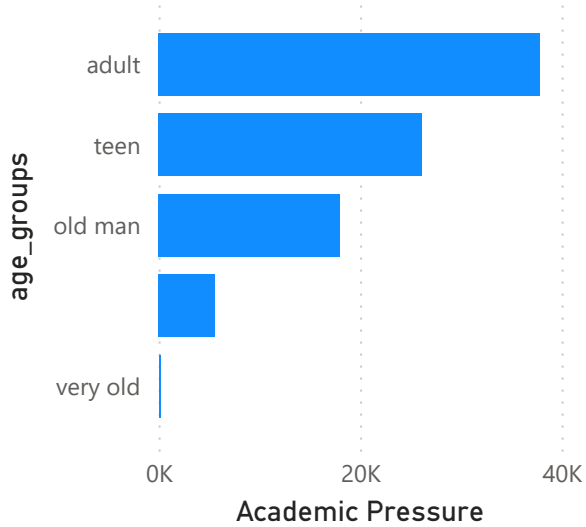
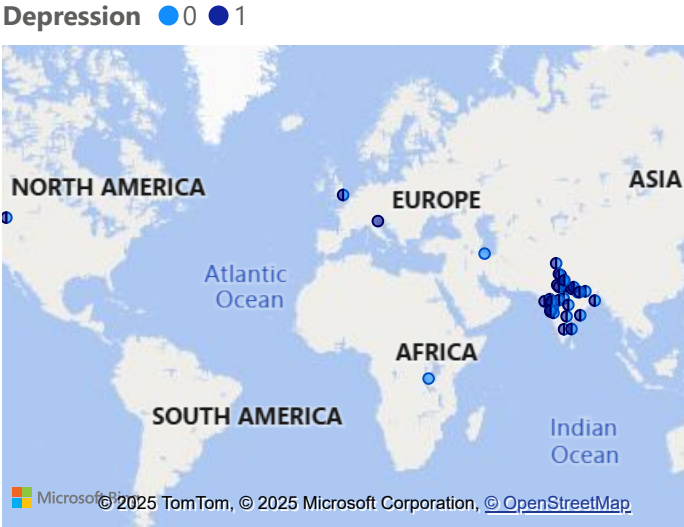


