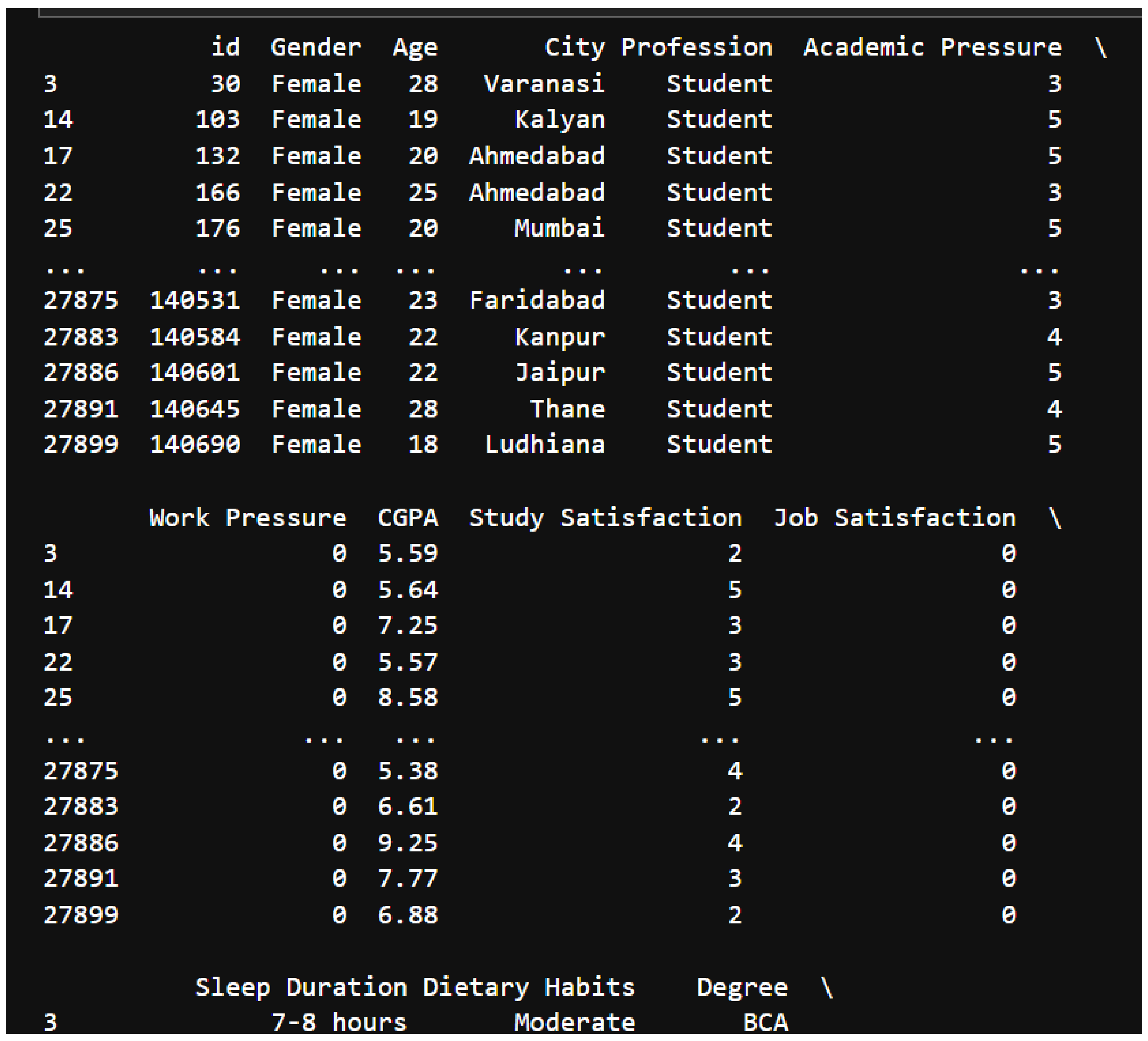


```
[31]:

'Male'
```

17. How to Find All Rows Where Gender is Female and Depression is 1?

```
female_depressed = df[(df['Gender'] == 'Female') & (df['Depression'] == 1)]
print(female_depressed)
```

OUTPUT:

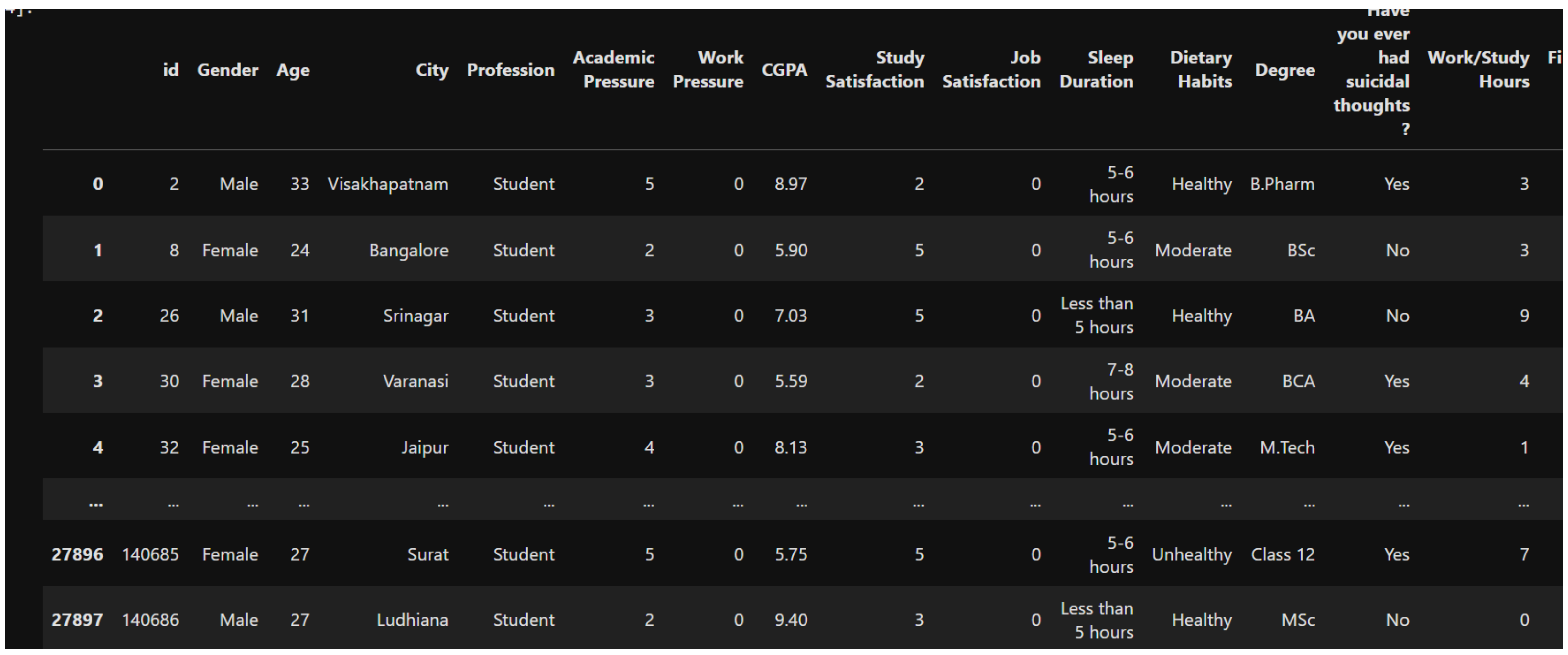


```
           id  Gender  Age       City Profession  Academic Pressure  \
3          30  Female   28   Varanasi    Student                  3
14        103  Female   19     Kalyan    Student                  5
17        132  Female   20  Ahmedabad    Student                  5
22        166  Female   25  Ahmedabad    Student                  3
25        176  Female   20     Mumbai    Student                  5
...       ...     ...  ...        ...        ...                ...
27875  140531  Female   23  Faridabad    Student                  3
27883  140584  Female   22     Kanpur    Student                  4
27886  140601  Female   22     Jaipur    Student                  5
27891  140645  Female   28      Thane    Student                  4
27899  140690  Female   18   Ludhiana    Student                  5

       Work Pressure  CGPA  Study Satisfaction  Job Satisfaction  \
3                  0  5.59                   2                 0
14                 0  5.64                   5                 0
17                 0  7.25                   3                 0
22                 0  5.57                   3                 0
25                 0  8.58                   5                 0
...              ...   ...                 ...               ...
27875              0  5.38                   4                 0
27883              0  6.61                   2                 0
27886              0  9.25                   4                 0
27891              0  7.77                   3                 0
27899              0  6.88                   2                 0

          Sleep Duration Dietary Habits    Degree  \
3             7-8 hours       Moderate       BCA
```

18. How to Count Unique Values in the "City" Column?

```
unique_cities = df['City'].nunique()
print(f"Unique cities: {unique_cities}")
```

OUTPUT:

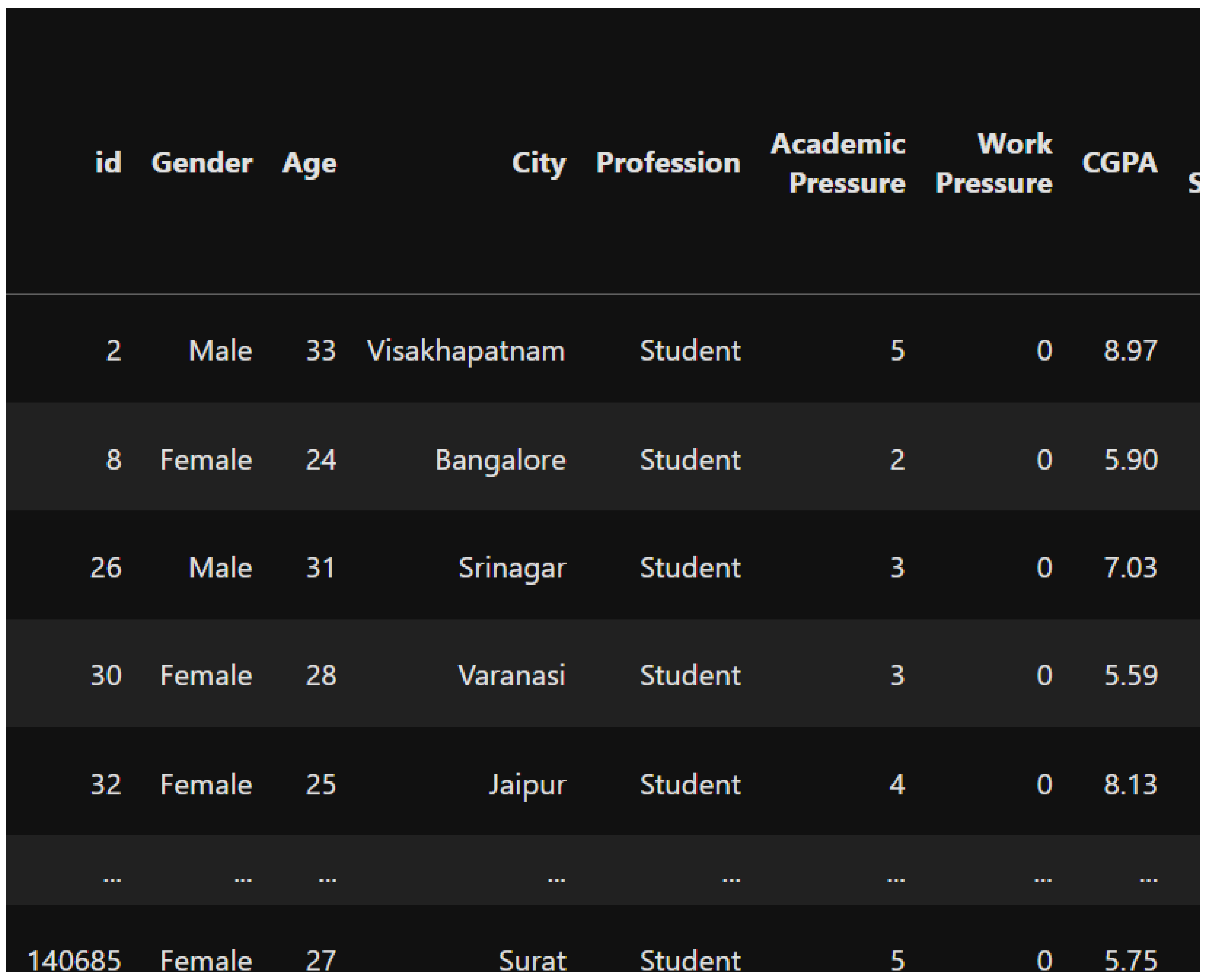


	id	Gender	Age	City	Profession	Academic Pressure	Work Pressure	CGPA	Study Satisfaction	Job Satisfaction	Sleep Duration	Dietary Habits	Degree	Have you ever had suicidal thoughts ?	Work/Study Hours	Fi
0	2	Male	33	Visakhapatnam	Student	5	0	8.97	2	0	5-6 hours	Healthy	B.Pharm	Yes	3	
1	8	Female	24	Bangalore	Student	2	0	5.90	5	0	5-6 hours	Moderate	BSc	No	3	
2	26	Male	31	Srinagar	Student	3	0	7.03	5	0	Less than 5 hours	Healthy	BA	No	9	
3	30	Female	28	Varanasi	Student	3	0	5.59	2	0	7-8 hours	Moderate	BCA	Yes	4	
4	32	Female	25	Jaipur	Student	4	0	8.13	3	0	5-6 hours	Moderate	M.Tech	Yes	1	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
27896	140685	Female	27	Surat	Student	5	0	5.75	5	0	5-6 hours	Unhealthy	Class 12	Yes	7	
27897	140686	Male	27	Ludhiana	Student	2	0	9.40	3	0	Less than 5 hours	Healthy	MSc	No	0	

19.How to Fill Missing Values with 0?

```
df.fillna(0, inplace=True)
```

OUTPUT:

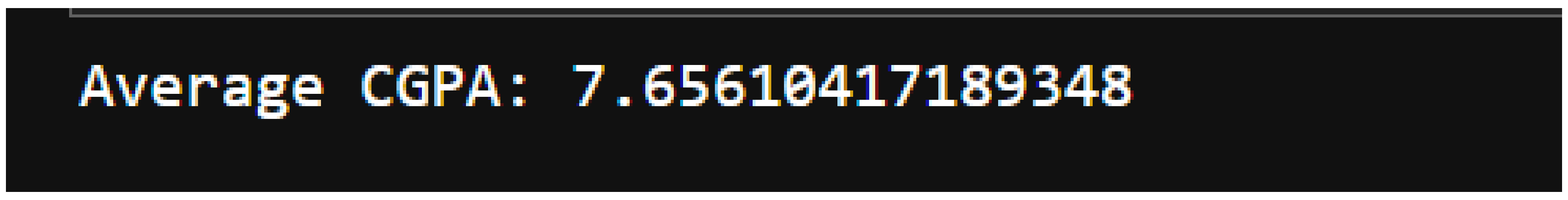


```
      id  Gender  Age           City  Profession  Academic  Work      CGPA
                                                  Pressure  Pressure
       2    Male   33  Visakhapatnam     Student         5         0  8.97
       8  Female   24      Bangalore     Student         2         0  5.90
      26    Male   31       Srinagar     Student         3         0  7.03
      30  Female   28       Varanasi     Student         3         0  5.59
      32  Female   25         Jaipur     Student         4         0  8.13
     ...     ...  ...            ...         ...       ...       ...   ...
  140685  Female   27          Surat     Student         5         0  5.75
```

20.How to Calculate the Average CGPA?

```
average_cgpa = df['CGPA'].mean()
print(f"Average CGPA: {average_cgpa}")
```

OUTPUT:

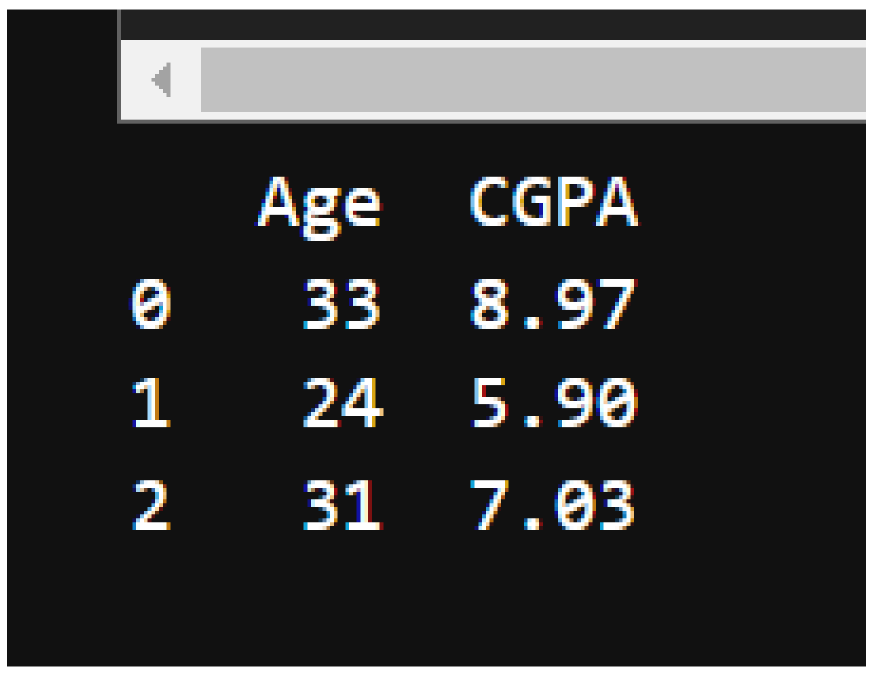


```
Average CGPA: 7.65610417189348
```

21.How to Select the First 3 Rows of Age and CGPA?

```
df.loc[:2, ['Age', 'CGPA']]
```

OUTPUT:

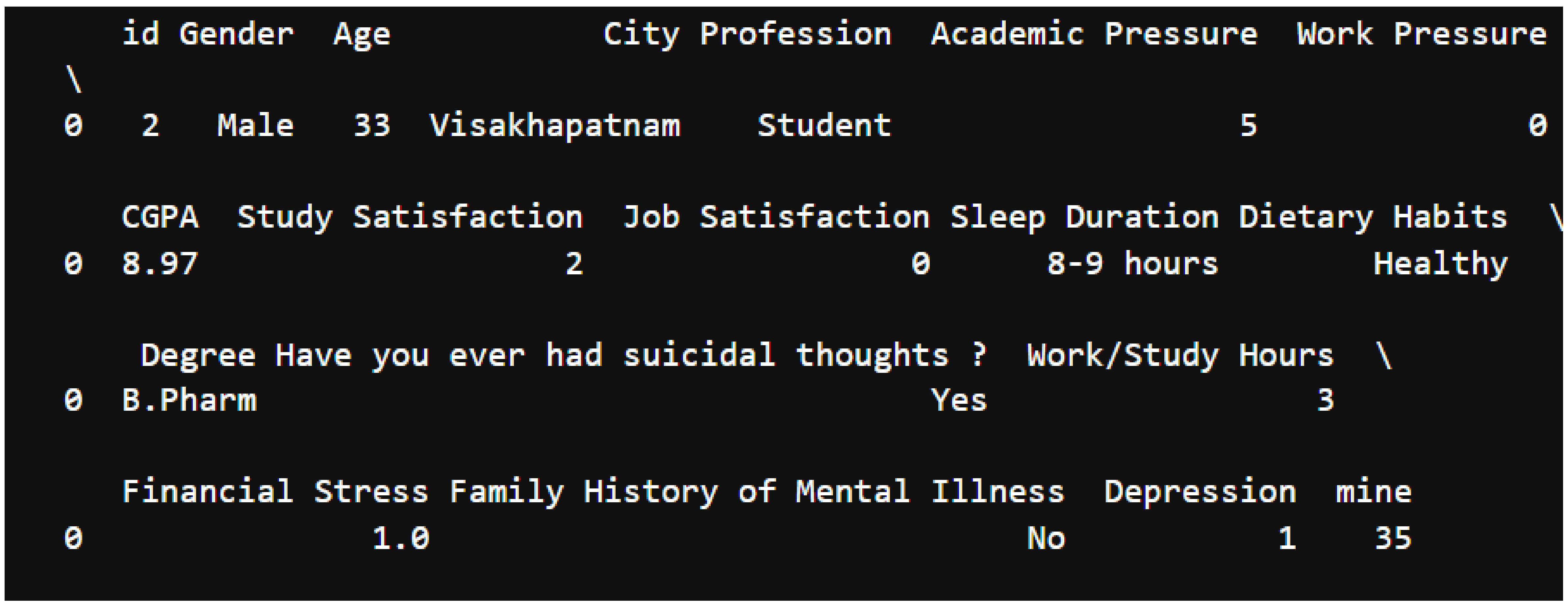


```
   Age  CGPA
0   33  8.97
1   24  5.90
2   31  7.03
```