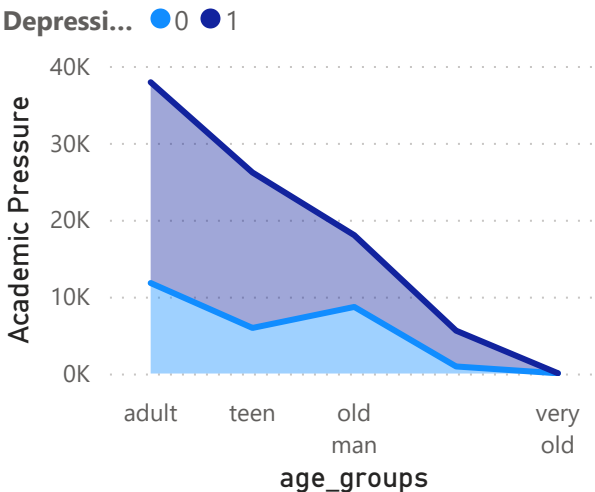
Academic Pressure by age_groups



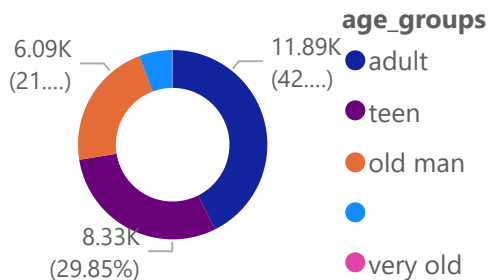
City and Depression



Academic Pressure by age_groups and Depression

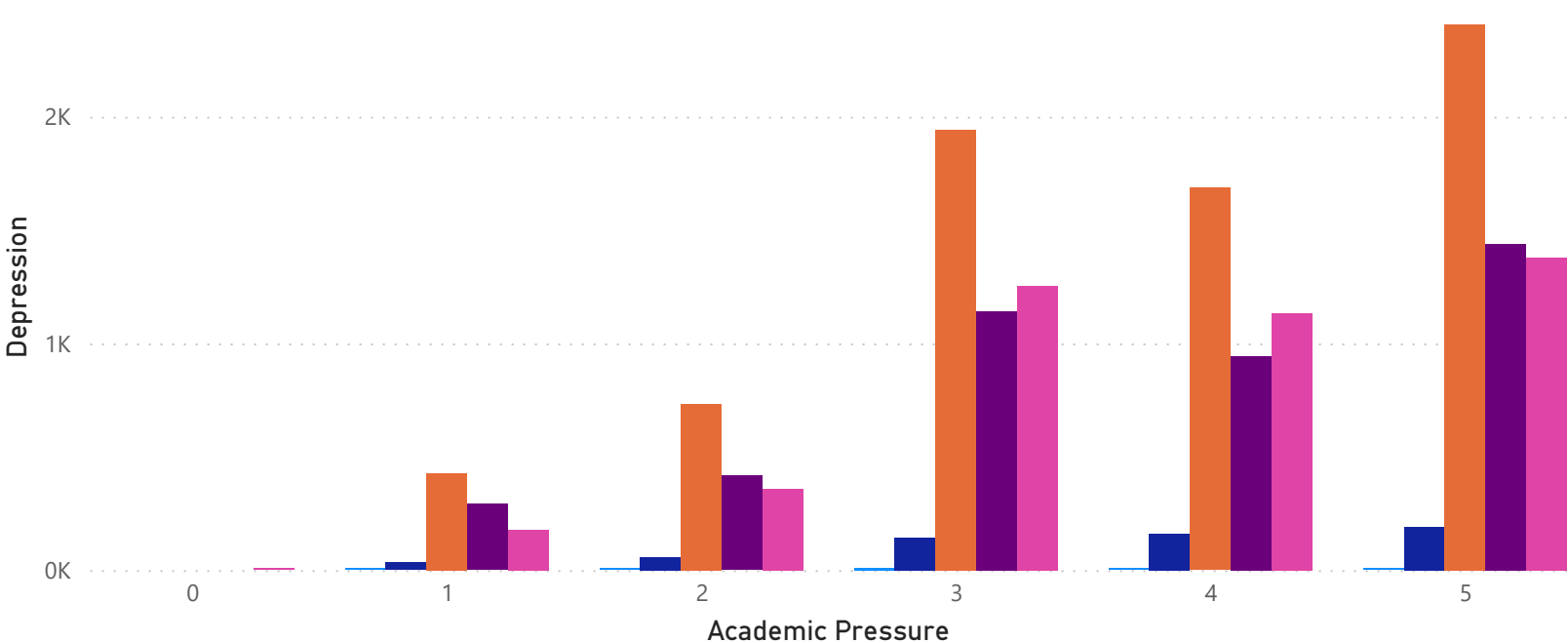


Count of age_groups and First age_groups by age_groups and age_groups

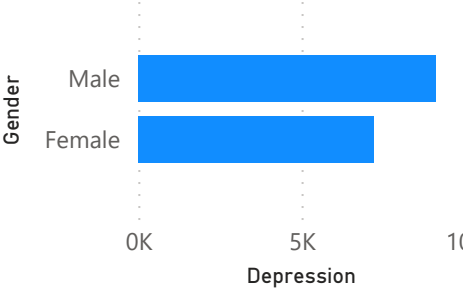


Depression by Academic Pressure and Degree_Level

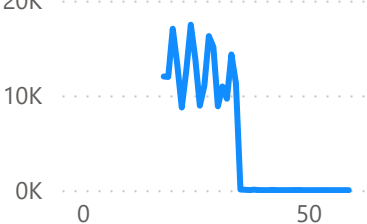
Degree_Level ● Doctorate ● Graduate ● Postgraduate ● Pre-University