22. How to Update a Student's Sleep Duration (id=2 to "8-9 hours")?

```
df.loc[df['id'] == 2, 'Sleep Duration'] = "8-9 hours"
print(df.loc[df['id'] == 2])
```

OUTPUT:

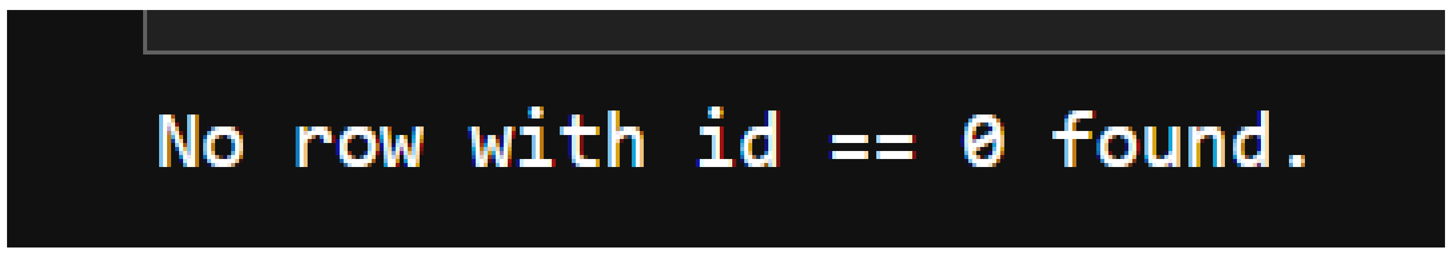


```
   id Gender  Age          City Profession  Academic Pressure  Work Pressure  \
0   2   Male   33  Visakhapatnam    Student                  5              0

   CGPA  Study Satisfaction  Job Satisfaction Sleep Duration Dietary Habits  \
0  8.97                   2                 0      8-9 hours        Healthy

    Degree Have you ever had suicidal thoughts ?  Work/Study Hours  \
0  B.Pharm                                    Yes                 3

   Financial Stress Family History of Mental Illness  Depression  mine
0               1.0                                No           1    35
```

23. How to Check If a Row with id=0 Exists and Modify Its Sleep Duration?

```
if (df['id'] == 0).any():
df.loc[df['id'] == 0, 'Sleep Duration'] = "8-9 hours"
print(df.loc[df['id'] == 0])
else:
print("No row with id == 0 found.")
```

OUTPUT:



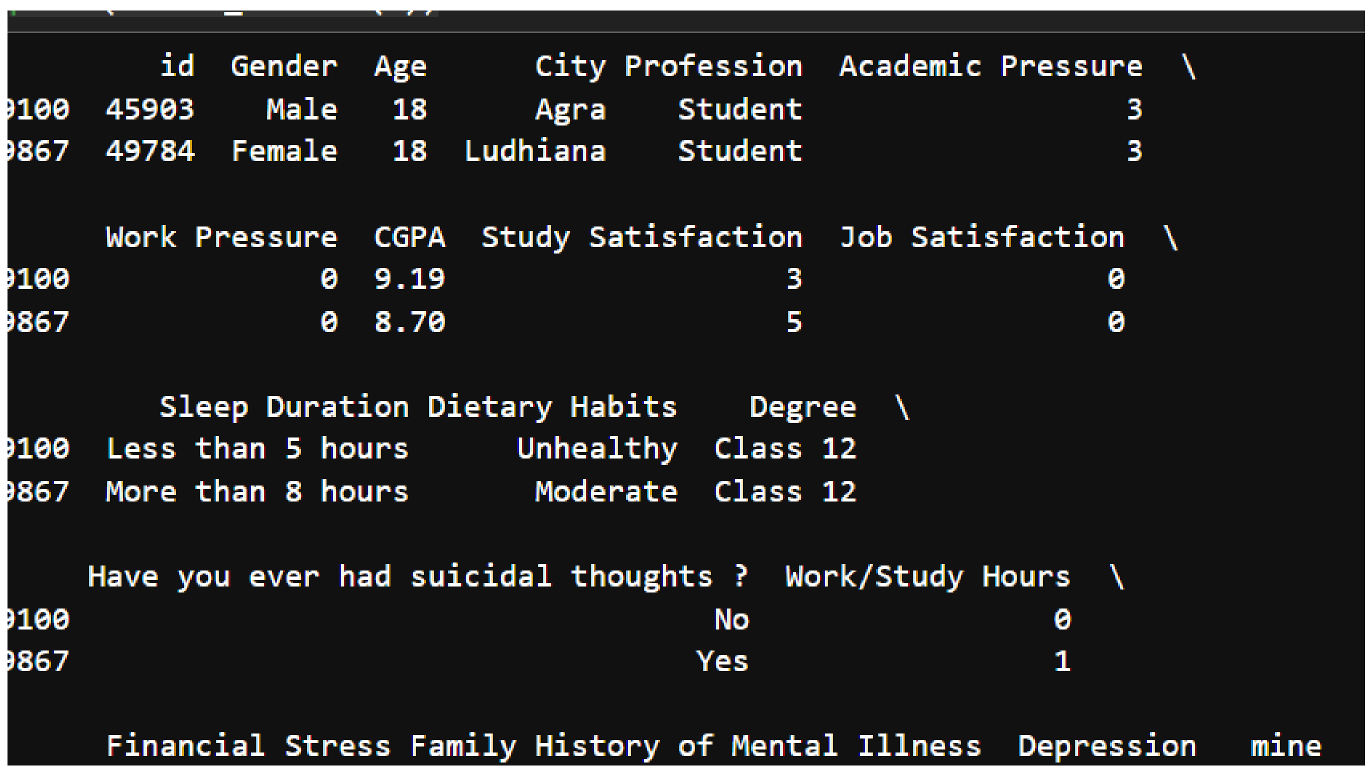
```
No row with id == 0 found.
```

24. How to Sort the Data by Age in Ascending Order?

sorted_df = df.sort_values(by='Age')

print(sorted_df.head(2))

OUTPUT:

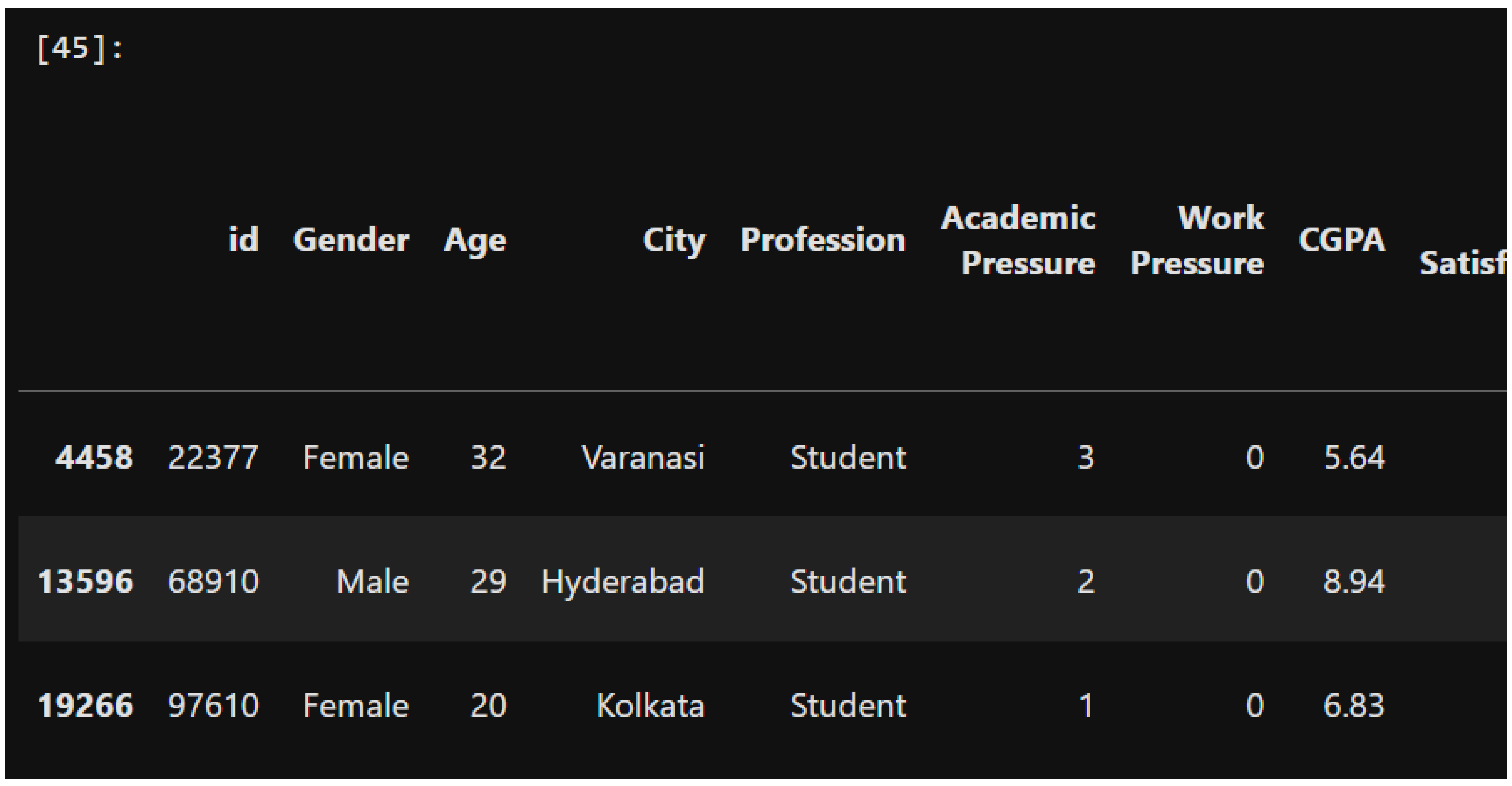


```
       id  Gender  Age      City Profession  Academic Pressure  \
100  45903    Male   18      Agra    Student                  3
867  49784  Female   18  Ludhiana    Student                  3

     Work Pressure  CGPA  Study Satisfaction  Job Satisfaction  \
100              0  9.19                   3                 0
867              0  8.70                   5                 0

        Sleep Duration Dietary Habits    Degree  \
100  Less than 5 hours      Unhealthy  Class 12
867  More than 8 hours       Moderate  Class 12

    Have you ever had suicidal thoughts ?  Work/Study Hours  \
100                                    No                 0
867                                   Yes                 1

     Financial Stress Family History of Mental Illness  Depression   mine
```

25.How to Find Rows with Missing or Empty Values?

df[df.isnull().any(axis=1) | (df == '').any(axis=1)]

OUTPUT:



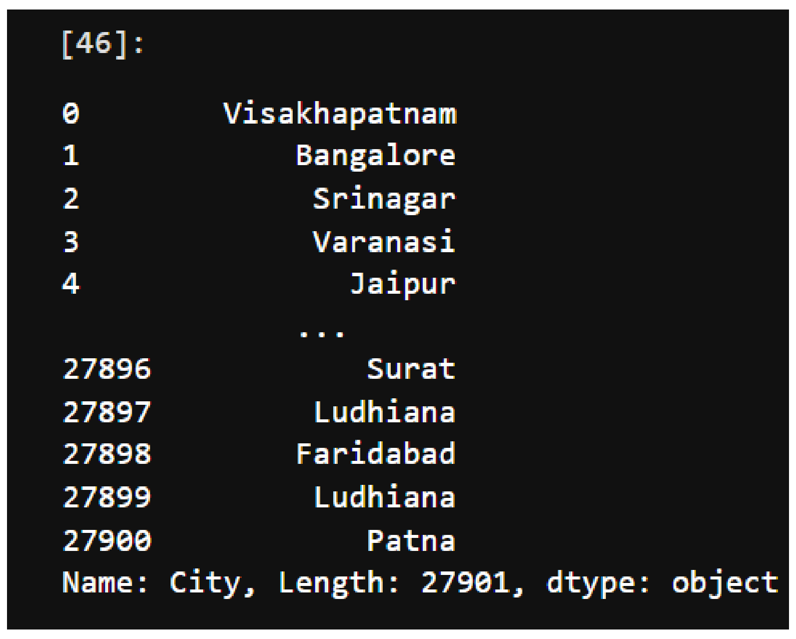
[45]:

	id	Gender	Age	City	Profession	Academic Pressure	Work Pressure	CGPA	Satisf
4458	22377	Female	32	Varanasi	Student	3	0	5.64	
13596	68910	Male	29	Hyderabad	Student	2	0	8.94	
19266	97610	Female	20	Kolkata	Student	1	0	6.83	

26.How to Fill Missing Values in the "City" Column with the Mode of "Financial Stress"?

```
df['City'].fillna(df['Financial Stress'].mode()[0])  # Filling NaN with mode
```

OUTPUT:

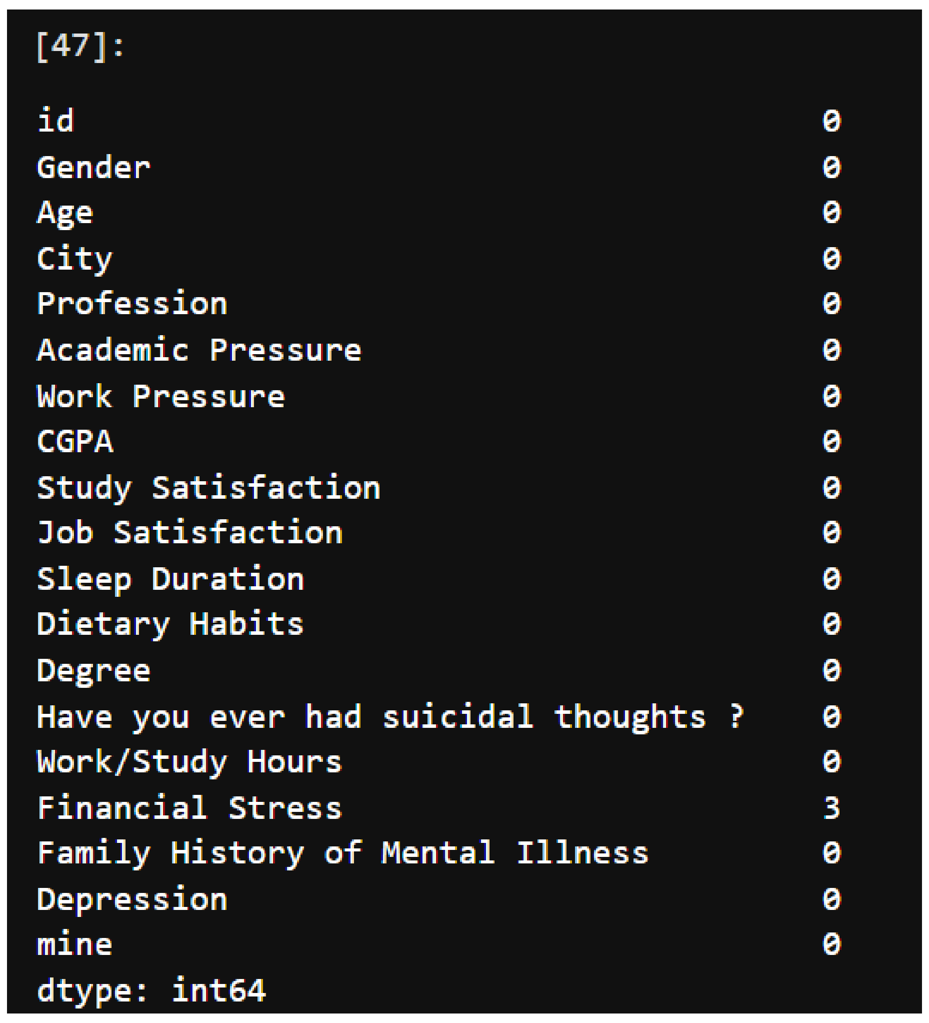


```
[46]:

0        Visakhapatnam
1            Bangalore
2             Srinagar
3             Varanasi
4               Jaipur
             ...
27896            Surat
27897         Ludhiana
27898        Faridabad
27899         Ludhiana
27900            Patna
Name: City, Length: 27901, dtype: object
```

27. How to Check the Count of Missing Values in Each Column?

```
df.isnull().sum()
```

OUTPUT:

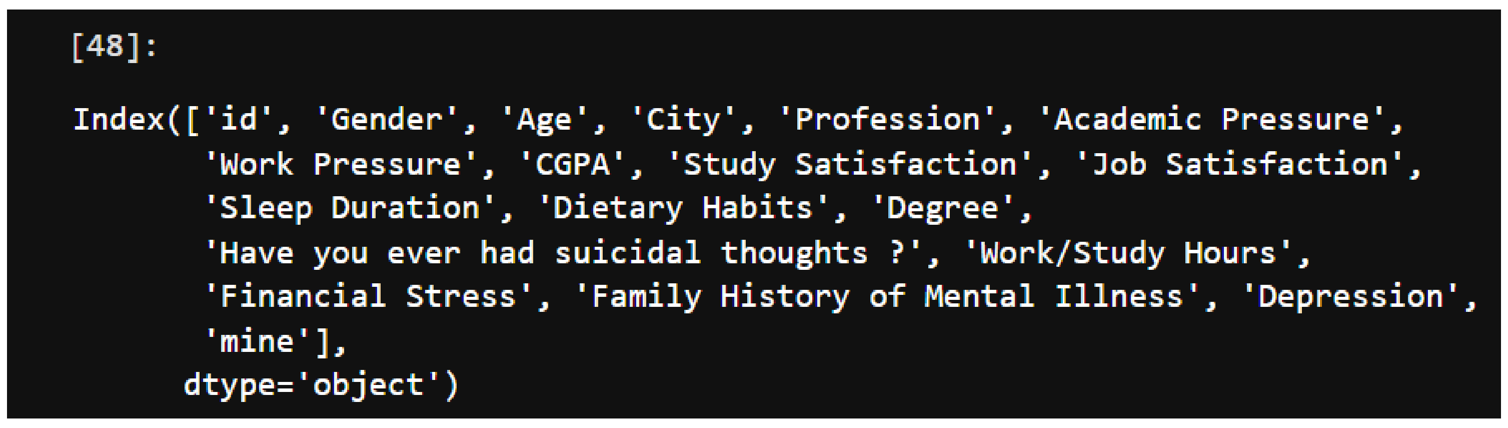


```
[47]:

id                                       0
Gender                                   0
Age                                      0
City                                     0
Profession                               0
Academic Pressure                        0
Work Pressure                            0
CGPA                                     0
Study Satisfaction                       0
Job Satisfaction                         0
Sleep Duration                           0
Dietary Habits                           0
Degree                                   0
Have you ever had suicidal thoughts ?    0
Work/Study Hours                         0
Financial Stress                         3
Family History of Mental Illness         0
Depression                               0
mine                                     0
dtype: int64
```

28.How to Display All Column Names in the Dataset?

```
df.columns
```

OUTPUT:

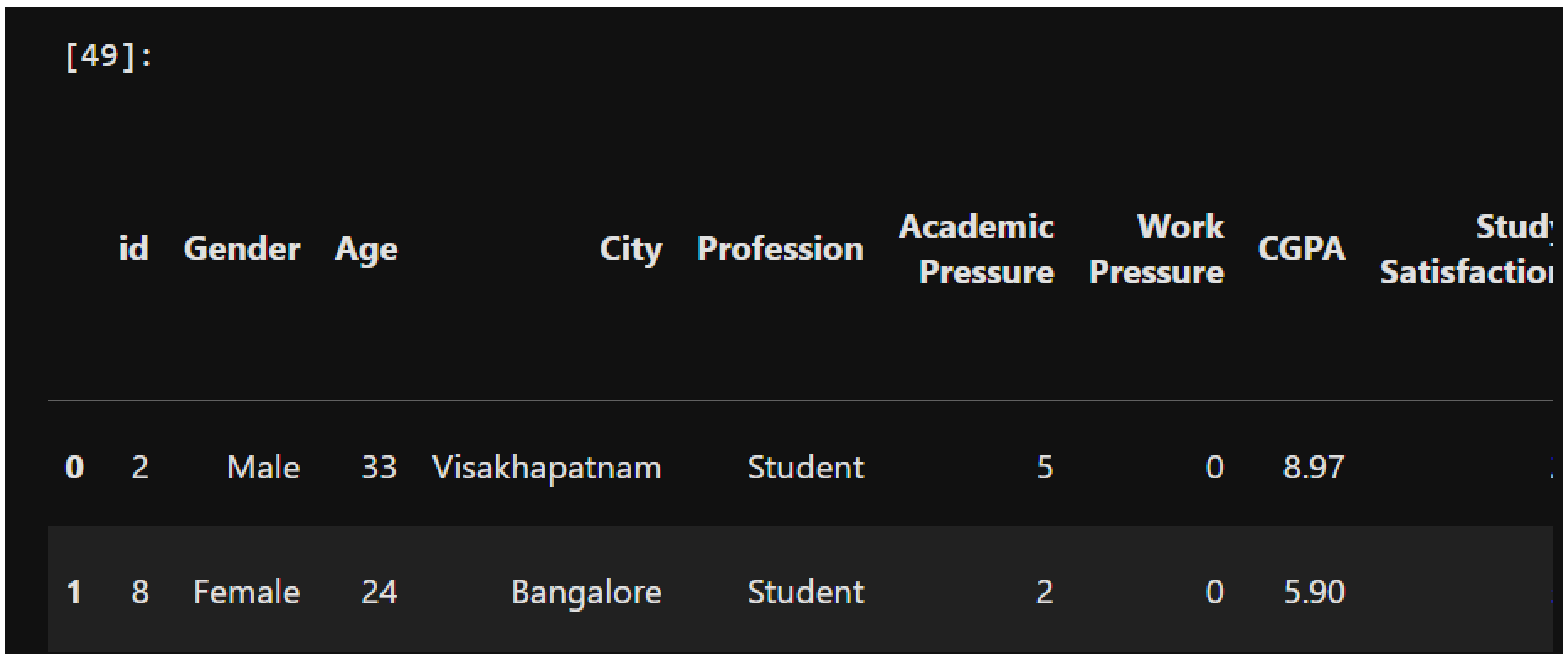


```
[48]:

Index(['id', 'Gender', 'Age', 'City', 'Profession', 'Academic Pressure',
       'Work Pressure', 'CGPA', 'Study Satisfaction', 'Job Satisfaction',
       'Sleep Duration', 'Dietary Habits', 'Degree',
       'Have you ever had suicidal thoughts ?', 'Work/Study Hours',
       'Financial Stress', 'Family History of Mental Illness', 'Depression',
       'mine'],
      dtype='object')
```

29.How to Select Multiple Rows by Index?

df.iloc[[0,1]] # Selects the first and second row

OUTPUT:



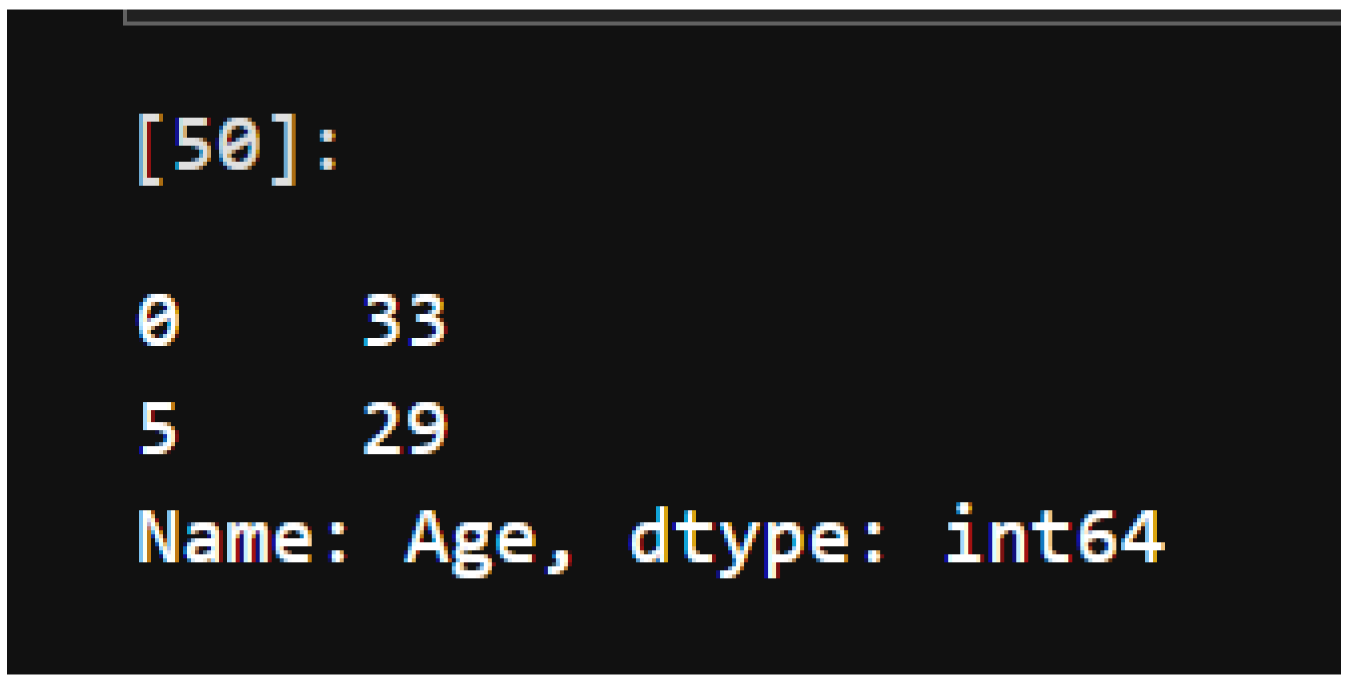
```
[49]:
```

	id	Gender	Age	City	Profession	Academic Pressure	Work Pressure	CGPA	Stud Satisfactio
0	2	Male	33	Visakhapatnam	Student	5	0	8.97	
1	8	Female	24	Bangalore	Student	2	0	5.90	

30.How to Select a Specific Value Using Row and Column Index?

df.iloc[[0,5],2] # Selects the value at row indices 0 and 5, column index 2

OUTPUT:

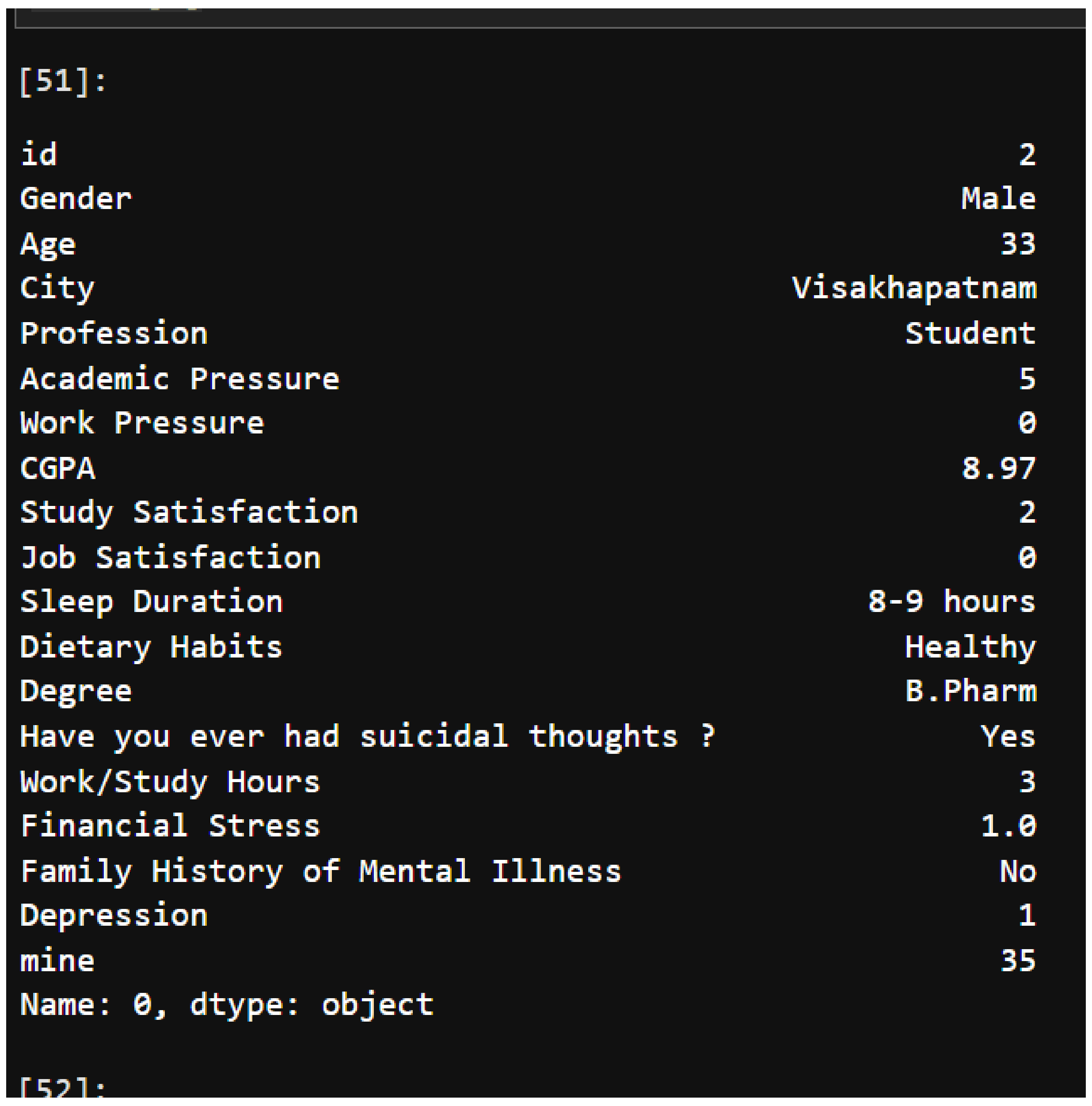


```
[50]:

0    33
5    29
Name: Age, dtype: int64
```

31.How to Select a Row by Index Using .loc[]?

df.loc[0] # Selects the row where index = 0

OUTPUT:

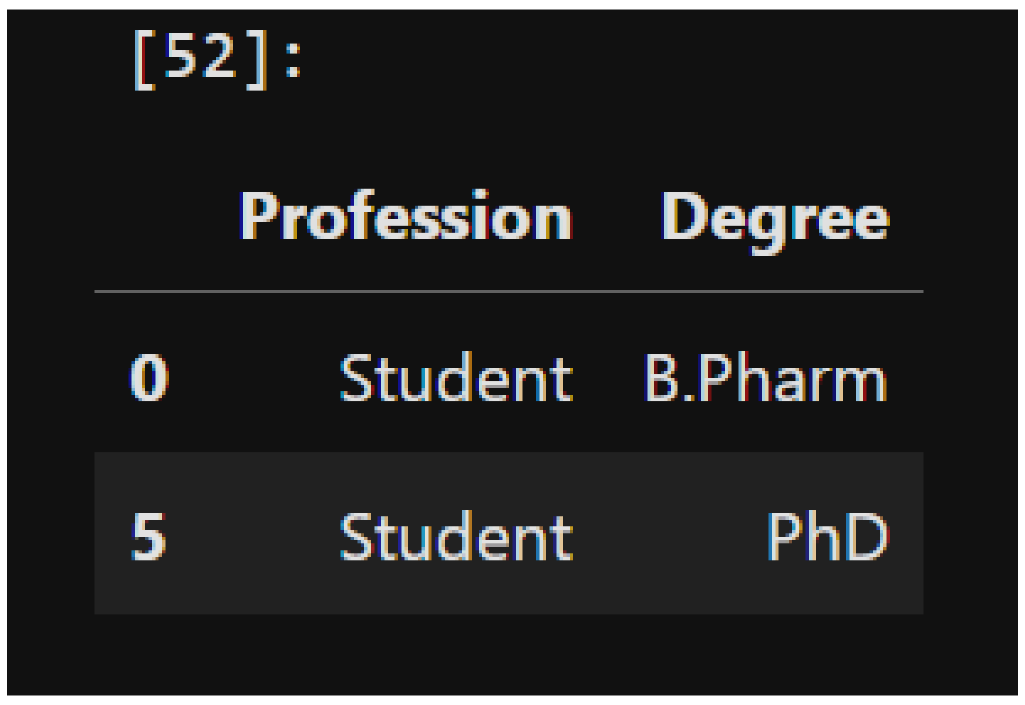


```
[51]:

id                                            2
Gender                                     Male
Age                                          33
City                              Visakhapatnam
Profession                              Student
Academic Pressure                             5
Work Pressure                                 0
CGPA                                       8.97
Study Satisfaction                            2
Job Satisfaction                              0
Sleep Duration                        8-9 hours
Dietary Habits                          Healthy
Degree                                  B.Pharm
Have you ever had suicidal thoughts ?       Yes
Work/Study Hours                              3
Financial Stress                            1.0
Family History of Mental Illness             No
Depression                                    1
mine                                         35
Name: 0, dtype: object
```

32. How to Select Specific Rows and Columns Using .loc[]?

df.loc[[0,5],['Profession','Degree']]

OUTPUT:



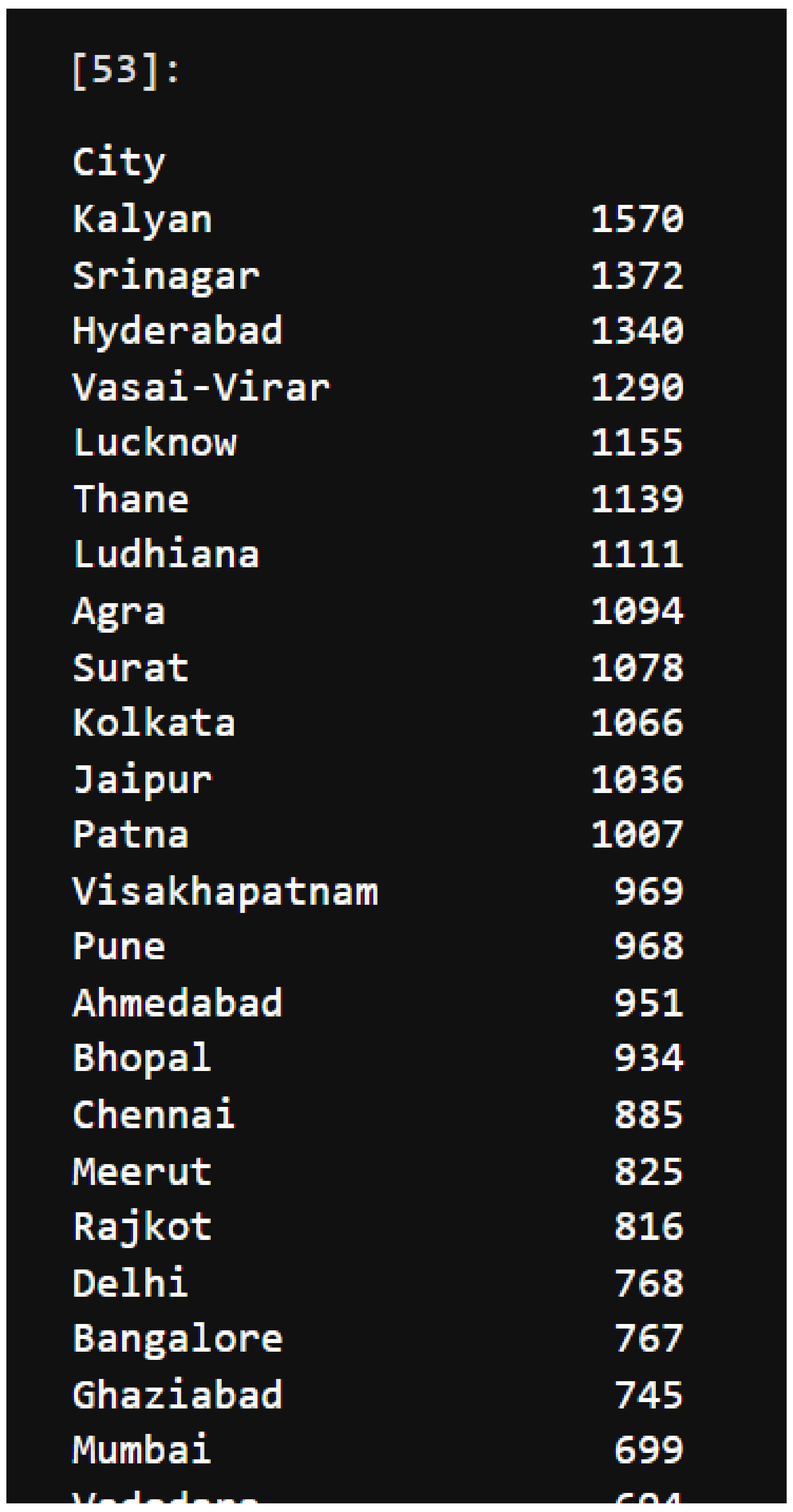
[52]:

	Profession	Degree
0	Student	B.Pharm
5	Student	PhD

33.How to Get the Count of Unique Values in the "City" Column?

df['City'].value_counts()

OUTPUT:

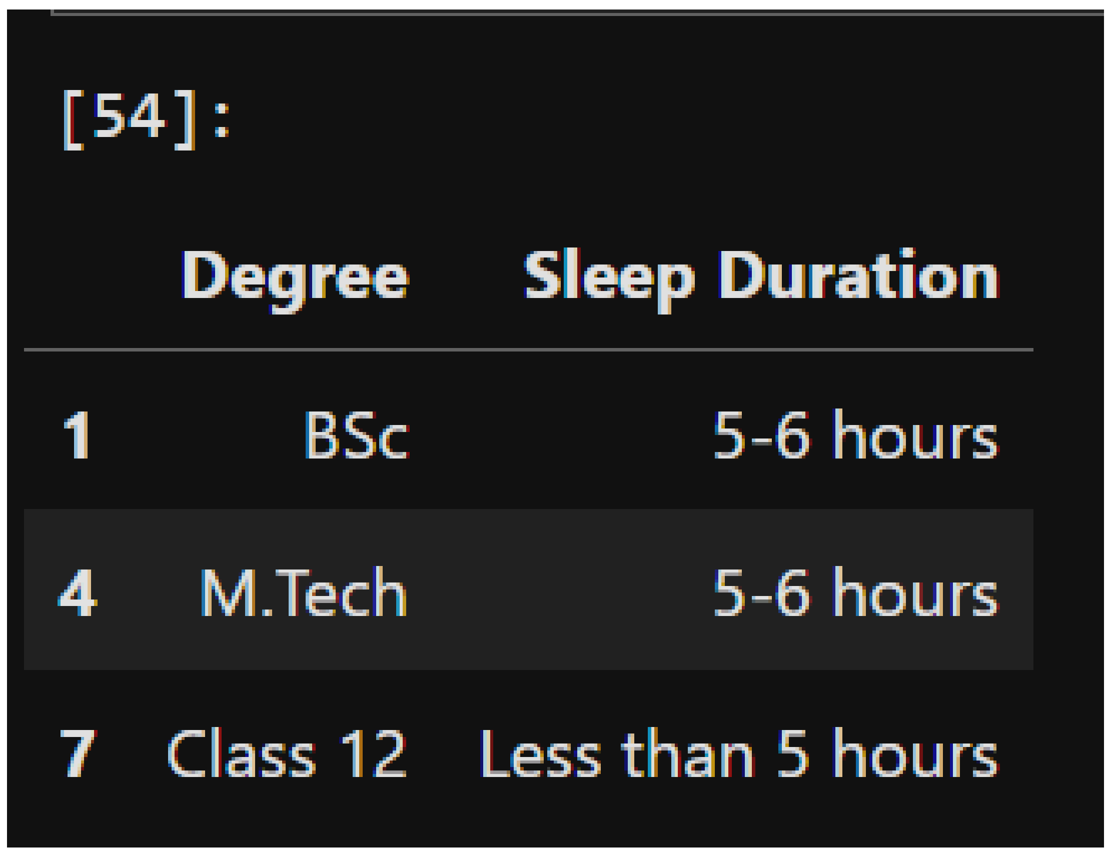


```
[53]:

City
Kalyan                  1570
Srinagar                1372
Hyderabad               1340
Vasai-Virar             1290
Lucknow                 1155
Thane                   1139
Ludhiana                1111
Agra                    1094
Surat                   1078
Kolkata                 1066
Jaipur                  1036
Patna                   1007
Visakhapatnam            969
Pune                     968
Ahmedabad                951
Bhopal                   934
Chennai                  885
Meerut                   825
Rajkot                   816
Delhi                    768
Bangalore                767
Ghaziabad                745
Mumbai                   699
```

34. How to Select Specific Rows and Columns?

df.loc[[1,4,7],['Degree','Sleep Duration']]

OUTPUT:



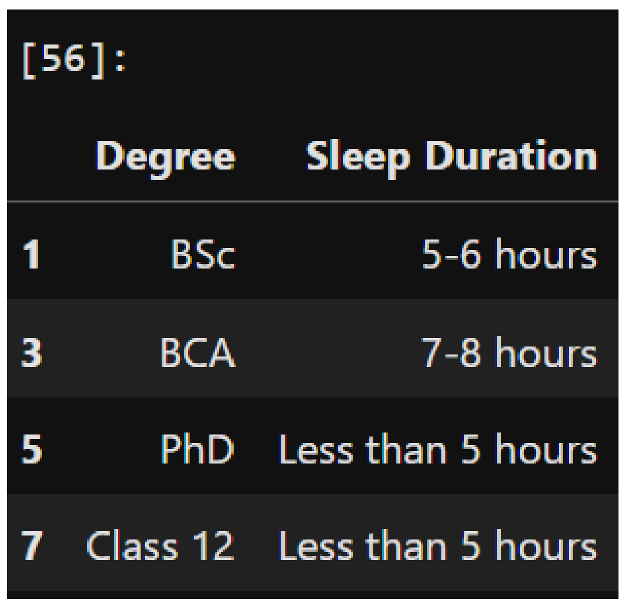
[54]:

	Degree	Sleep Duration
1	BSc	5-6 hours
4	M.Tech	5-6 hours
7	Class 12	Less than 5 hours

35. How to Select a Range of Rows with Specific Columns Using .loc[]?

df.loc[1:8:2, ['Degree','Sleep Duration']]

OUTPUT:



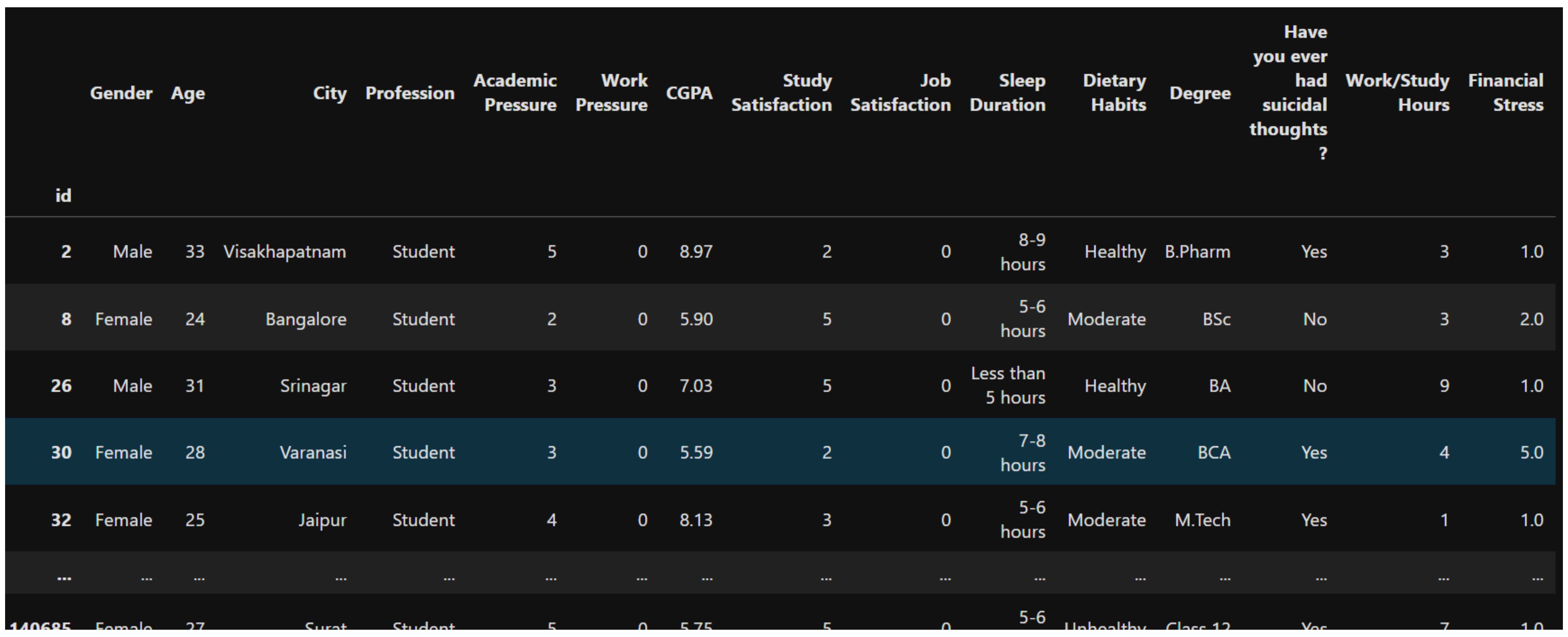
[56]:

	Degree	Sleep Duration
1	BSc	5-6 hours
3	BCA	7-8 hours
5	PhD	Less than 5 hours
7	Class 12	Less than 5 hours

36. How to Set the "id" Column as the Index?

df.set_index('id', inplace=True)

OUTPUT:

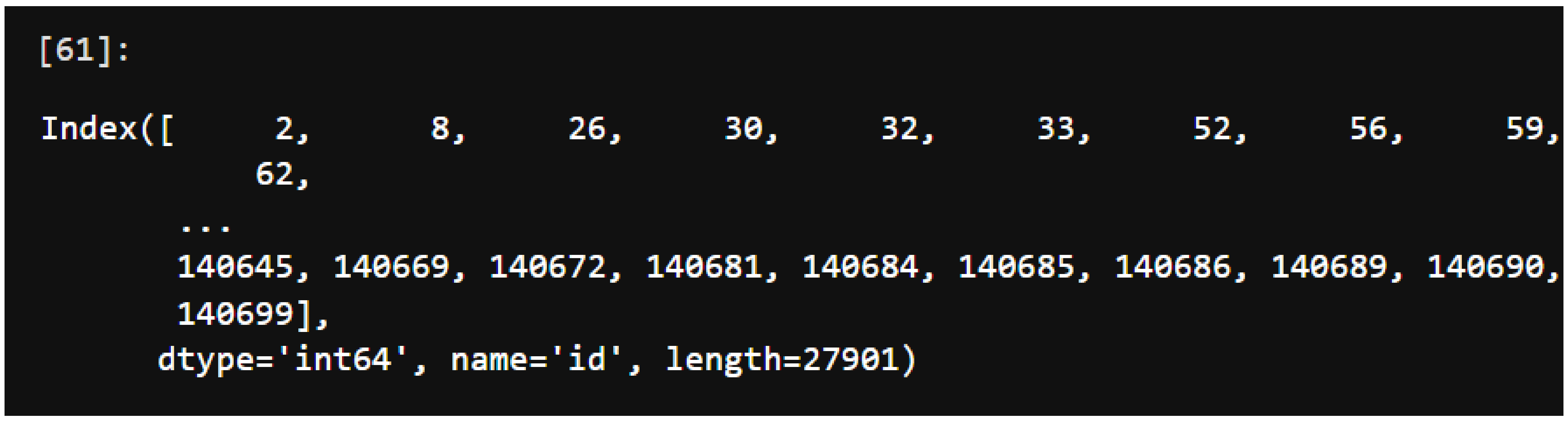


	Gender	Age	City	Profession	Academic Pressure	Work Pressure	CGPA	Study Satisfaction	Job Satisfaction	Sleep Duration	Dietary Habits	Degree	Have you ever had suicidal thoughts ?	Work/Study Hours	Financial Stress
id															
2	Male	33	Visakhapatnam	Student	5	0	8.97	2	0	8-9 hours	Healthy	B.Pharm	Yes	3	1.0
8	Female	24	Bangalore	Student	2	0	5.90	5	0	5-6 hours	Moderate	BSc	No	3	2.0
26	Male	31	Srinagar	Student	3	0	7.03	5	0	Less than 5 hours	Healthy	BA	No	9	1.0
30	Female	28	Varanasi	Student	3	0	5.59	2	0	7-8 hours	Moderate	BCA	Yes	4	5.0
32	Female	25	Jaipur	Student	4	0	8.13	3	0	5-6 hours	Moderate	M.Tech	Yes	1	1.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
140685	Female	27	Surat	Student	5	0	5.75	5	0	5-6	Unhealthy	Class 12	Yes	7	1.0

37. How to Display the Current Index of the DataFrame?

df.index

OUTPUT:

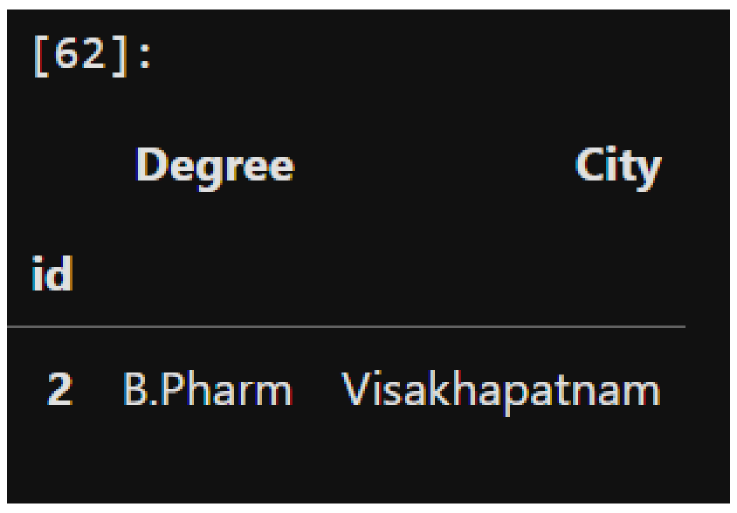


```
[61]:

Index([     2,      8,     26,     30,     32,     33,     52,     56,     59,
           62,
       ...
       140645, 140669, 140672, 140681, 140684, 140685, 140686, 140689, 140690,
       140699],
      dtype='int64', name='id', length=27901)
```

38. How to Select a Specific Row and Columns Using .loc[]?

df.loc[[2],['Degree','City']]

OUTPUT:



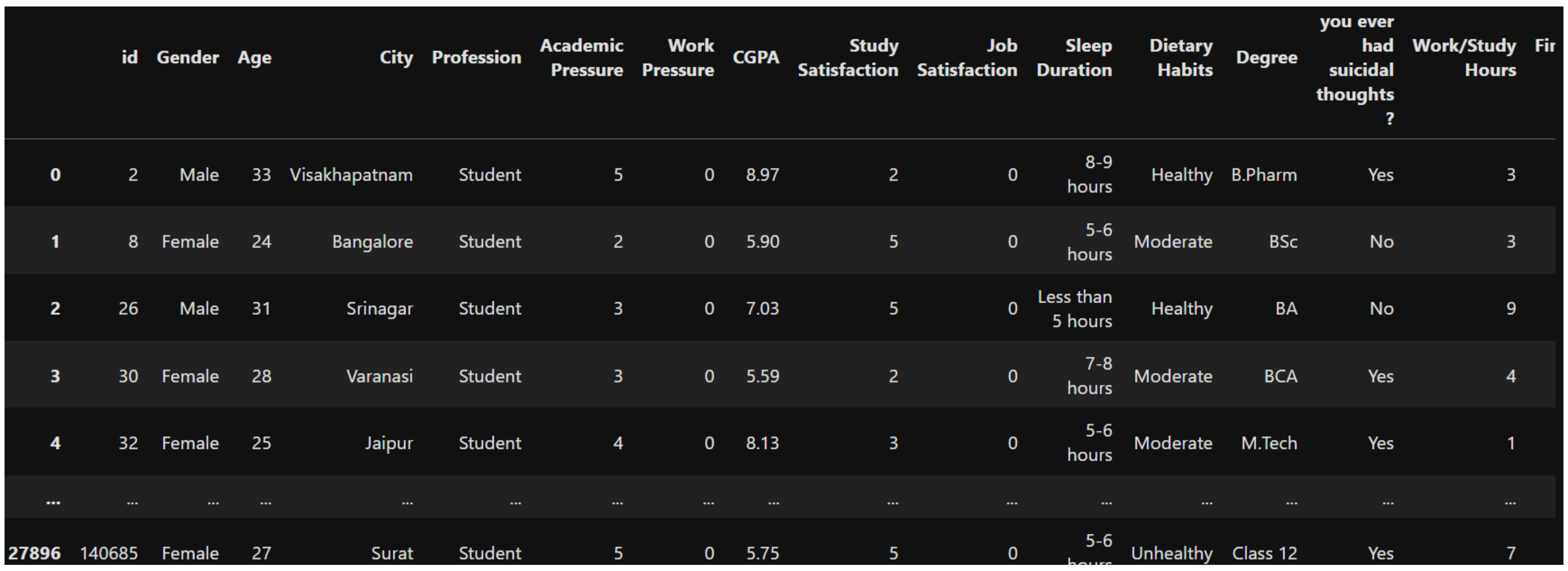
[62]:

id	Degree	City
2	B.Pharm	Visakhapatnam

39. How to Reset the Index of the DataFrame?

```
df.reset_index(inplace=True)
```

OUTPUT:

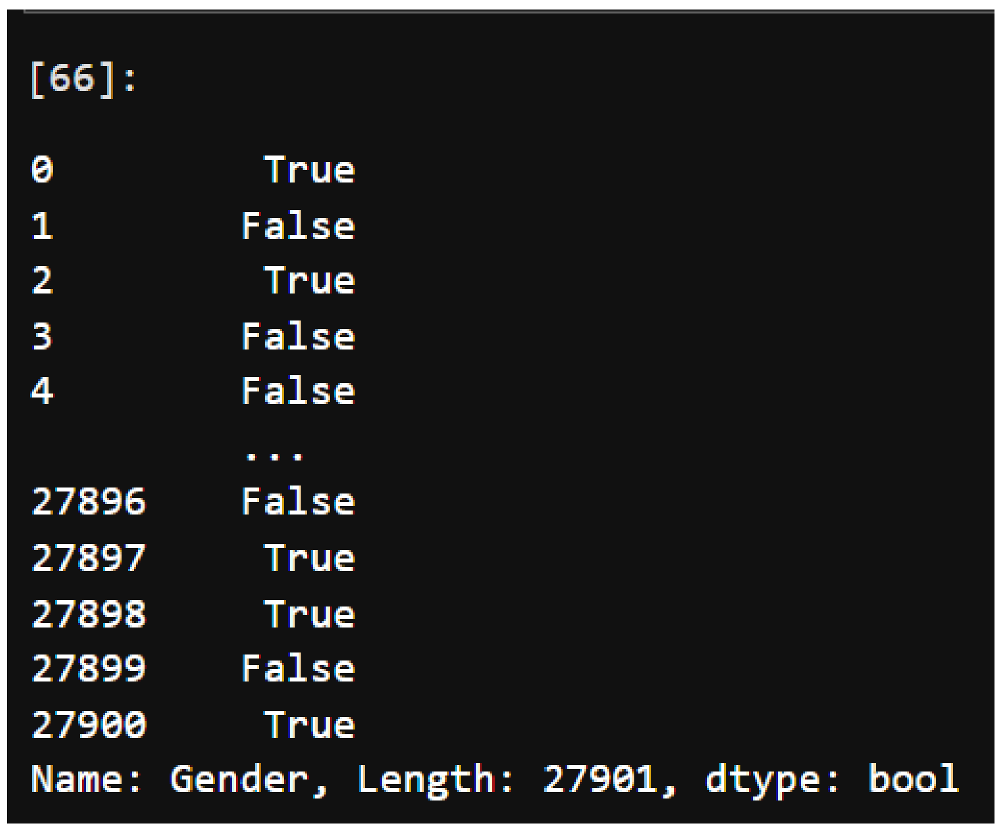


	id	Gender	Age	City	Profession	Academic Pressure	Work Pressure	CGPA	Study Satisfaction	Job Satisfaction	Sleep Duration	Dietary Habits	Degree	you ever had suicidal thoughts ?	Work/Study Hours	Fir
0	2	Male	33	Visakhapatnam	Student	5	0	8.97	2	0	8-9 hours	Healthy	B.Pharm	Yes	3	
1	8	Female	24	Bangalore	Student	2	0	5.90	5	0	5-6 hours	Moderate	BSc	No	3	
2	26	Male	31	Srinagar	Student	3	0	7.03	5	0	Less than 5 hours	Healthy	BA	No	9	
3	30	Female	28	Varanasi	Student	3	0	5.59	2	0	7-8 hours	Moderate	BCA	Yes	4	
4	32	Female	25	Jaipur	Student	4	0	8.13	3	0	5-6 hours	Moderate	M.Tech	Yes	1	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
27896	140685	Female	27	Surat	Student	5	0	5.75	5	0	5-6	Unhealthy	Class 12	Yes	7	

40. How to Filter Male Students Using a Boolean Condition?

```
df[df['Gender'] == 'Male']
```

OUTPUT:

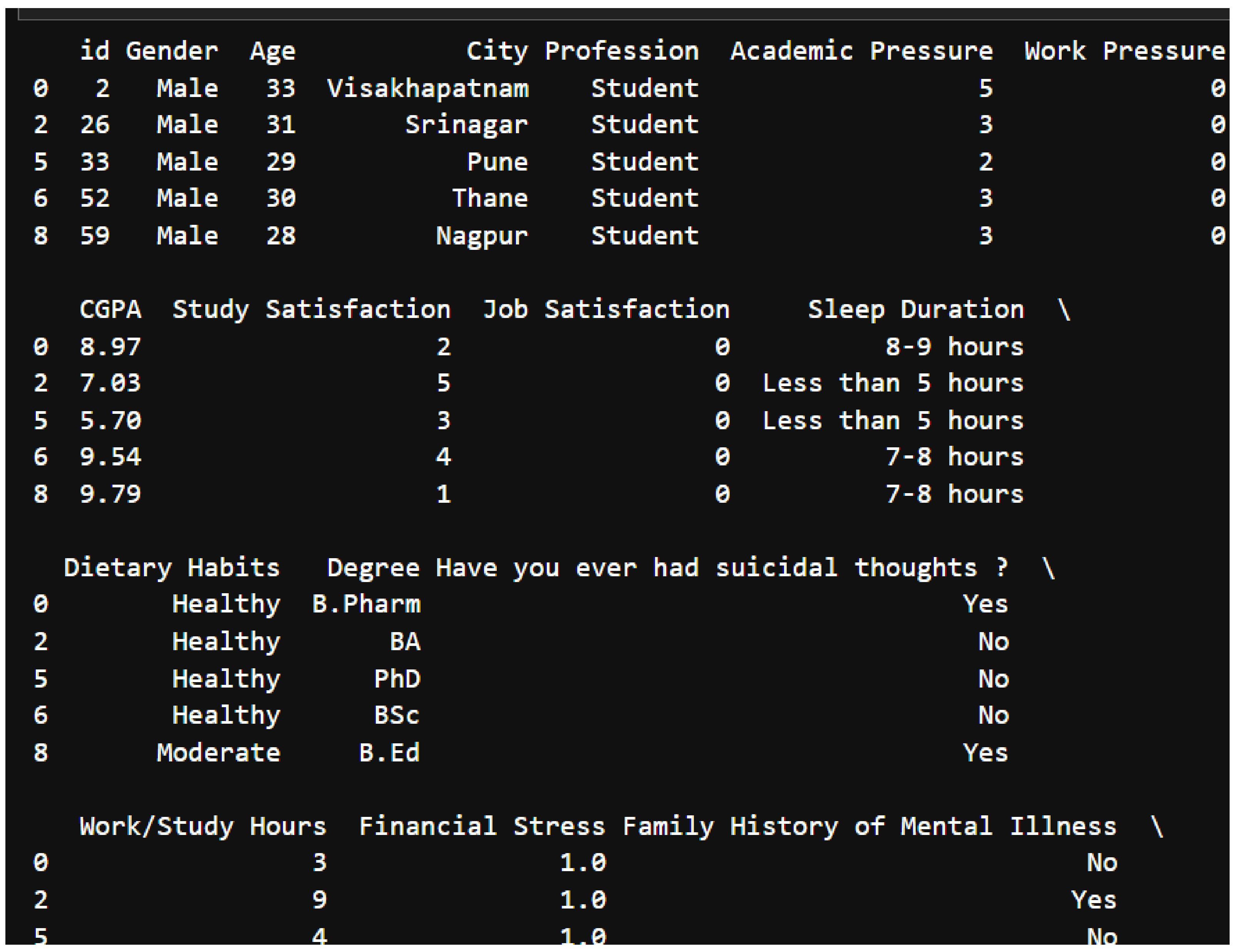


```
[66]:

0         True
1        False
2         True
3        False
4        False
          ...
27896    False
27897     True
27898     True
27899    False
27900     True
Name: Gender, Length: 27901, dtype: bool
```