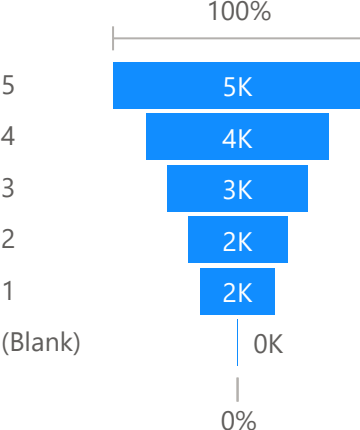
Depression by Gender



CGPA by Age



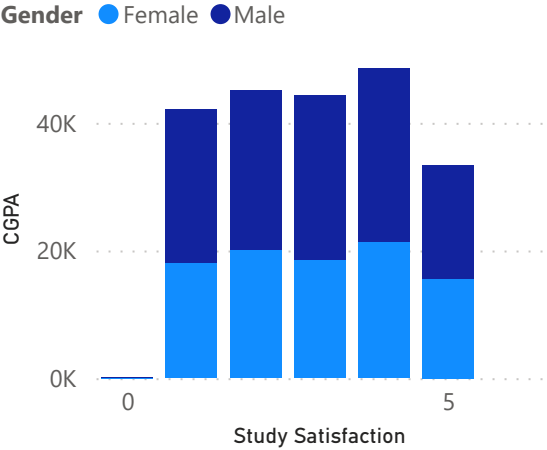
Depression by Financial Stress



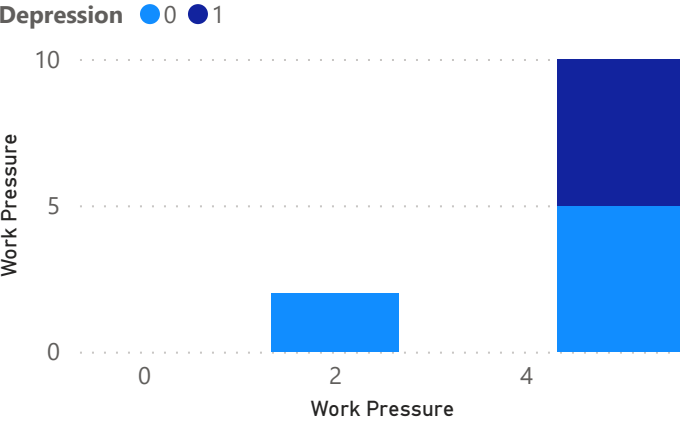
7.66

Average of CGPA

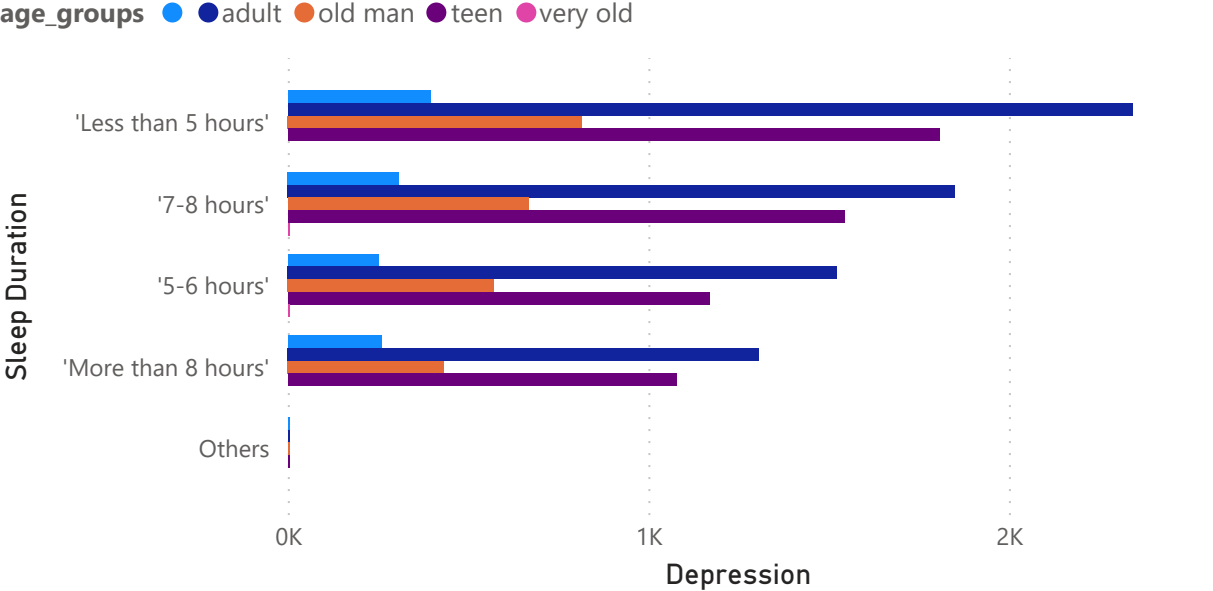
CGPA by Study Satisfaction and Gender



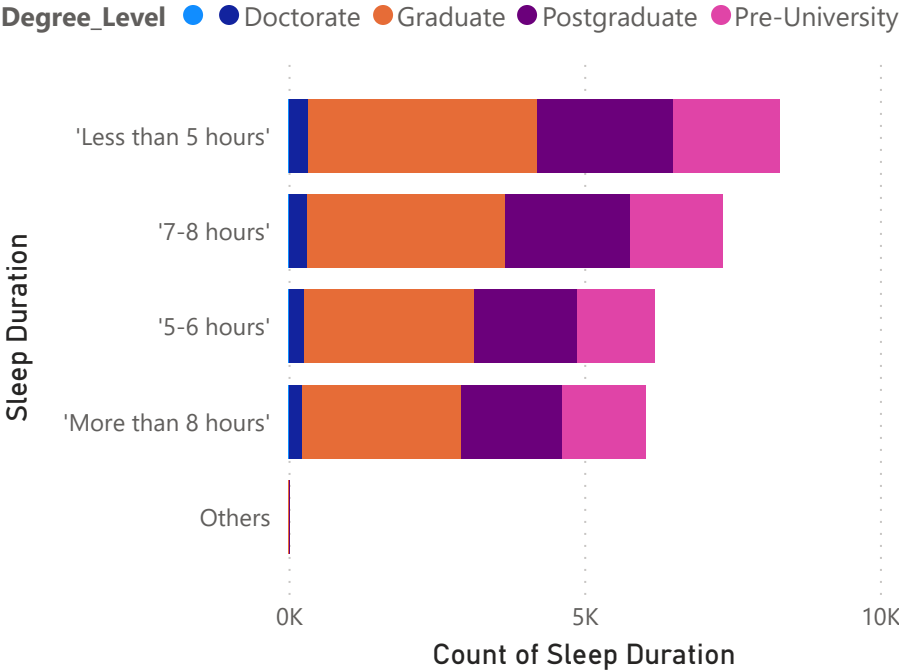
Work Pressure by Work Pressure and Depression



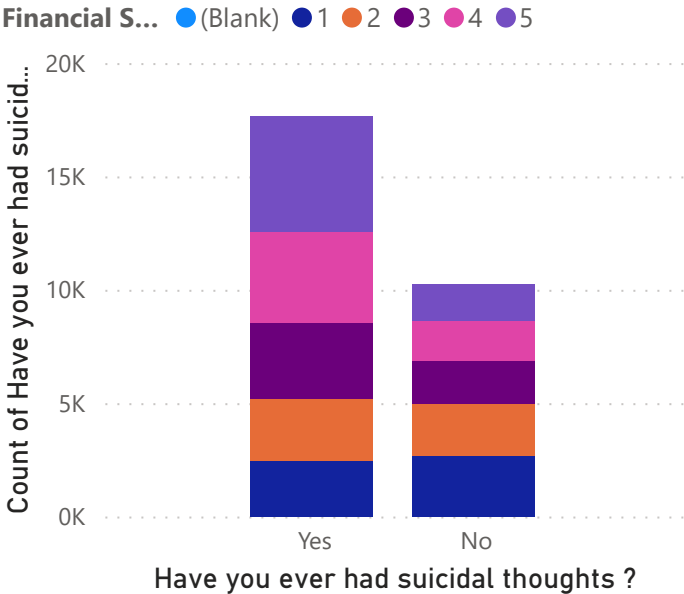
Depression by Sleep Duration and age_groups



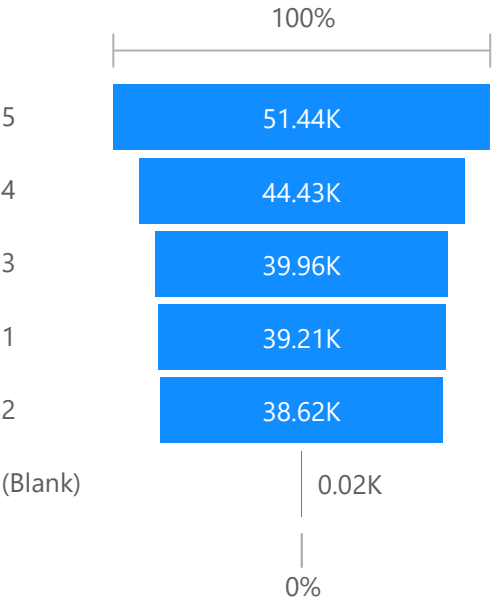
Count of Sleep Duration by Sleep Duration and Degree_Level



Count of Have you ever had suicidal thoughts ? by Have you ever had suicidal thoughts ? and Financial Stress



CGPA by Financial Stress



In [1]: `import pandas as pd`

In [2]: `df = pd.read_csv('student_depression_dataset.csv')`

In [3]: `df`

Out[3]:

	id	Gender	Age	City	Profession	Academic Pressure	Work Pressure	CGPA	Si
0	2	Male	33.0	Visakhapatnam	Student	5.0	0.0	8.97	
1	8	Female	24.0	Bangalore	Student	2.0	0.0	5.90	
2	26	Male	31.0	Srinagar	Student	3.0	0.0	7.03	
3	30	Female	28.0	Varanasi	Student	3.0	0.0	5.59	
4	32	Female	25.0	Jaipur	Student	4.0	0.0	8.13	
...
27896	140685	Female	27.0	Surat	Student	5.0	0.0	5.75	
27897	140686	Male	27.0	Ludhiana	Student	2.0	0.0	9.40	
27898	140689	Male	31.0	Faridabad	Student	3.0	0.0	6.61	
27899	140690	Female	18.0	Ludhiana	Student	5.0	0.0	6.88	
27900	140699	Male	27.0	Patna	Student	4.0	0.0	9.24	

27901 rows × 18 columns



In [4]: `df.info()`

```

<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  float64
 3   City                                       27901 non-null  object
 4   Profession                                27901 non-null  object
 5   Academic Pressure                         27901 non-null  float64
 6   Work Pressure                             27901 non-null  float64
 7   CGPA                                       27901 non-null  float64
 8   Study Satisfaction                       27901 non-null  float64
 9   Job Satisfaction                         27901 non-null  float64
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  float64
15   Financial Stress                         27901 non-null  object
16   Family History of Mental Illness         27901 non-null  object
17   Depression                               27901 non-null  int64
dtypes: float64(7), int64(2), object(9)
memory usage: 3.8+ MB