41.How to Select Only Male Students Using .query()?

```
        p = df.query('Gender == "Male"')
print(p.head())
```

OUTPUT:

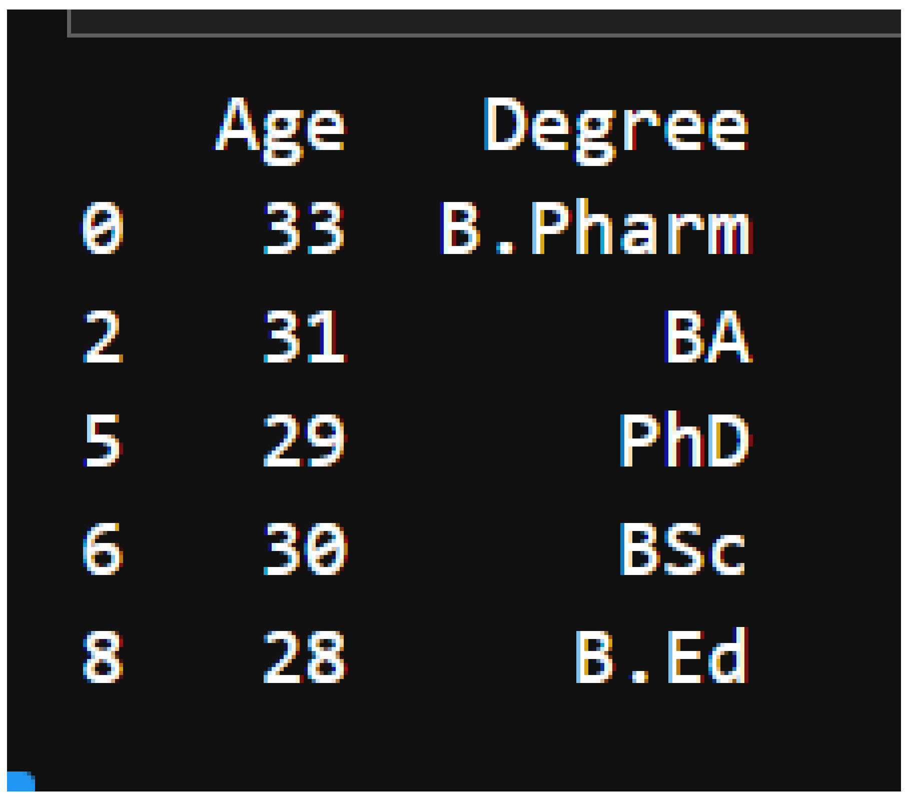


```
   id Gender  Age          City Profession  Academic Pressure  Work Pressure
0   2   Male   33  Visakhapatnam    Student                  5              0
2  26   Male   31       Srinagar    Student                  3              0
5  33   Male   29           Pune    Student                  2              0
6  52   Male   30          Thane    Student                  3              0
8  59   Male   28         Nagpur    Student                  3              0

   CGPA  Study Satisfaction  Job Satisfaction     Sleep Duration  \
0  8.97                   2                 0          8-9 hours
2  7.03                   5                 0  Less than 5 hours
5  5.70                   3                 0  Less than 5 hours
6  9.54                   4                 0          7-8 hours
8  9.79                   1                 0          7-8 hours

  Dietary Habits   Degree Have you ever had suicidal thoughts ?  \
0        Healthy  B.Pharm                                   Yes
2        Healthy       BA                                    No
5        Healthy      PhD                                    No
6        Healthy      BSc                                    No
8       Moderate     B.Ed                                   Yes

   Work/Study Hours  Financial Stress Family History of Mental Illness  \
0                 3               1.0                               No
2                 9               1.0                              Yes
5                 4               1.0                               No
```

42.How to Select Only Male Students and Display Their "Age" and "Degree"?

```
p = df.query('Gender == "Male"')[['Age','Degree']]
print(p.head())
```

OUTPUT:

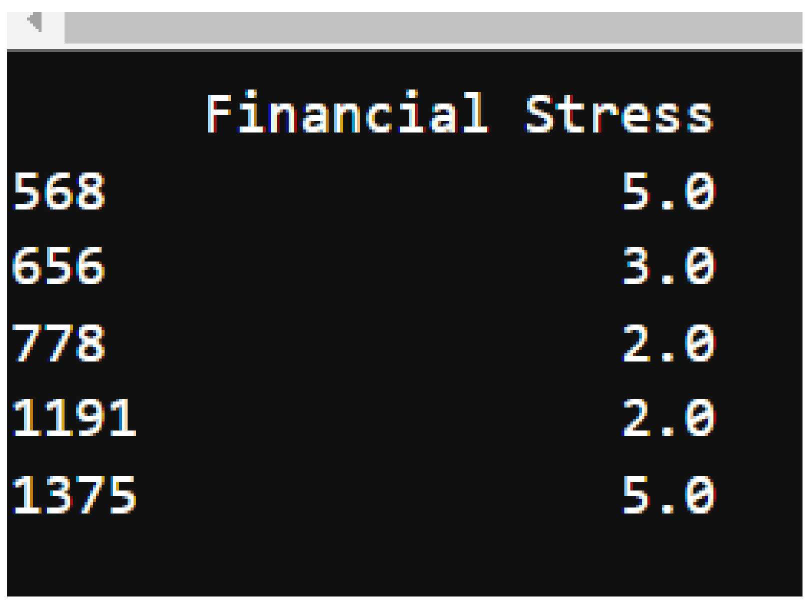


```
   Age   Degree
0   33  B.Pharm
2   31       BA
5   29      PhD
6   30      BSc
8   28     B.Ed
```

43.How to Find Students from "Kalyan" With a "BCA" Degree and Their Financial Stress?

```
a = df[(df['City'] == "Kalyan") & (df['Degree'] == "BCA")][['Financial Stress']]
print(a.head())
```

OUTPUT:

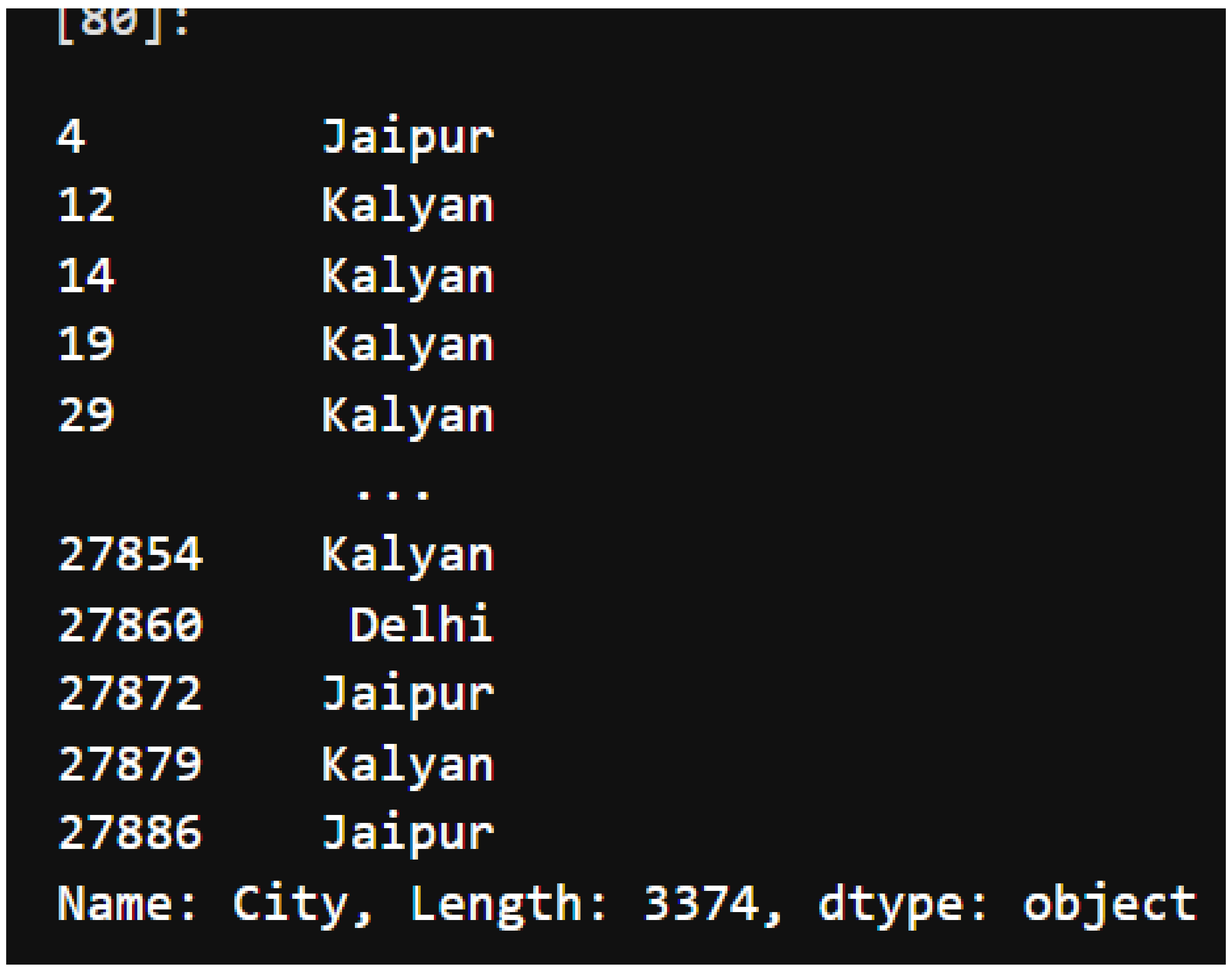


```
      Financial Stress
568                5.0
656                3.0
778                2.0
1191               2.0
1375               5.0
```

44.How to Filter Students From a List of Cities (Kalyan, Jaipur, Delhi)?

```
        ci_ty = ['Kalyan','Jaipur','Delhi']
    filter = df['City'].isin(ci_ty)
    df.loc[filter, 'City']
```

OUTPUT:

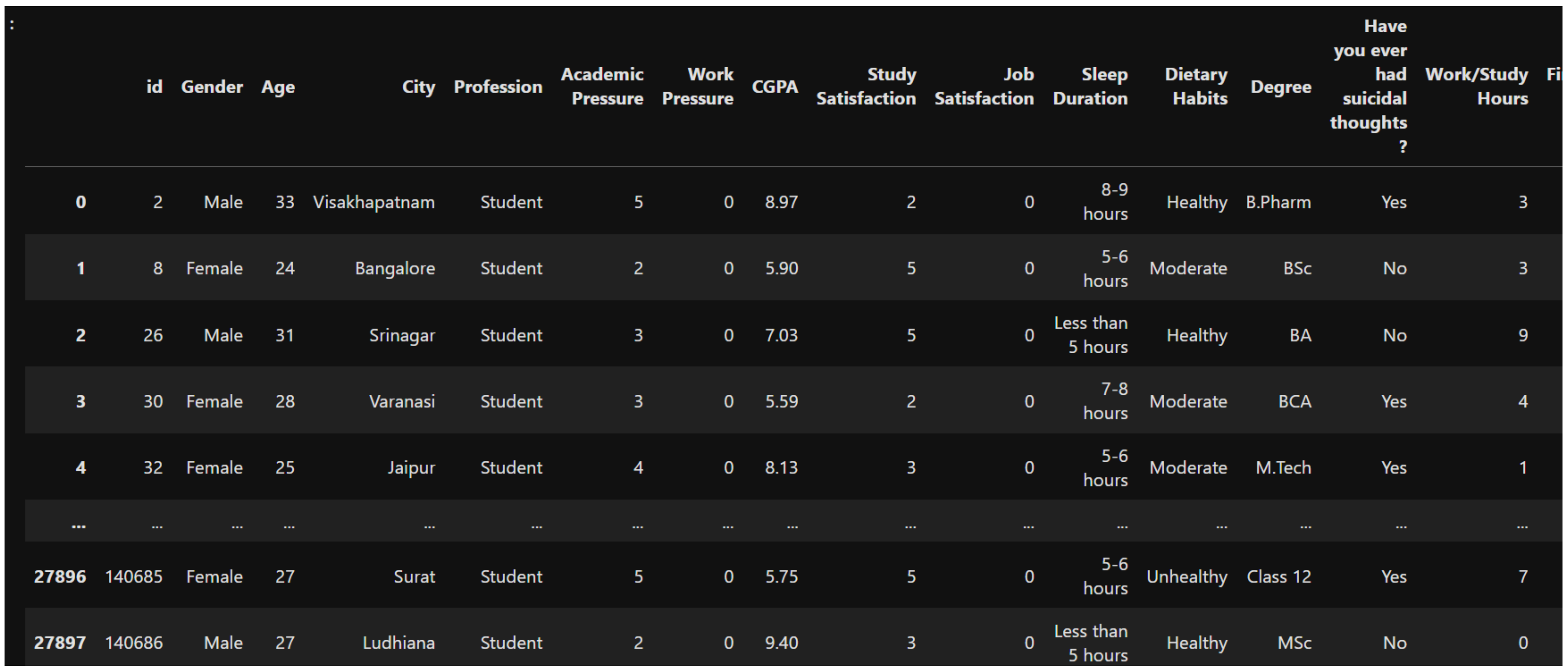


```
[80]:

4        Jaipur
12       Kalyan
14       Kalyan
19       Kalyan
29       Kalyan
          ...
27854    Kalyan
27860     Delhi
27872    Jaipur
27879    Kalyan
27886    Jaipur
Name: City, Length: 3374, dtype: object
```

45.How to Check If Any Student Has an MSc Degree?

```
        m = df['Degree'].str.contains('MSc', na=False)
```

OUTPUT:

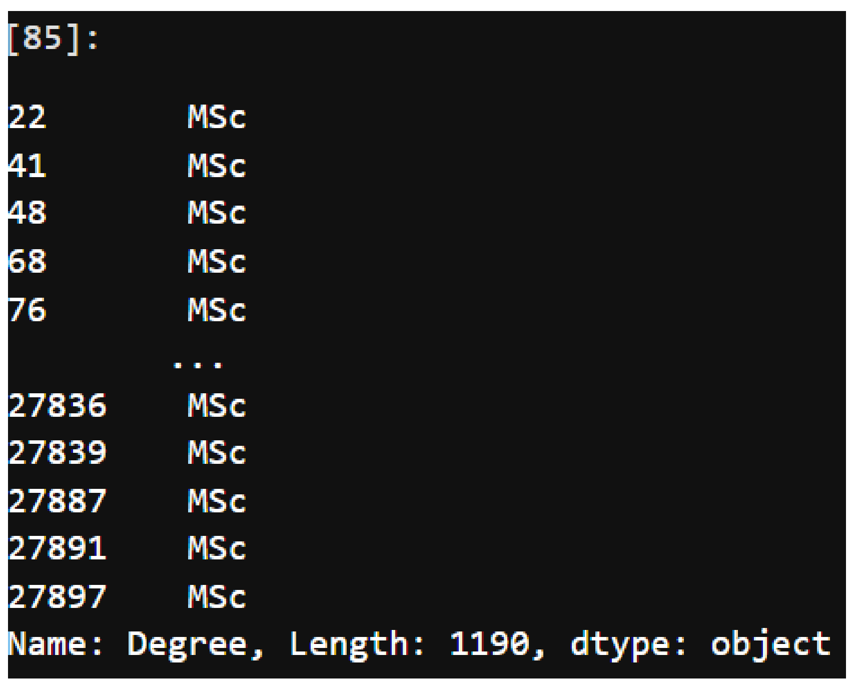


:

	id	Gender	Age	City	Profession	Academic Pressure	Work Pressure	CGPA	Study Satisfaction	Job Satisfaction	Sleep Duration	Dietary Habits	Degree	Have you ever had suicidal thoughts ?	Work/Study Hours	Fi
0	2	Male	33	Visakhapatnam	Student	5	0	8.97	2	0	8-9 hours	Healthy	B.Pharm	Yes	3	
1	8	Female	24	Bangalore	Student	2	0	5.90	5	0	5-6 hours	Moderate	BSc	No	3	
2	26	Male	31	Srinagar	Student	3	0	7.03	5	0	Less than 5 hours	Healthy	BA	No	9	
3	30	Female	28	Varanasi	Student	3	0	5.59	2	0	7-8 hours	Moderate	BCA	Yes	4	
4	32	Female	25	Jaipur	Student	4	0	8.13	3	0	5-6 hours	Moderate	M.Tech	Yes	1	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
27896	140685	Female	27	Surat	Student	5	0	5.75	5	0	5-6 hours	Unhealthy	Class 12	Yes	7	
27897	140686	Male	27	Ludhiana	Student	2	0	9.40	3	0	Less than 5 hours	Healthy	MSc	No	0	

```
    df.loc[m, 'Degree']
```

OUTPUT:



```
[85]:

22       MSc
41       MSc
48       MSc
68       MSc
76       MSc
        ...
27836    MSc
27839    MSc
27887    MSc
27891    MSc
27897    MSc
Name: Degree, Length: 1190, dtype: object
```

Questions on DataFrame Creation:

1. What are the different methods used in the code to create a Pandas DataFrame?

There are four methods used to create a Pandas DataFrame:

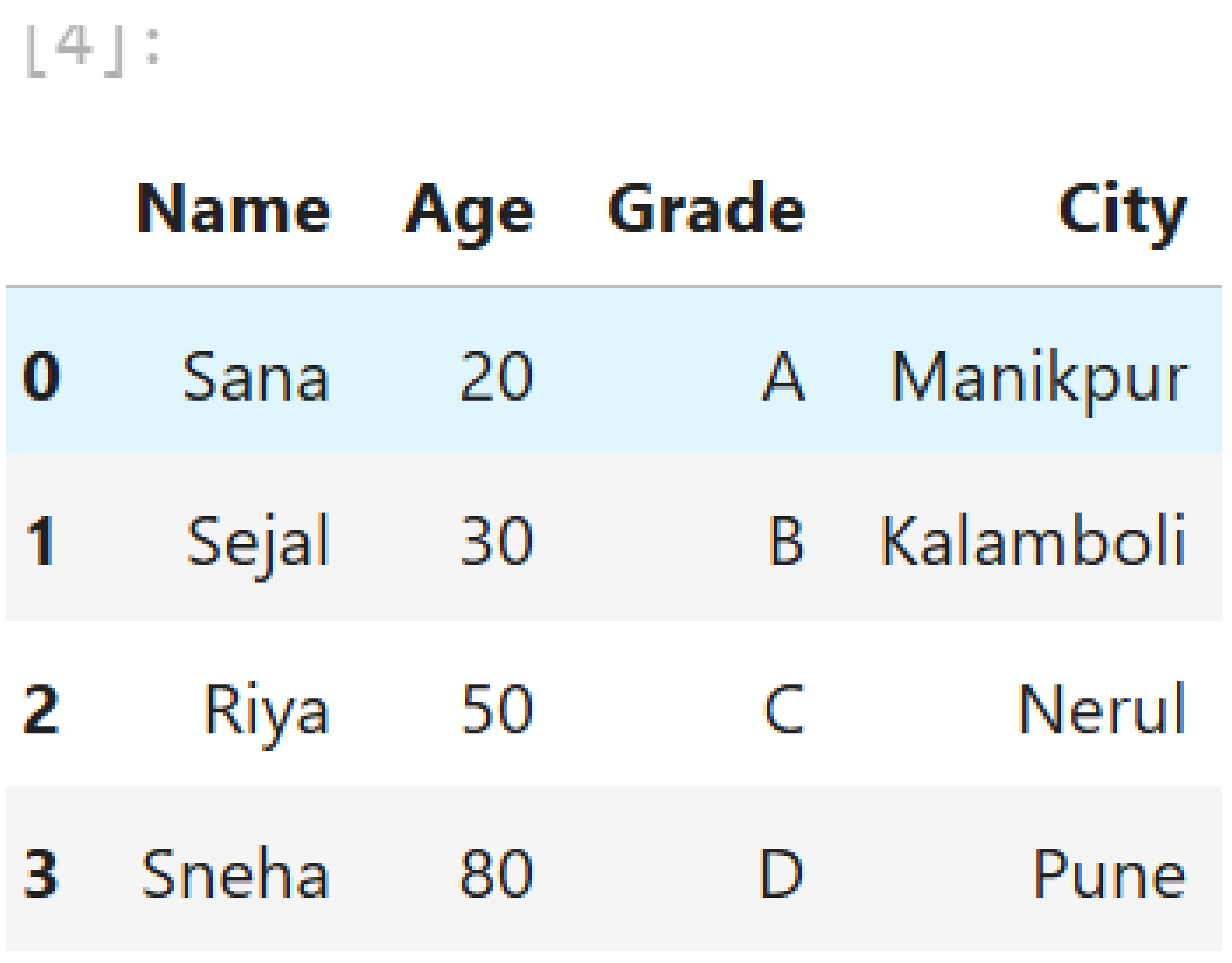
- Using a dictionary
- Using the zip() function with lists
- Using NumPy arrays
- Using a list of dictionaries

- **Code for creating a Pandas DataFrame using a dictionary:**

```
import pandas as pd

# Creating a dictionary with student data
data = {
    "Name": ["Sana", "Sejal", "Riya", "Sneha"],
    "Age": [20, 30, 50, 80],
    "Grade": ["A", "B", "C", "D"],
    "City": ["Manikpur", "Kalamboli", "Nerul", "Pune"]
}

# Creating DataFrame from the dictionary
df = pd.DataFrame(data)

# Display the DataFrame
```

```
print(df)
```

OUTPUT:



[4]:

	Name	Age	Grade	City
0	Sana	20	A	Manikpur
1	Sejal	30	B	Kalamboli
2	Riya	50	C	Nerul
3	Sneha	80	D	Pune

- Using the zip() function with lists

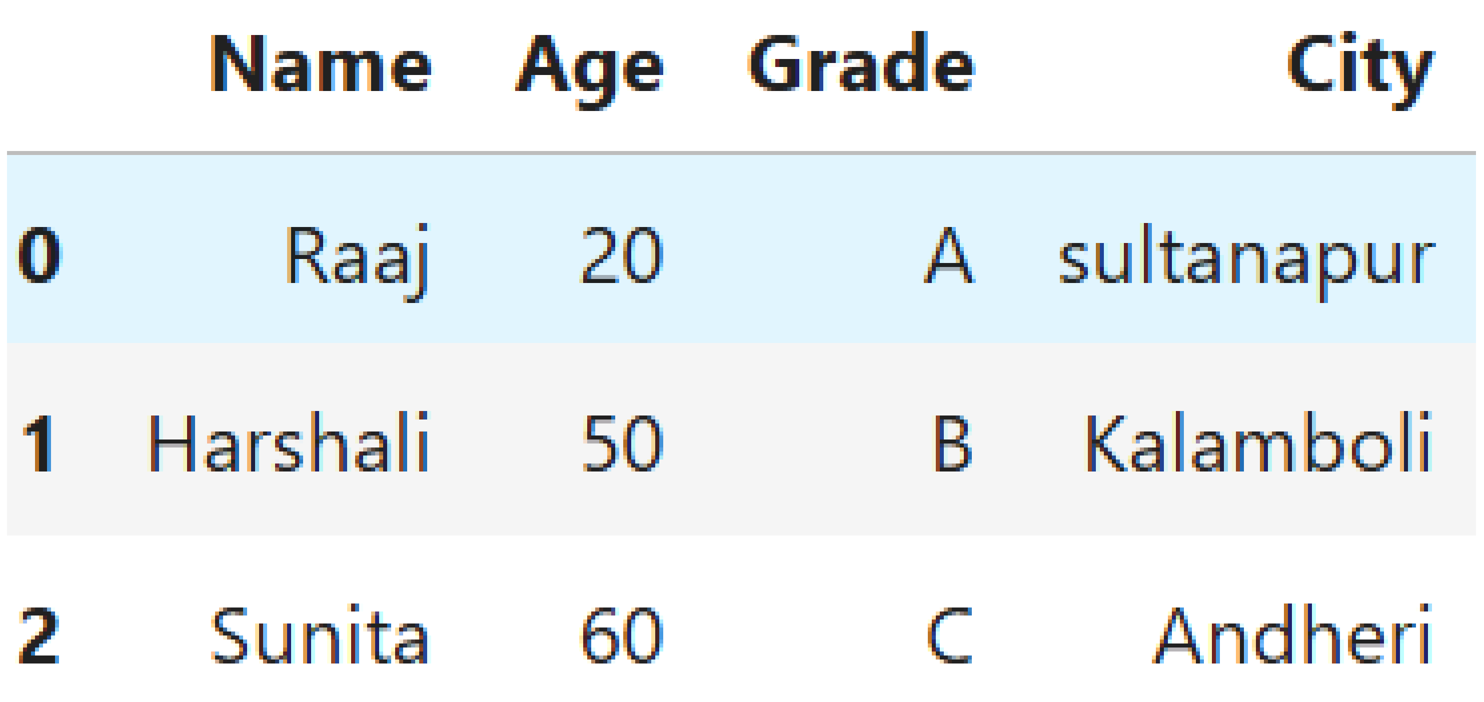
  The zip() function pairs corresponding elements from multiple lists and allows us to create a DataFrame by converting them into tuples.

MANUAL APPROACH

```
# Data for each column
names=["Raaj","Harshali","Sunita"]
ages=[20,50,60]
grades=["A","B","C"]
cities=["sultanapur","Kalamboli","Andheri"]
```

```
df=pd.DataFrame(list(zip(names,ages,grades,cities)),columns
=["Name","Age","Grade","City"])
df
```

OUTPUT:



	Name	Age	Grade	City
0	Raaj	20	A	sultanapur
1	Harshali	50	B	Kalamboli
2	Sunita	60	C	Andheri

- Using NumPy arrays

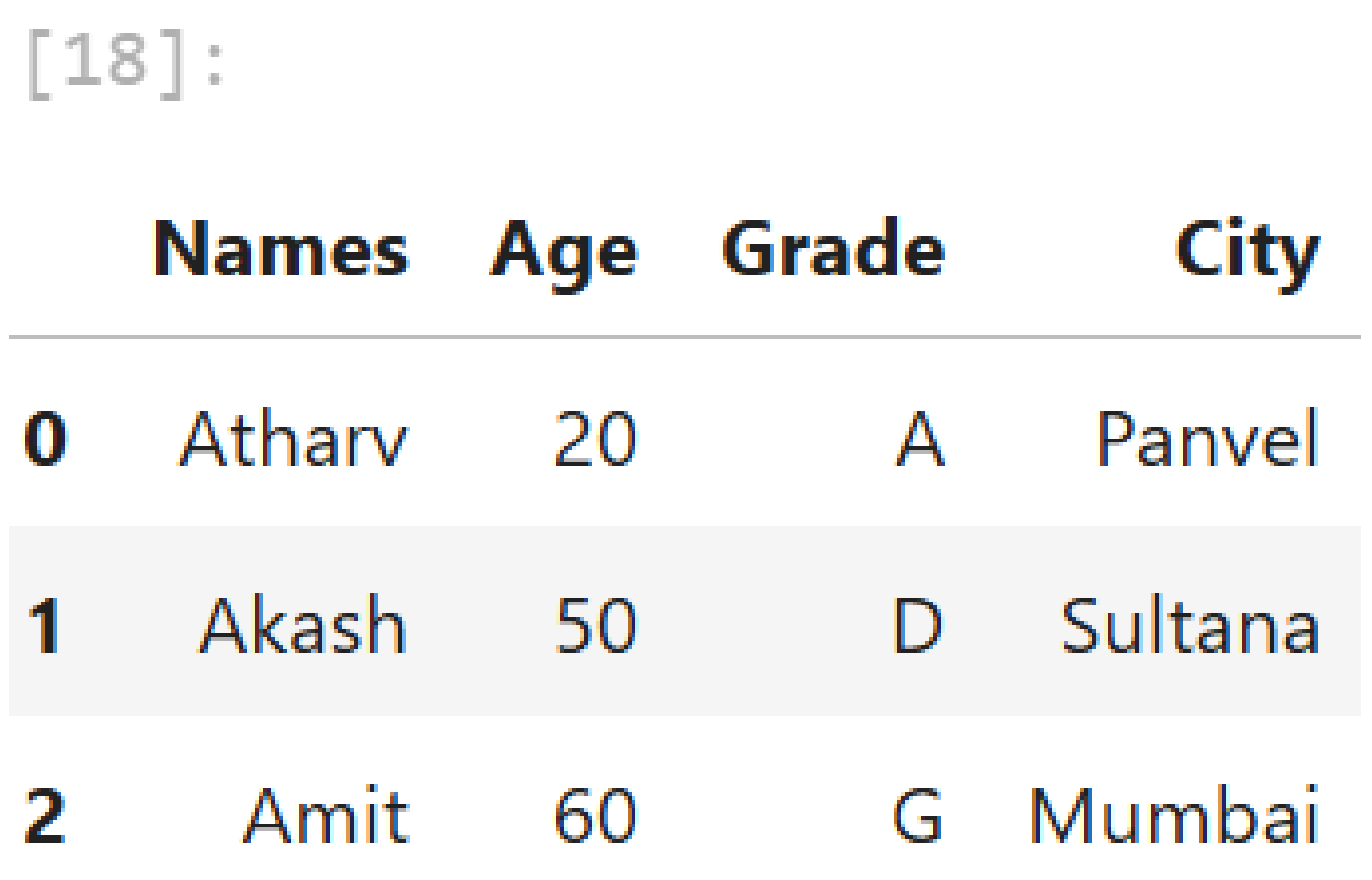
numpy.array() is used to store column data efficiently and perform mathematical operations more effectively.

# USING NUMPY ARRAY METHOD

```
            import pandas as pd
            import numpy as np

            # Using numpy arrays
            names=np.array(["Atharv","Akash","Amit"])
            ages=np.array([20,50,60])
            grades=np.array(["A","D","G"])
            cities=np.array(["Panvel","Sultana","Mumbai"])
```

```
df=pd.DataFrame({
    "Names":names,
    "Age":ages,
    "Grade":grades,
    "City":cities
})
df
```

OUTPUT:



[18]:

	Names	Age	Grade	City
0	Atharv	20	A	Panvel
1	Akash	50	D	Sultana
2	Amit	60	G	Mumbai

- Using a list of dictionaries

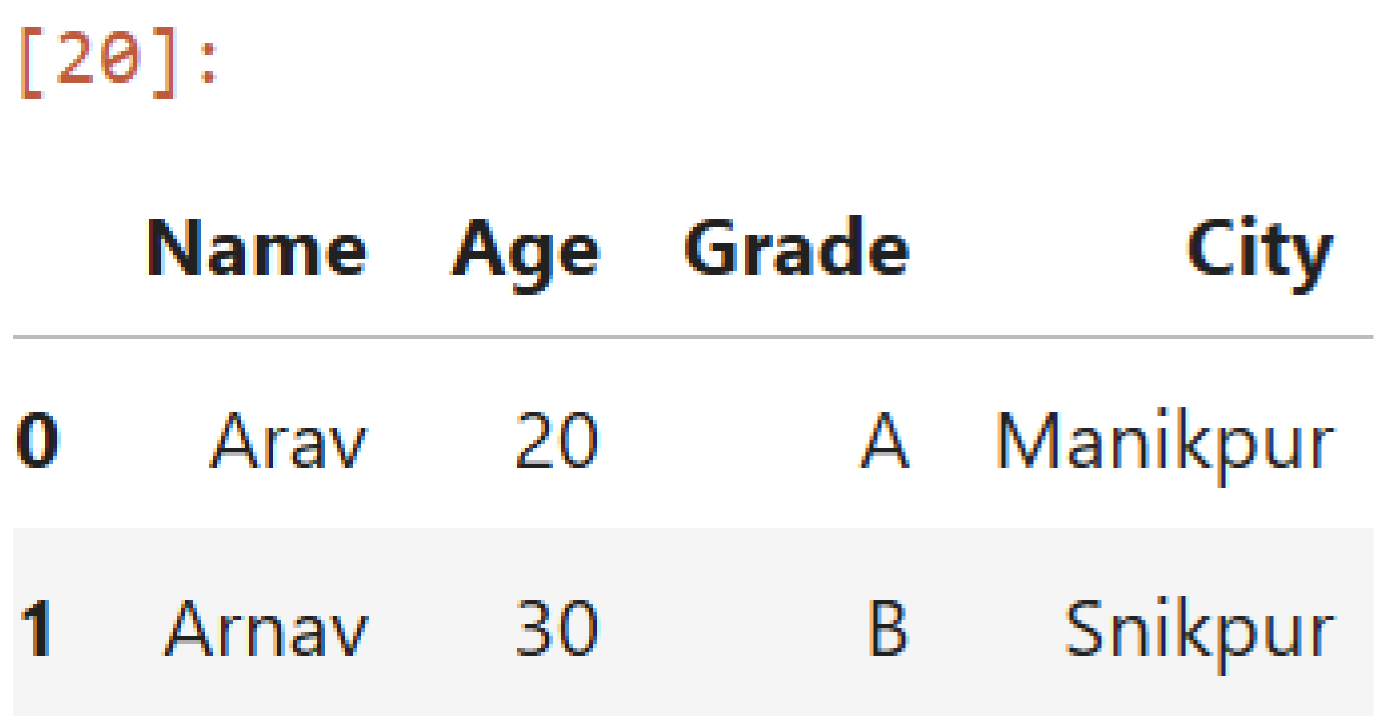
A list of dictionaries represents each row as a dictionary, making it easy to create records dynamically. Unlike the dictionary method, where keys are column names, each dictionary in the list represents a row.

# USING LIST OF DICTIONARIES

```
import pandas as pd
```

```
# List of dictionaries representing each student record
data=[
   {"Name":"Arav","Age":20,"Grade":"A","City":"Manikpur"},
   {"Name":"Arnav","Age":30,"Grade":"B","City":"Snikpur"}
]

# Creating DataFrame
df = pd.DataFrame(data)
df
```

OUTPUT:



[20]:

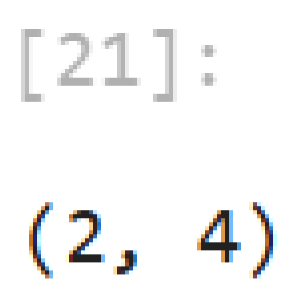
	Name	Age	Grade	City
0	Arav	20	A	Manikpur
1	Arnav	30	B	Snikpur

2. What does df.shape return, and how can we use it to find the number of rows and columns?

df.shape returns a tuple (rows, columns), where the first value is the number of rows and the second is the number of columns.

```
df.shape
```

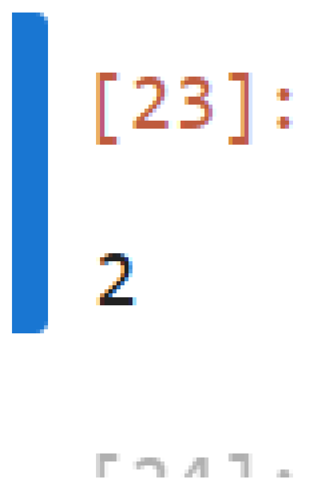
OUTPUT:



```
[21]:

(2, 4)
```

```
rows, columns = df.shape
rows
```

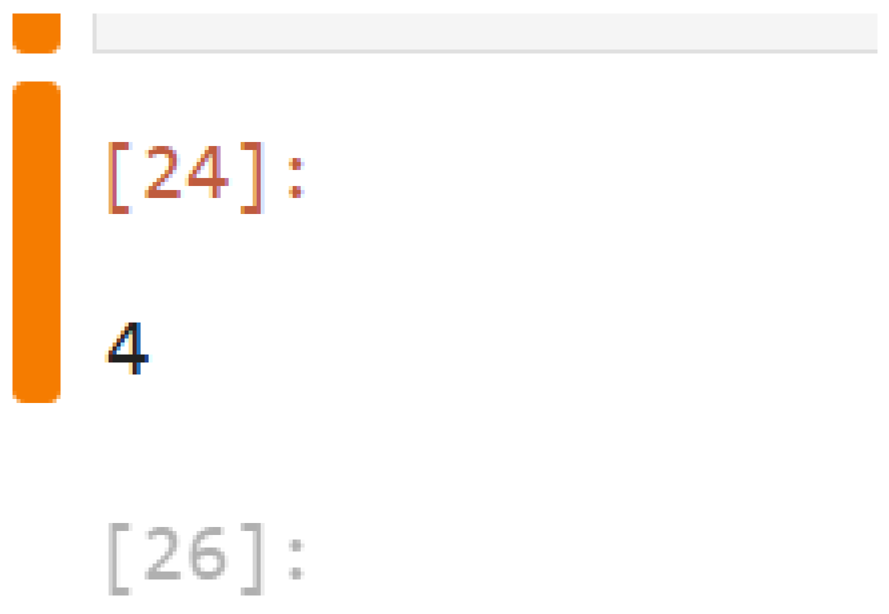
OUTPUT:



```
[23]:
2
```

```
Columns
```

OUTPUT:



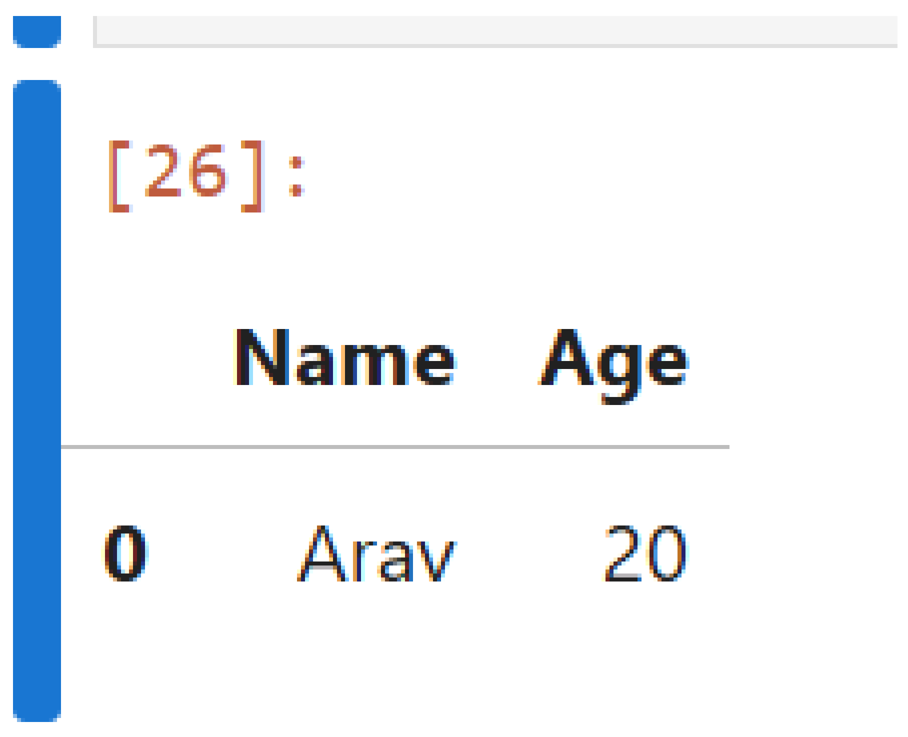
```
[24]:
4

[26]:
```

3. How can we filter out only those students who have received an "A" grade?

We can filter the DataFrame using a condition on the "Grade" column

```
df[['Name', 'Age']][df['Grade'] == 'A']
```

OUTPUT:



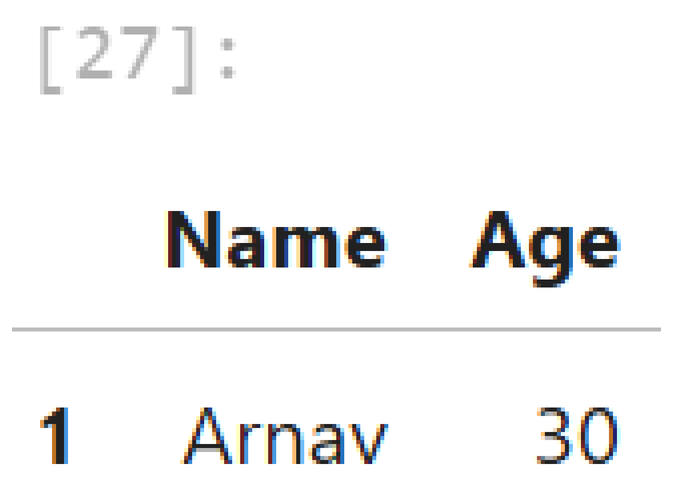
```
[26]:
     Name  Age
0    Arav   20
```

4. How can we find students whose age is the maximum in the DataFrame?

We can use the max() function to get the maximum age and filter rows accordingly.