```

```
In [5]: df.duplicated().sum()
```

```
Out[5]: 0
```

```
In [6]: df['id'].duplicated().sum()
```

```
Out[6]: 0
```

```
In [7]: df.drop(columns=['id'], inplace=True)
```

```
In [8]: df
```

Out[8]:

	Gender	Age	City	Profession	Academic Pressure	Work Pressure	CGPA	Stud Satisfactio
0	Male	33.0	Visakhapatnam	Student	5.0	0.0	8.97	2.
1	Female	24.0	Bangalore	Student	2.0	0.0	5.90	5.
2	Male	31.0	Srinagar	Student	3.0	0.0	7.03	5.
3	Female	28.0	Varanasi	Student	3.0	0.0	5.59	2.
4	Female	25.0	Jaipur	Student	4.0	0.0	8.13	3.
...
27896	Female	27.0	Surat	Student	5.0	0.0	5.75	5.
27897	Male	27.0	Ludhiana	Student	2.0	0.0	9.40	3.
27898	Male	31.0	Faridabad	Student	3.0	0.0	6.61	4.
27899	Female	18.0	Ludhiana	Student	5.0	0.0	6.88	2.
27900	Male	27.0	Patna	Student	4.0	0.0	9.24	1.

27901 rows × 17 columns

In [9]: `df.groupby('Gender')['Age'].mean()`

Out[9]:

Gender	
Female	25.772381
Male	25.861967

Name: Age, dtype: float64

In [10]: `pd.crosstab(df['Gender'], df['City'])`

Out[10]:

	City	'Less Delhi'	'Less than 5 Kalyan'	3.0	Agra	Ahmedabad	Bangalore	Bhavna	Bhopal	Chennai
Gender										
Female		1	0	1	446	404	380	0	421	494
Male		0	1	0	648	547	387	2	513	391

2 rows × 52 columns

In [11]: `df['Profession'].unique()`

Out[11]: array(['Student', "'Civil Engineer'", 'Architect', "'UX/UI Designer'",
 "'Digital Marketer'", "'Content Writer'",
 "'Educational Consultant'", 'Teacher', 'Manager', 'Chef', 'Doctor',
 'Lawyer', 'Entrepreneur', 'Pharmacist'], dtype=object)

In [12]: `df['Profession'].value_counts()`

Out[12]: Profession
 Student 27870
 Architect 8
 Teacher 6
 'Digital Marketer' 3
 'Content Writer' 2
 Chef 2
 Doctor 2
 Pharmacist 2
 'Civil Engineer' 1
 'UX/UI Designer' 1
 'Educational Consultant' 1
 Manager 1
 Lawyer 1
 Entrepreneur 1
 Name: count, dtype: int64

In [13]: `df['Age'].unique()`

Out[13]: array([33., 24., 31., 28., 25., 29., 30., 27., 19., 20., 23., 18., 21.,
 22., 34., 32., 26., 39., 35., 42., 36., 58., 49., 38., 51., 44.,
 43., 46., 59., 54., 48., 56., 37., 41.])

In [14]: `df['City'].unique()`

Out[14]: array(['Visakhapatnam', 'Bangalore', 'Srinagar', 'Varanasi', 'Jaipur',
 'Pune', 'Thane', 'Chennai', 'Nagpur', 'Nashik', 'Vadodara',
 'Kalyan', 'Rajkot', 'Ahmedabad', 'Kolkata', 'Mumbai', 'Lucknow',
 'Indore', 'Surat', 'Ludhiana', 'Bhopal', 'Meerut', 'Agra',
 'Ghaziabad', 'Hyderabad', 'Vasai-Virar', 'Kanpur', 'Patna',
 'Faridabad', 'Delhi', 'Saanvi', 'M.Tech', 'Bhavna', "'Less Delhi'",
 'City', '3.0', "'Less than 5 Kalyan'", 'Mira', 'Harsha', 'Vaanya',
 'Gaurav', 'Harsh', 'Reyansh', 'Kibara', 'Rashi', 'ME', 'M.Com',
 'Nalyan', 'Mihir', 'Nalini', 'Nandini', 'Khaziabad'], dtype=object)

In [15]: `df['Degree'].unique()`

```
Out[15]: array(['B.Pharm', 'BSc', 'BA', 'BCA', 'M.Tech', 'PhD', "'Class 12'",
               'B.Ed', 'LLB', 'BE', 'M.Ed', 'MSc', 'BHM', 'M.Pharm', 'MCA', 'MA',
               'B.Com', 'MD', 'MBA', 'MBBS', 'M.Com', 'B.Arch', 'LLM', 'B.Tech',
               'BBA', 'ME', 'MHM', 'Others'], dtype=object)
```

```
In [16]: df[df['Degree'].isna() == True]
```

```
Out[16]:
```

Gender	Age	City	Profession	Academic Pressure	Work Pressure	CGPA	Study Satisfaction	Job Satisfaction

```
In [17]: df.loc[df['City'] == "M.Tech"]
```

```
Out[17]:
```

Gender	Age	City	Profession	Academic Pressure	Work Pressure	CGPA	Study Satisfaction	Satisf
5697	Male	29.0	M.Tech	Student	3.0	0.0	7.04	5.0

```
In [18]: df.loc[df['City'] == "M.Com"]
```

```
Out[18]:
```

Gender	Age	City	Profession	Academic Pressure	Work Pressure	CGPA	Study Satisfaction	Satis
22682	Female	26.0	M.Com	Student	4.0	0.0	8.95	1.0

```
In [19]: df[df['City'] == 'ME']['City'] = df['City'].mode()
```

C:\Users\lenovo\AppData\Local\Temp\ipykernel_576\661363757.py:1: SettingWithCopyWarning:
 A value is trying to be set on a copy of a slice from a DataFrame.
 Try using .loc[row_indexer,col_indexer] = value instead


 See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
 df[df['City'] == 'ME']['City'] = df['City'].mode()

```
In [20]: df.loc[df['City'] == 'ME', 'City'] = df['City'].mode()
```

```
In [21]: df.loc[df['City'] == "ME"]
```

Out[21]:


Gender	Age	City	Profession	Academic Pressure	Work Pressure	CGPA	Study Satisfaction	Job Satisfaction
--------	-----	------	------------	-------------------	---------------	------	--------------------	------------------