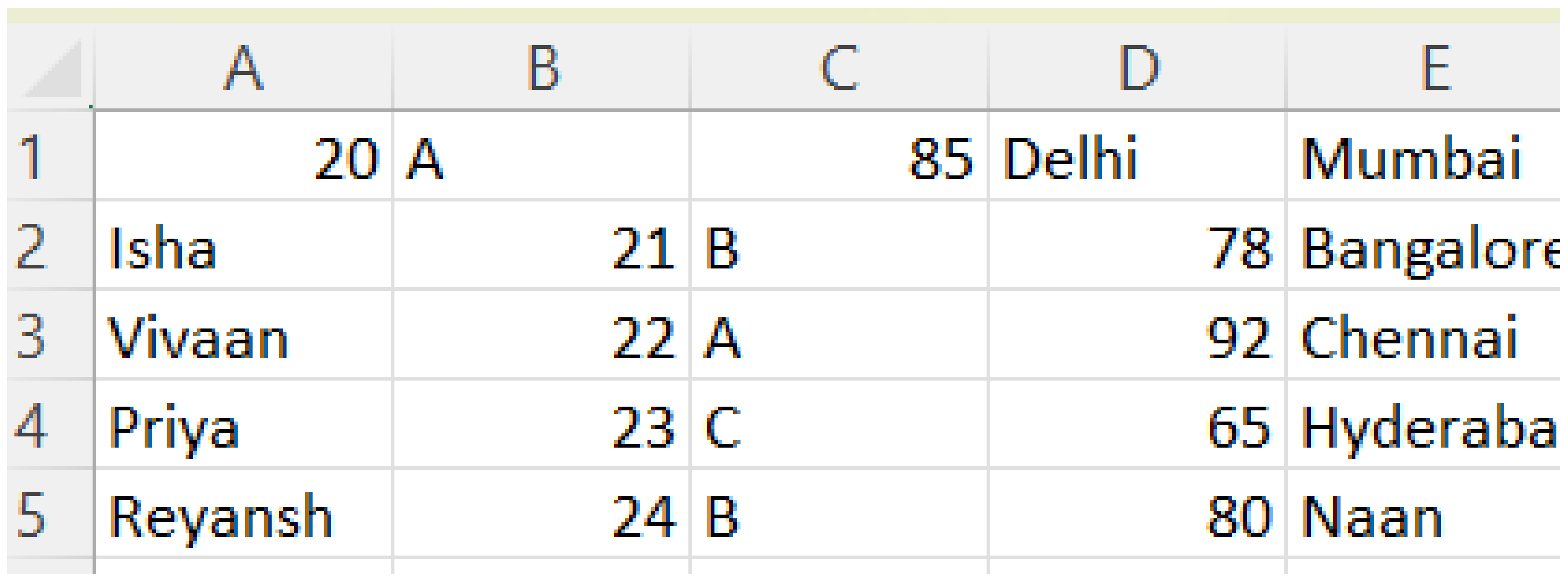
```
#SET INDEX
        df[['Name' ,'Age']] [df['Age']==df['Age'].max()]
```

OUTPUT:



```
[27]:
     Name   Age
1    Arnav   30
```

Importing Data from Excel-Sheet:



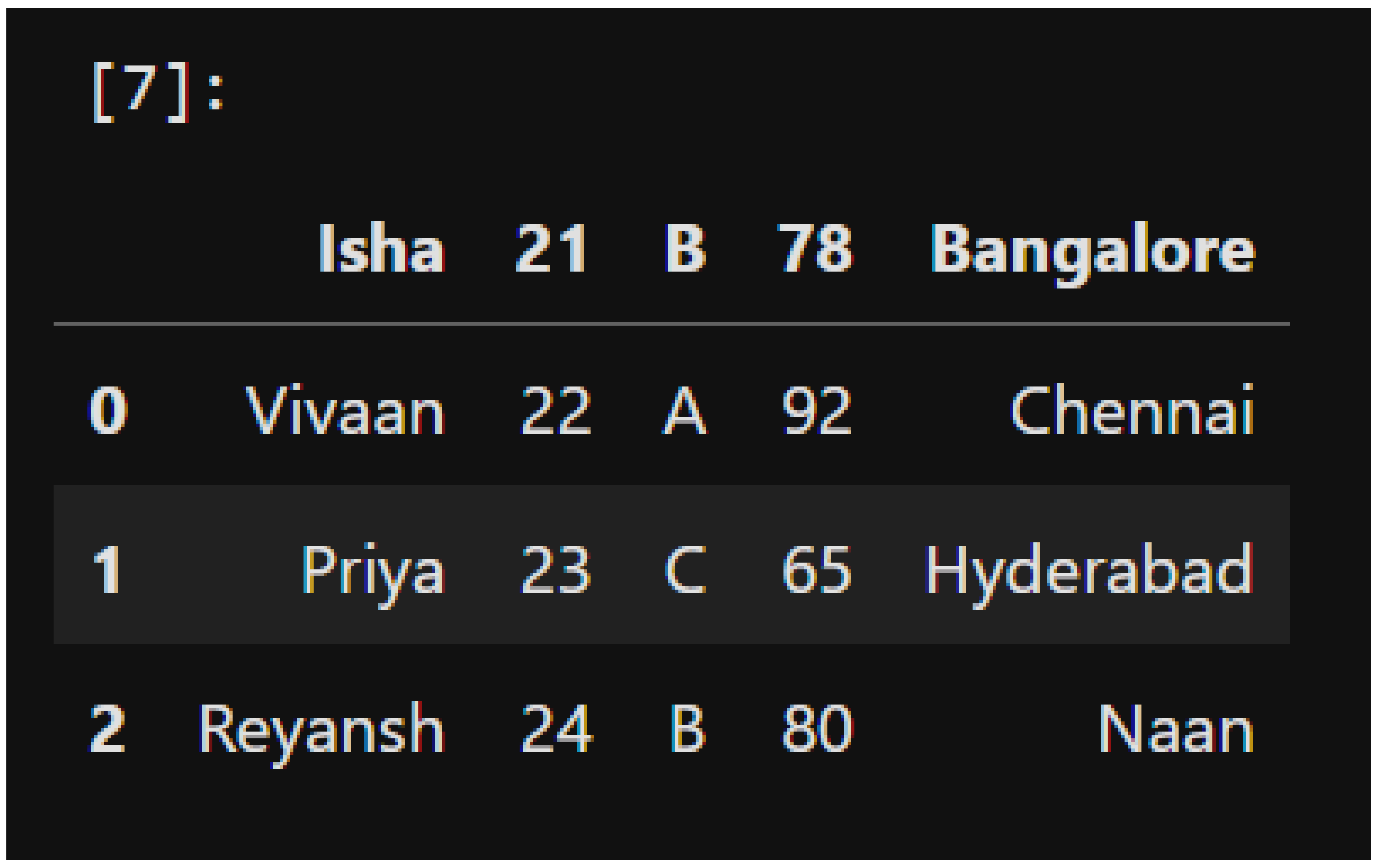
	A	B	C	D	E
1	20	A	85	Delhi	Mumbai
2	Isha	21	B	78	Bangalore
3	Vivaan	22	A	92	Chennai
4	Priya	23	C	65	Hyderaba
5	Reyansh	24	B	80	Naan

Q: What does the skiprows=1 argument do in pd.read_csv()?

A: It skips the first row of the CSV file while reading the data.

```
        import pandas as pd
        df=pd.read_csv("C:\\Users\\Sana\\OneDrive\\Desktop\\
    Book1.csv",skiprows=1)
        df
```

OUTPUT:



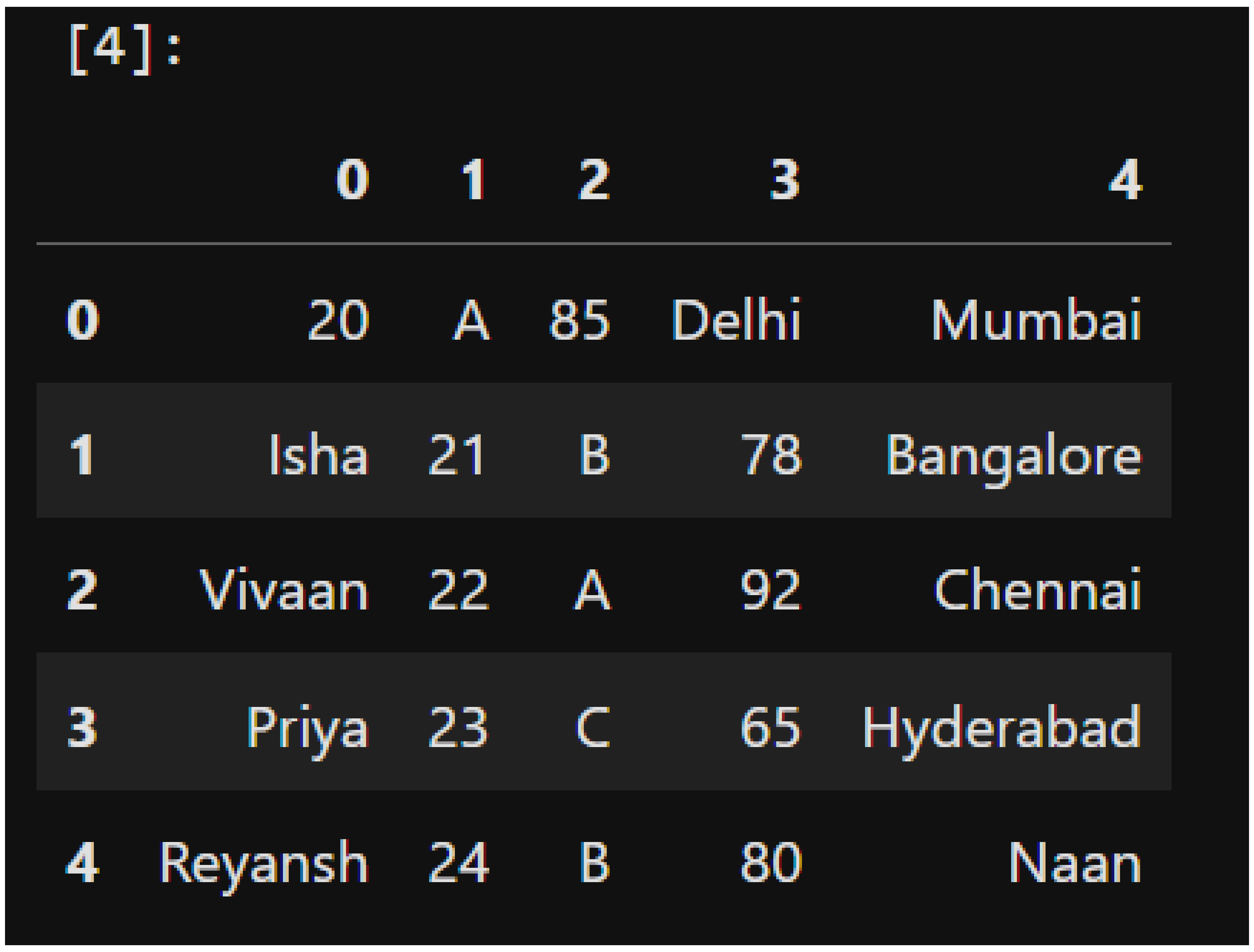
```
[7]:
         Isha  21  B  78  Bangalore
0      Vivaan  22  A  92    Chennai
1       Priya  23  C  65  Hyderabad
2     Reyansh  24  B  80       Naan
```

Q: What is the effect of using header=None when reading a CSV file?

A: It treats the first row as data instead of column headers.

```
df=pd.read_csv("C:\\Users\\Sana\\OneDrive\\Desktop\\Book1.csv",header=None)
df
```

OUTPUT:



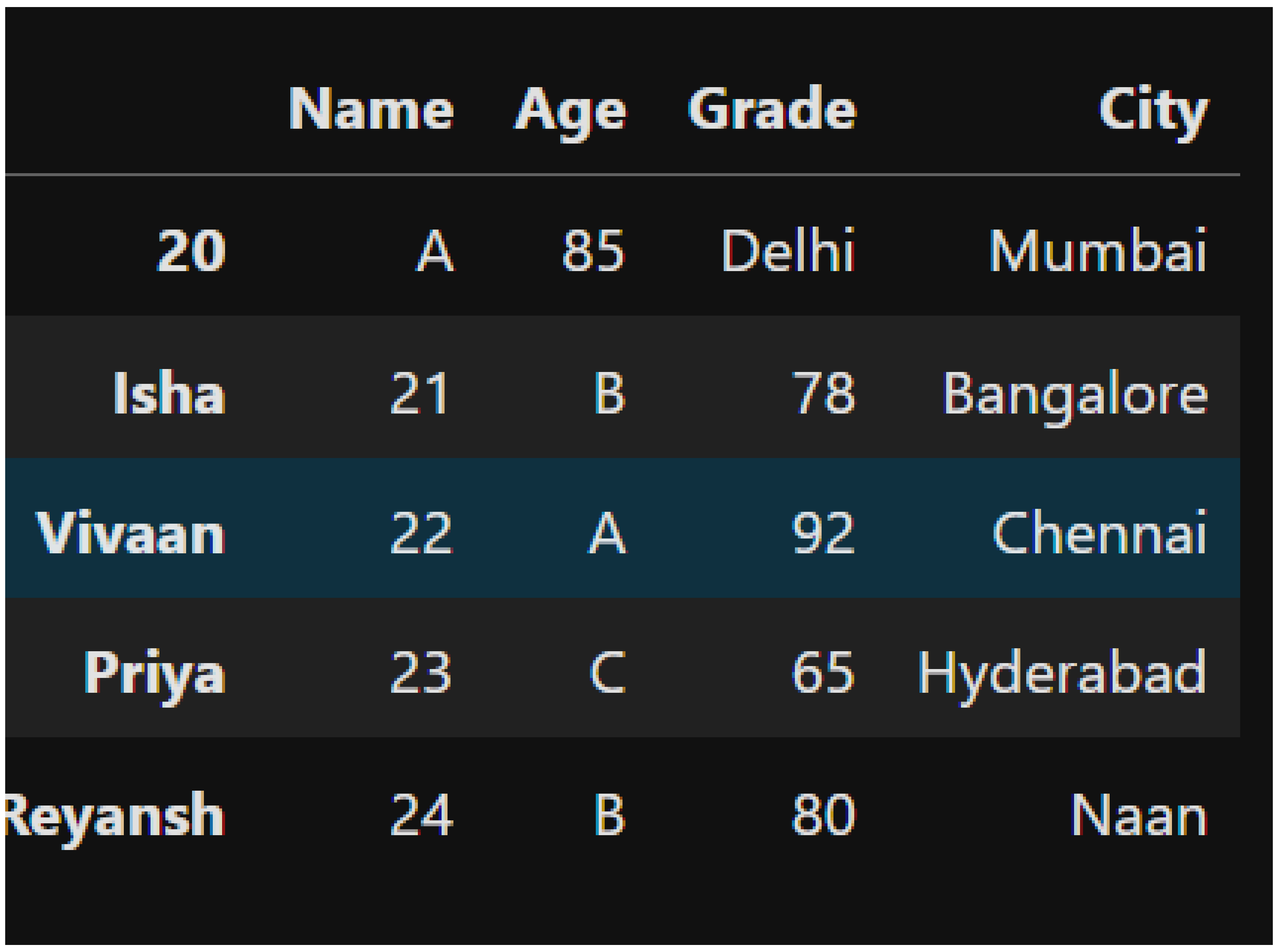
```
[4]:
          0    1   2      3          4
0        20    A  85  Delhi     Mumbai
1      Isha   21   B     78  Bangalore
2    Vivaan   22   A     92    Chennai
3     Priya   23   C     65  Hyderabad
4   Reyansh   24   B     80       Naan
```

Q: How does specifying names=["Name","Age","Grade","City"] affect the DataFrame?

A: It assigns custom column names instead of using the default ones from the file.

```
df=pd.read_csv("C:\\Users\\Sana\\OneDrive\\Desktop\\Book1.csv",names=["Name","Age","Grade","City"])
```

OUTPUT:



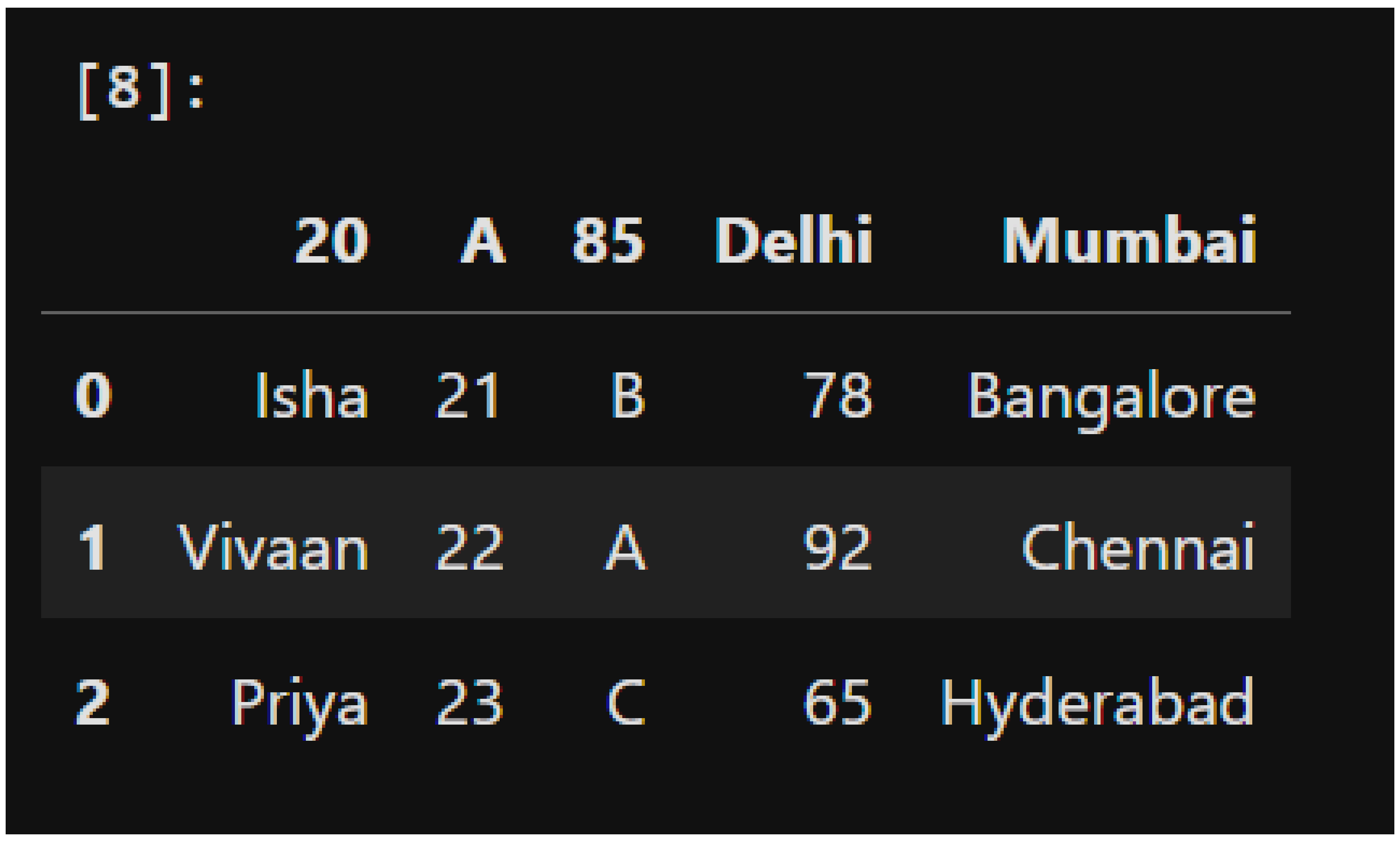
	Name	Age	Grade	City
20	A	85	Delhi	Mumbai
Isha	21	B	78	Bangalore
Vivaan	22	A	92	Chennai
Priya	23	C	65	Hyderabad
Reyansh	24	B	80	Naan

Q: What happens when nrows=3 is used while reading the CSV file?