◀  ▶

In [22]: `df.loc[df['City'] == "M.Com", 'City'] = df['City'].mode()`In [23]: `df['City'].mode()`Out[23]: 0 Kalyan
Name: City, dtype: objectIn [24]: `df.loc[df['City'] == "3.0", 'City'] = df['City'].mode()`In [25]: `df.loc[df['City'] == "3.0"]`

Out[25]:

Gender	Age	City	Profession	Academic Pressure	Work Pressure	CGPA	Study Satisfaction	Job Satisfaction
--------	-----	------	------------	-------------------	---------------	------	--------------------	------------------


◀  ▶

In [26]: `df.loc[df['City'] == "'Less than 5 Kalyan'"]`

Out[26]:

Gender	Age	City	Profession	Academic Pressure	Work Pressure	CGPA	Study Satisfaction	Satisf
--------	-----	------	------------	-------------------	---------------	------	--------------------	--------


7355	Male	31.0	'Less than 5 Kalyan'	Student	1.0	0.0	7.21	1.0
------	------	------	----------------------	---------	-----	-----	------	-----

◀  ▶

In [27]: `df.loc[df['City'] == "'Less than 5 Kalyan'", 'City'] = df['City'].mode()`In [28]: `df[df['City'] == "'Less than 5 Kalyan'"]`

Out[28]:

Gender	Age	City	Profession	Academic Pressure	Work Pressure	CGPA	Study Satisfaction	Job Satisfaction
--------	-----	------	------------	-------------------	---------------	------	--------------------	------------------

◀  ▶


```
In [29]: df['City'].value_counts()
```

```
Out[29]: 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
Vadodara    694
Varanasi    685
Nagpur      651
Indore      643
Kanpur      609
Nashik      547
Faridabad   461
City        2
Harsha      2
Saanvi      2
Bhavna      2
'Less Delhi' 1
Mira        1
M.Tech      1
Vaanya      1
Gaurav      1
Harsh       1
Reyansh     1
Kibara      1
Rashi       1
Nalyan      1
Mihir       1
Nalini      1
Nandini     1
Khaziabad   1
Name: count, dtype: int64
```

```
In [30]: df['Academic Pressure'].unique()
```

```
Out[30]: array([5., 2., 3., 4., 1., 0.])
```

```
In [31]: df['Academic Pressure'].value_counts()
```

```
Out[31]: Academic Pressure
3.0    7462
5.0    6296
4.0    5155
1.0    4801
2.0    4178
0.0      9
Name: count, dtype: int64
```

```
In [32]: df['Work Pressure'].unique()
```

```
Out[32]: array([0., 5., 2.])
```

```
In [33]: df['Work Pressure'].value_counts()
```

```
Out[33]: Work Pressure
0.0    27898
5.0      2
2.0      1
Name: count, dtype: int64
```

```
In [34]: df['CGPA'].nunique() #not categeorical
```

```
Out[34]: 332
```

```
In [35]: df['Study Satisfaction'].unique()
```

```
Out[35]: array([2., 5., 3., 4., 1., 0.])
```

```
In [36]: df['Job Satisfaction'].unique()
```

```
Out[36]: array([0., 3., 4., 2., 1.])
```

```
In [37]: df['Sleep Duration'].unique()
```

```
Out[37]: array(['5-6 hours', 'Less than 5 hours', '7-8 hours',
               'More than 8 hours', 'Others'], dtype=object)
```

```
In [38]: df['Dietary Habits'].unique()
```

```
Out[38]: array(['Healthy', 'Moderate', 'Unhealthy', 'Others'], dtype=object)
```

```
In [39]: df['Degree'].unique()
```

```
Out[39]: array(['B.Pharm', 'BSc', 'BA', 'BCA', 'M.Tech', 'PhD', 'Class 12',
               'B.Ed', 'LLB', 'BE', 'M.Ed', 'MSc', 'BHM', 'M.Pharm', 'MCA', 'MA',
               'B.Com', 'MD', 'MBA', 'MBBS', 'M.Com', 'B.Arch', 'LLM', 'B.Tech',
               'BBA', 'ME', 'MHM', 'Others'], dtype=object)
```

```
In [40]: df['Work/Study Hours'].unique()
```

```
Out[40]: array([ 3.,  9.,  4.,  1.,  0., 12.,  2., 11., 10.,  6.,  8.,  5.,  7.])
```

```
In [41]: df['Financial Stress'].unique()
```

```
Out[41]: array(['1.0', '2.0', '5.0', '3.0', '4.0', '?'], dtype=object)
```

In [42]: `#all are categorical except age and cgpa`

In [43]: `df`

Out[43]:

	Gender	Age	City	Profession	Academic Pressure	Work Pressure	CGPA	Stud Satisfactio
0	Male	33.0	Visakhapatnam	Student	5.0	0.0	8.97	2.
1	Female	24.0	Bangalore	Student	2.0	0.0	5.90	5.
2	Male	31.0	Srinagar	Student	3.0	0.0	7.03	5.
3	Female	28.0	Varanasi	Student	3.0	0.0	5.59	2.
4	Female	25.0	Jaipur	Student	4.0	0.0	8.13	3.
...
27896	Female	27.0	Surat	Student	5.0	0.0	5.75	5.
27897	Male	27.0	Ludhiana	Student	2.0	0.0	9.40	3.
27898	Male	31.0	Faridabad	Student	3.0	0.0	6.61	4.
27899	Female	18.0	Ludhiana	Student	5.0	0.0	6.88	2.
27900	Male	27.0	Patna	Student	4.0	0.0	9.24	1.

27901 rows × 17 columns



In [44]: `df['Sleep Duration'].unique()`

Out[44]: `array([''5-6 hours'', ''Less than 5 hours'', ''7-8 hours'', ''More than 8 hours'', 'Others'], dtype=object)`

In [45]: `df.loc[df['Sleep Duration'] == 'Kalyan']`

Out[45]:

Gender	Age	City	Profession	Academic Pressure	Work Pressure	CGPA	Study Satisfaction	Job Satisfaction
--------	-----	------	------------	-------------------	---------------	------	--------------------	------------------

In [46]: `df['Family History of Mental Illness'].unique()`Out[46]: `array(['No', 'Yes'], dtype=object)`In [47]: `df.isna().sum()`

```
Out[47]: Gender                0
Age                0
City               4
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    0
Family History of Mental Illness      0
Depression         0
dtype: int64
```

In [48]: `df[df['City'].isnull()]`

Out[48]:

	Gender	Age	City	Profession	Academic Pressure	Work Pressure	CGPA	Study Satisfaction	Satisfa
7021	Female	21.0	NaN	Student	3.0	0.0	8.69	4.0	
7355	Male	31.0	NaN	Student	1.0	0.0	7.21	1.0	
18377	Female	25.0	NaN	Student	2.0	0.0	6.00	2.0	
22682	Female	26.0	NaN	Student	4.0	0.0	8.95	1.0	

In [49]: `df['City'].fillna(df['City'].mode()[0], inplace=True)`

C:\Users\lenovo\AppData\Local\Temp\ipykernel_576\1488218936.py:1: FutureWarning:
A value is trying to be set on a copy of a DataFrame or Series through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value) instead, to perform the operation inplace on the original object.

```
df['City'].fillna(df['City'].mode()[0], inplace=True)
```

```
In [50]: df[df['City'].isna() == True]
```

```
Out[50]:
```

Gender	Age	City	Profession	Academic Pressure	Work Pressure	CGPA	Study Satisfaction	Job Satisfaction
--------	-----	------	------------	-------------------	---------------	------	--------------------	------------------

```
In [51]: pd.crosstab(df['Gender'], df['Profession'])
```

Profession	'Civil Engineer'	'Content Writer'	'Digital Marketer'	'Educational Consultant'	'UX/UI Designer'	Architect	Chef	Doctor
Gender								
Female	0	1	2	0	1	5	2	
Male	1	1	1	1	0	3	0	

```
In [52]: pd.crosstab(df['Gender'], df['Profession'], values = df['Age'], aggfunc = "mean")
```

Profession	'Civil Engineer'	'Content Writer'	'Digital Marketer'	'Educational Consultant'	'UX/UI Designer'	Architect	Chef	Doctor
Gender								
Female	NaN	34.0	24.5	NaN	32.0	26.000000	26.0	
Male	21.0	25.0	28.0	32.0	NaN	27.666667	NaN	

```
In [53]: df
```

Out[53]:

	Gender	Age	City	Profession	Academic Pressure	Work Pressure	CGPA	Stud Satisfactio
0	Male	33.0	Visakhapatnam	Student	5.0	0.0	8.97	2.
1	Female	24.0	Bangalore	Student	2.0	0.0	5.90	5.
2	Male	31.0	Srinagar	Student	3.0	0.0	7.03	5.
3	Female	28.0	Varanasi	Student	3.0	0.0	5.59	2.
4	Female	25.0	Jaipur	Student	4.0	0.0	8.13	3.
...
27896	Female	27.0	Surat	Student	5.0	0.0	5.75	5.
27897	Male	27.0	Ludhiana	Student	2.0	0.0	9.40	3.
27898	Male	31.0	Faridabad	Student	3.0	0.0	6.61	4.
27899	Female	18.0	Ludhiana	Student	5.0	0.0	6.88	2.
27900	Male	27.0	Patna	Student	4.0	0.0	9.24	1.

27901 rows × 17 columns

In [54]: `pd.crosstab(df['Gender'], df['Profession'], values = df['Academic Pressure'], ag`

Out[54]:

Profession	'Civil Engineer'	'Content Writer'	'Digital Marketer'	'Educational Consultant'	'UX/UI Designer'	Architect	Chef	Do
Gender								
Female	NaN	5.0	5.0	NaN	3.0	3.800000	2.5	
Male	5.0	5.0	1.0	3.0	NaN	3.666667	NaN	

In [55]: `pd.crosstab(df['Age'], df['Academic Pressure'], values = df['Academic Pressure']`

Out[55]: **Academic Pressure** **0.0** **1.0** **2.0** **3.0** **4.0** **5.0**

Age							
18.0	0.0	1.0	2.0	3.0	4.0	5.0	
19.0	NaN	1.0	2.0	3.0	4.0	5.0	
20.0	0.0	1.0	2.0	3.0	4.0	5.0	
21.0	0.0	1.0	2.0	3.0	4.0	5.0	
22.0	NaN	1.0	2.0	3.0	4.0	5.0	
23.0	NaN	1.0	2.0	3.0	4.0	5.0	
24.0	0.0	1.0	2.0	3.0	4.0	5.0	
25.0	NaN	1.0	2.0	3.0	4.0	5.0	
26.0	NaN	1.0	2.0	3.0	4.0	5.0	
27.0	NaN	1.0	2.0	3.0	4.0	5.0	
28.0	NaN	1.0	2.0	3.0	4.0	5.0	
29.0	NaN	1.0	2.0	3.0	4.0	5.0	
30.0	0.0	1.0	2.0	3.0	4.0	5.0	
31.0	NaN	1.0	2.0	3.0	4.0	5.0	
32.0	NaN	1.0	2.0	3.0	4.0	5.0	
33.0	NaN	1.0	2.0	3.0	4.0	5.0	
34.0	NaN	1.0	2.0	3.0	4.0	5.0	
35.0	NaN	1.0	NaN	3.0	4.0	5.0	
36.0	0.0	1.0	NaN	3.0	NaN	5.0	
37.0	NaN	1.0	NaN	NaN	NaN	NaN	
38.0	0.0	1.0	2.0	3.0	NaN	5.0	
39.0	NaN	NaN	NaN	NaN	4.0	5.0	
41.0	NaN	NaN	NaN	3.0	NaN	NaN	
42.0	NaN	NaN	2.0	NaN	NaN	5.0	
43.0	NaN	NaN	NaN	NaN	NaN	5.0	
44.0	NaN	1.0	NaN	NaN	NaN	NaN	
46.0	NaN	1.0	NaN	3.0	NaN	NaN	
48.0	NaN	1.0	NaN	3.0	NaN	NaN	
49.0	NaN	NaN	NaN	3.0	NaN	NaN	
51.0	NaN	NaN	2.0	NaN	NaN	NaN	
54.0	NaN	NaN	NaN	NaN	NaN	5.0	
56.0	NaN	NaN	NaN	3.0	NaN	NaN	

Academic Pressure 0.0 1.0 2.0 3.0 4.0 5.0

Age

58.0 NaN NaN NaN NaN 4.0 NaN

59.0 NaN 1.0 NaN NaN NaN NaN

In [56]: `df.head()`

Out[56]:

	Gender	Age	City	Profession	Academic Pressure	Work Pressure	CGPA	Study Satisfaction	Si
0	Male	33.0	Visakhapatnam	Student	5.0	0.0	8.97	2.0	
1	Female	24.0	Bangalore	Student	2.0	0.0	5.90	5.0	
2	Male	31.0	Srinagar	Student	3.0	0.0	7.03	5.0	
3	Female	28.0	Varanasi	Student	3.0	0.0	5.59	2.0	
4	Female	25.0	Jaipur	Student	4.0	0.0	8.13	3.0	

In [57]: `df['City'].unique()`

Out[57]: array(['Visakhapatnam', 'Bangalore', 'Srinagar', 'Varanasi', 'Jaipur', 'Pune', 'Thane', 'Chennai', 'Nagpur', 'Nashik', 'Vadodara', 'Kalyan', 'Rajkot', 'Ahmedabad', 'Kolkata', 'Mumbai', 'Lucknow', 'Indore', 'Surat', 'Ludhiana', 'Bhopal', 'Meerut', 'Agra', 'Ghaziabad', 'Hyderabad', 'Vasai-Virar', 'Kanpur', 'Patna', 'Faridabad', 'Delhi', 'Saanvi', 'M.Tech', 'Bhavna', '"Less Delhi"', 'City', 'Mira', 'Harsha', 'Vaanya', 'Gaurav', 'Harsh', 'Reyansh', 'Kibara', 'Rashi', 'Nalyan', 'Mihir', 'Nalini', 'Nandini', 'Khaziabad'], dtype=object)

In [58]: `df.loc[df['City'] == '"Less Delhi"]`

Out[58]:

	Gender	Age	City	Profession	Academic Pressure	Work Pressure	CGPA	Study Satisfaction	Satisfac
6010	Female	29.0	'Less Delhi'	Student	2.0	0.0	7.04	3.0	


```
In [59]: df.loc[df['City'] == "'Less Delhi'", 'City'] = 'Delhi'
```

```
In [60]: df.loc[df['City'] == "'Less Delhi'"]
```

Out[60]:

Gender	Age	City	Profession	Academic Pressure	Work Pressure	CGPA	Study Satisfaction	Job Satisfaction
--------	-----	------	------------	----------------------	------------------	------	-----------------------	---------------------



```
In [61]: df['Age'].unique()
```

```
Out[61]: array([33., 24., 31., 28., 25., 29., 30., 27., 19., 20., 23., 18., 21.,  
                22., 34., 32., 26., 39., 35., 42., 36., 58., 49., 38., 51., 44.,  
                43., 46., 59., 54., 48., 56., 37., 41.])
```

```
In [62]: pd.crosstab(df['Age'], df['Gender'], values=df['CGPA'], aggfunc='mean')
```

Out[62]:

Gender	Female	Male
Age		
18.0	7.474971	7.609261
19.0	7.588110	7.725241
20.0	7.557442	7.665872
21.0	7.643620	7.707397
22.0	7.470214	7.578783
23.0	7.572833	7.754555
24.0	7.726384	7.734716
25.0	7.696780	7.824478
26.0	7.732101	7.719987
27.0	7.414327	7.638639
28.0	7.533707	7.701796
29.0	7.738543	7.752194
30.0	7.740112	7.730984
31.0	7.621725	7.715946
32.0	7.382475	7.819287
33.0	7.493252	7.616163
34.0	7.691617	7.670869
35.0	7.618571	6.860000
36.0	7.745000	7.083333
37.0	9.410000	5.410000
38.0	8.186667	5.682000
39.0	5.903333	NaN
41.0	NaN	5.850000
42.0	9.295000	5.710000
43.0	9.670000	5.640000
44.0	6.830000	NaN
46.0	6.830000	6.100000
48.0	7.240000	NaN
49.0	NaN	6.470000
51.0	8.260000	NaN
54.0	NaN	9.600000
56.0	7.940000	NaN

Gender	Female	Male
Age		
58.0	8.580000	NaN
59.0	NaN	8.140000

In [63]: `df["age_groups"] = pd.cut(df['Age'], bins= (18,23,30,42,60), labels= ["teen", "`

In [64]: `df`

Out[64]:

	Gender	Age	City	Profession	Academic Pressure	Work Pressure	CGPA	Stud Satisfactio
0	Male	33.0	Visakhapatnam	Student	5.0	0.0	8.97	2.
1	Female	24.0	Bangalore	Student	2.0	0.0	5.90	5.
2	Male	31.0	Srinagar	Student	3.0	0.0	7.03	5.
3	Female	28.0	Varanasi	Student	3.0	0.0	5.59	2.
4	Female	25.0	Jaipur	Student	4.0	0.0	8.13	3.
...
27896	Female	27.0	Surat	Student	5.0	0.0	5.75	5.
27897	Male	27.0	Ludhiana	Student	2.0	0.0	9.40	3.
27898	Male	31.0	Faridabad	Student	3.0	0.0	6.61	4.
27899	Female	18.0	Ludhiana	Student	5.0	0.0	6.88	2.
27900	Male	27.0	Patna	Student	4.0	0.0	9.24	1.

27901 rows × 18 columns



In [65]: `pd.crosstab(df['age_groups'], df['Gender'], values=df['CGPA'], aggfunc='mean')`

Out[65]:

	Gender	Female	Male
--	--------	--------	------

age_groups

teen	7.569767	7.693716
-------------	----------	----------

adult	7.649622	7.731417
--------------	----------	----------

old man	7.557461	7.688630
----------------	----------	----------

very old	7.758889	7.190000
-----------------	----------	----------

In [66]: `new_tab = df.groupby(['age_groups', 'Gender'])['Academic Pressure'].mean()`

C:\Users\lenovo\AppData\Local\Temp\ipykernel_576\1715857613.py:1: FutureWarning:
The default of observed=False is deprecated and will be changed to True in a future version of pandas. Pass observed=False to retain current behavior or observed=True to adopt the future default and silence this warning.
`new_tab = df.groupby(['age_groups', 'Gender'])['Academic Pressure'].mean()`

In [67]: `pd.DataFrame(new_tab)`

Out[67]:

Academic Pressure		
-------------------	--	--

age_groups	Gender	
------------	--------	--

teen	Female	3.183038
-------------	---------------	----------

Male	3.104520
-------------	----------

adult	Female	3.208458
--------------	---------------	----------

Male	3.170105
-------------	----------

old man	Female	3.008092
----------------	---------------	----------

Male	2.918911
-------------	----------

very old	Female	2.777778
-----------------	---------------	----------

Male	3.000000
-------------	----------

In [68]: `pd.DataFrame(df.groupby(['age_groups', 'Gender'])['Study Satisfaction'].mean())`