A: It only reads the first three rows of the dataset.

```
df=pd.read_csv("C:\\Users\\Sana\\OneDrive\\Desktop\\Book1.
csv",nrows=3)
```

OUTPUT:



```
[8]:
```

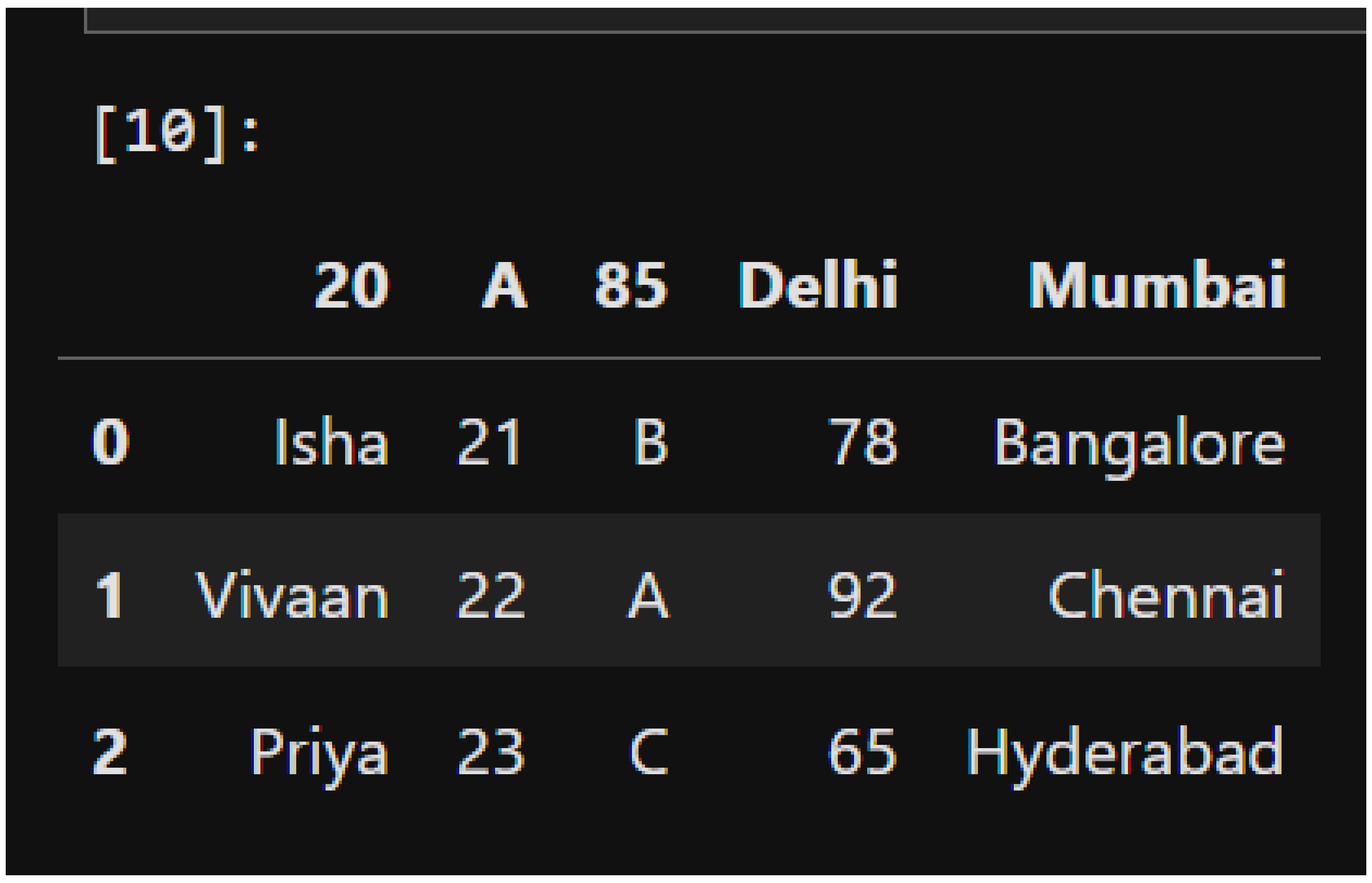
	20	A	85	Delhi	Mumbai
0	Isha	21	B	78	Bangalore
1	Vivaan	22	A	92	Chennai
2	Priya	23	C	65	Hyderabad

Q: What does df.to_csv("Book2.csv", index=False) do?

A: It saves the DataFrame to a CSV file named "Book2.csv", excluding the index column.

```
df.to_csv("Book2.csv",index=False)
```

OUTPUT:



```
[10]:
       20    A  85  Delhi     Mumbai
0    Isha   21   B     78  Bangalore
1  Vivaan   22   A     92    Chennai
2   Priya   23   C     65  Hyderabad
```

Data Conversion

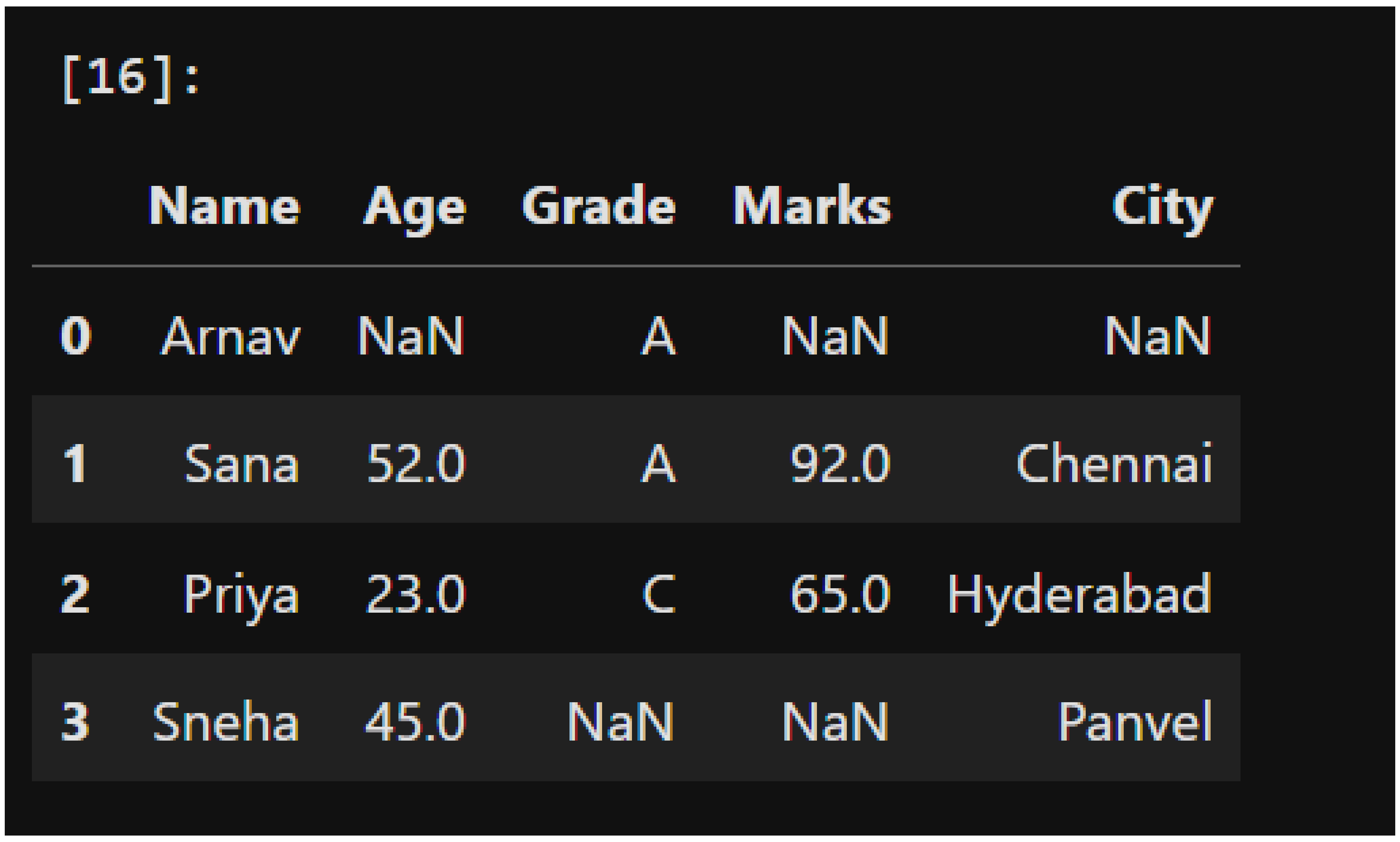
Q: What is the purpose of the convert_City_cell function?

A: It replaces any cell with the string 'NaN' in the "City" column with 'Hello'.

```
import pandas as pd
import numpy as np
df=pd.read_csv("C:\\Users\\Sana\\pandas\\Book2.csv",skiprows=0)
```

df

OUTPUT:



```
[16]:
```

	Name	Age	Grade	Marks	City
0	Arnav	NaN	A	NaN	NaN
1	Sana	52.0	A	92.0	Chennai
2	Priya	23.0	C	65.0	Hyderabad
3	Sneha	45.0	NaN	NaN	Panvel

Q: What is the purpose of the convert_City_cell function?

A: It replaces any cell with the string 'NaN' in the "City" column with 'Hello'.

```
     # Define the converter function
     def convert_City_cell(cell):
  if cell=='NaN':  # Check if the cell is NaN
    return 'Hello'
  return cell
```

Q: How does the convert_Grade_cell function modify the data?

A: It replaces any cell with the string 'NaN' in the "Grade" column with 'zero'.

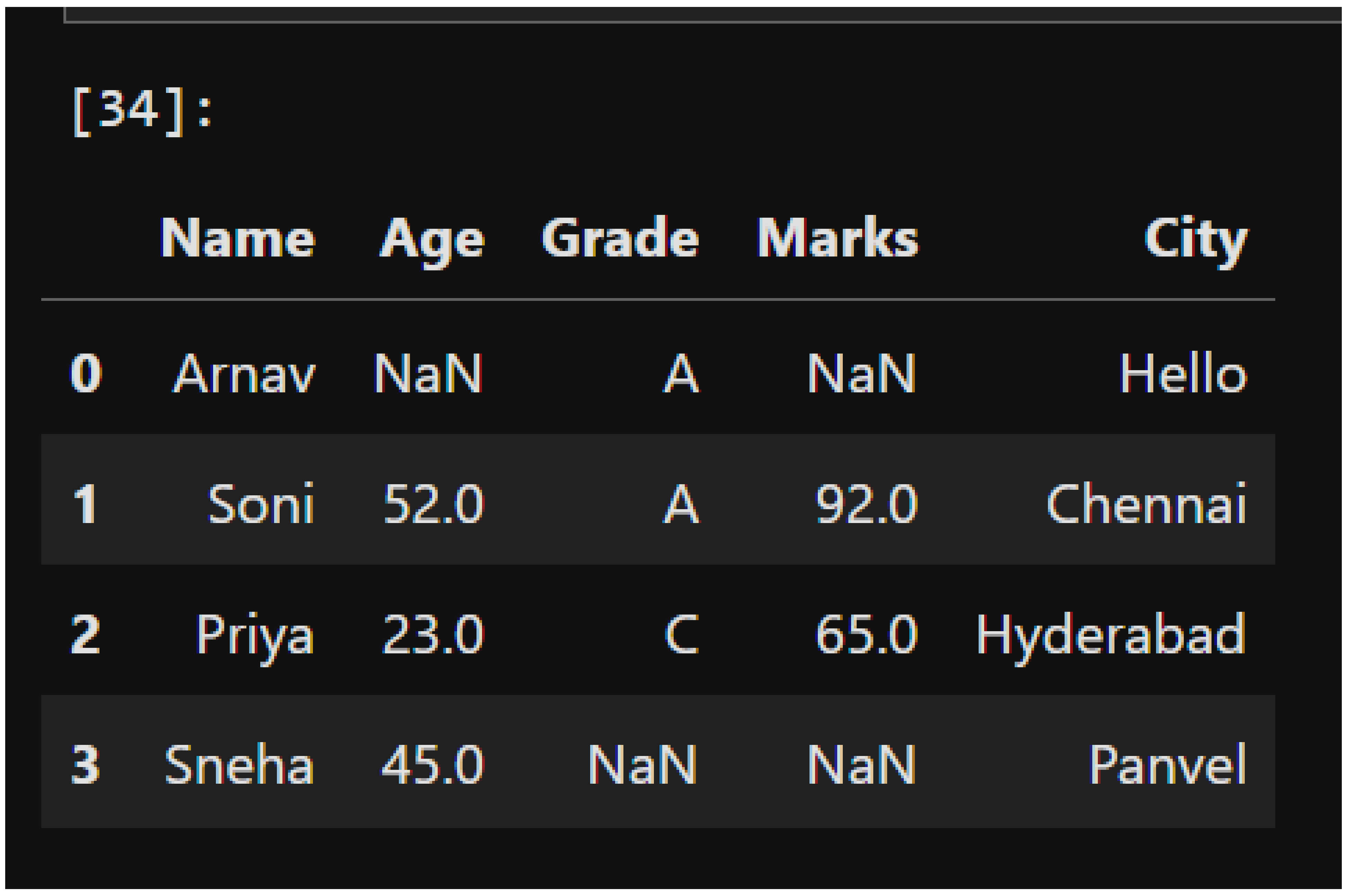
```
        def convert_Grade_cell(cell):
            if cell=='NaN':  # Check if the cell is NaN
      return 'zero'
  return cell
```

Q: What will happen if a cell contains 'NaN' in the 'City' or 'Grade' column?

A: It will be replaced with 'Hello' in the "City" column and 'zero' in the "Grade" column.

```
# Read the CSV file using the converter for the 'City' column
df = pd.read_csv('C:\\Users\\Sana\\pandas\\Book2.csv', converters={'City': convert_City_cell,'Grade': convert_Grade_cell})
```

OUTPUT:



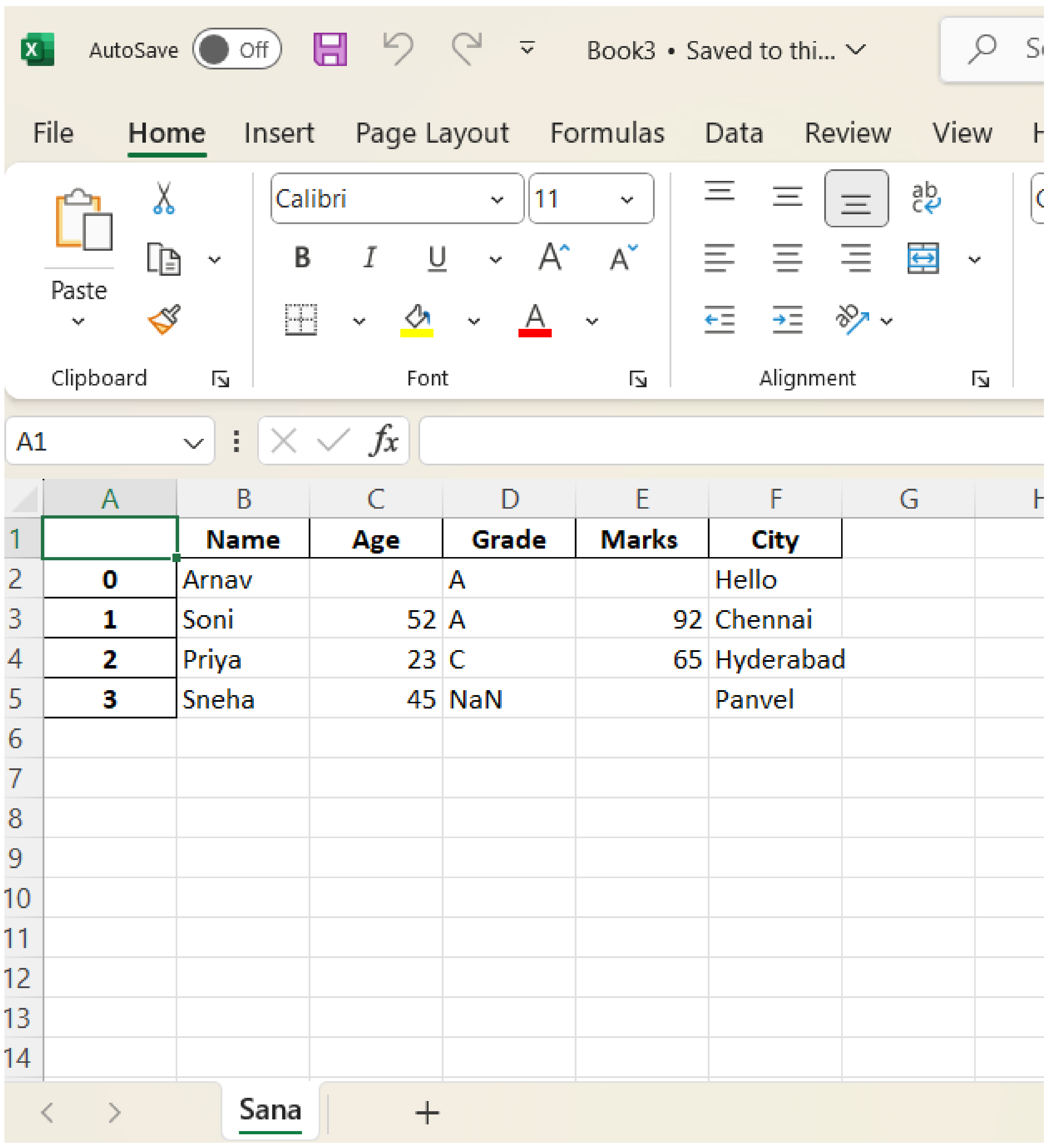
```
[34]:
    Name   Age  Grade  Marks       City
0  Arnav   NaN      A    NaN      Hello
1   Soni  52.0      A   92.0    Chennai
2  Priya  23.0      C   65.0  Hyderabad
3  Sneha  45.0    NaN    NaN     Panvel
```

What does df.to_excel("Book3.xlsx", sheet_name='Sana') do?

A: It saves the DataFrame to an Excel file named "Book3.xlsx" with the sheet name "Sana".

```
df.to_excel("Book3.xlsx",sheet_name='Sana')
```

OUTPUT:



	Name	Age	Grade	Marks	City
0	Arnav		A		Hello
1	Soni	52	A	92	Chennai
2	Priya	23	C	65	Hyderabad
3	Sneha	45	NaN		Panvel