C:\Users\lenovo\AppData\Local\Temp\ipykernel_576\3792105466.py:1: FutureWarning:
The default of observed=False is deprecated and will be changed to True in a future version of pandas. Pass observed=False to retain current behavior or observed=True to adopt the future default and silence this warning.
`pd.DataFrame(df.groupby(['age_groups', 'Gender'])['Study Satisfaction'].mean())`

Out[68]: **Study Satisfaction**

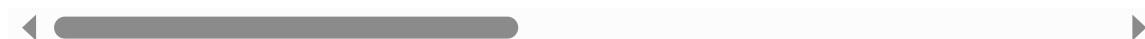
age_groups	Gender	
teen	Female	2.929683
	Male	2.837462
adult	Female	2.934319
	Male	2.900352
old man	Female	3.003083
	Male	3.025215
very old	Female	3.000000
	Male	2.200000

In [69]: df

Out[69]:

	Gender	Age	City	Profession	Academic Pressure	Work Pressure	CGPA	Stud Satisfactio
0	Male	33.0	Visakhapatnam	Student	5.0	0.0	8.97	2.
1	Female	24.0	Bangalore	Student	2.0	0.0	5.90	5.
2	Male	31.0	Srinagar	Student	3.0	0.0	7.03	5.
3	Female	28.0	Varanasi	Student	3.0	0.0	5.59	2.
4	Female	25.0	Jaipur	Student	4.0	0.0	8.13	3.
...
27896	Female	27.0	Surat	Student	5.0	0.0	5.75	5.
27897	Male	27.0	Ludhiana	Student	2.0	0.0	9.40	3.
27898	Male	31.0	Faridabad	Student	3.0	0.0	6.61	4.
27899	Female	18.0	Ludhiana	Student	5.0	0.0	6.88	2.
27900	Male	27.0	Patna	Student	4.0	0.0	9.24	1.

27901 rows × 18 columns

In [70]: `df['Sleep Duration'].unique()`Out[70]: `array(['5-6 hours', 'Less than 5 hours', '7-8 hours', 'More than 8 hours', 'Others'], dtype=object)`In [71]: `pd.crosstab(df['age_groups'], df['Sleep Duration'])`

Out[71]:

	Sleep Duration	'5-6 hours'	'7-8 hours'	'Less than 5 hours'	'More than 8 hours'	Others
age_groups						
teen		1797	2263	2476	1788	4
adult		2663	3081	3570	2566	7
old man		1380	1592	1778	1329	6
very old		5	5	2	2	0

In [72]: `pd.crosstab(df['Family History of Mental Illness'], df['Gender'])`

Out[72]:

	Gender	
	Female	Male
Family History of Mental Illness		
No	6266	8132
Yes	6088	7415

In [73]: `df.columns`

Out[73]: Index(['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', 'age_groups'], dtype='object')

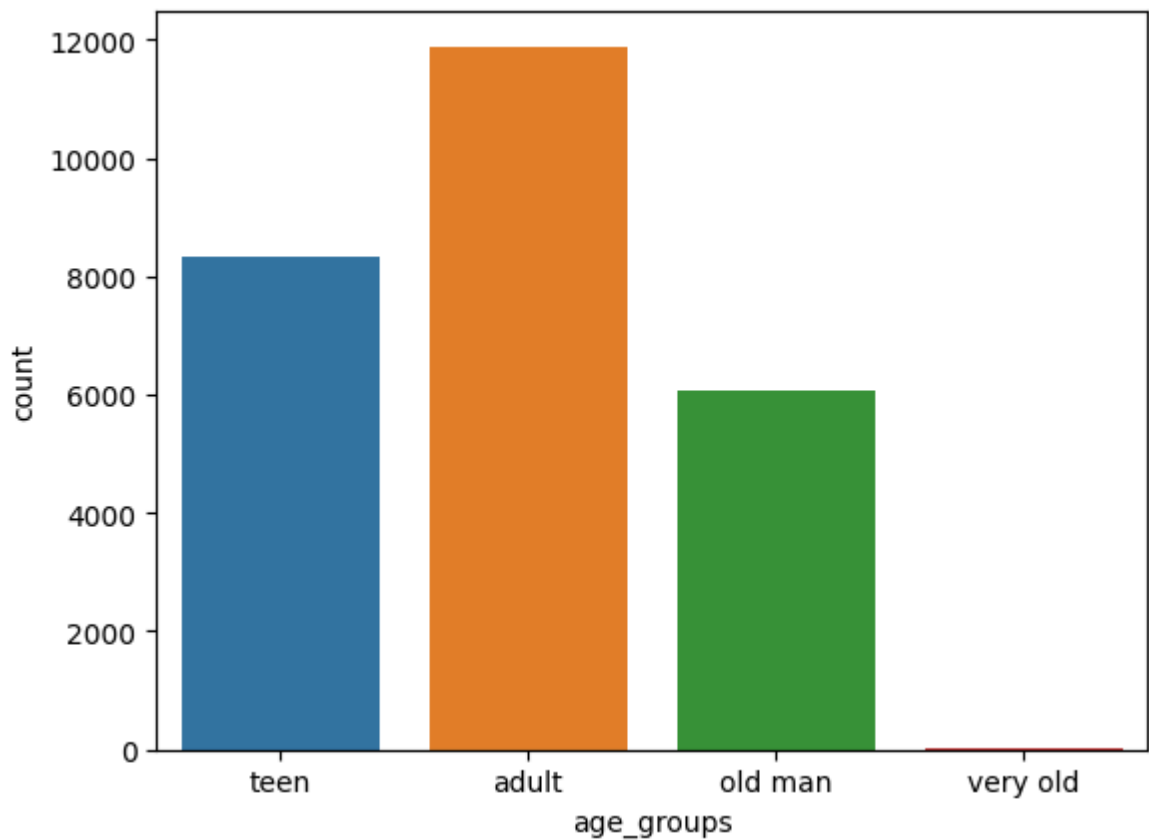
In [74]: `import seaborn as sns`
`import matplotlib.pyplot as plt`

In [75]: `sns.countplot(data=df, x='age_groups')`

c:\Users\lenovo\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\categorical.py:641: FutureWarning: The default of observed=False is deprecated and will be changed to True in a future version of pandas. Pass observed=False to retain current behavior or observed=True to adopt the future default and silence this warning.

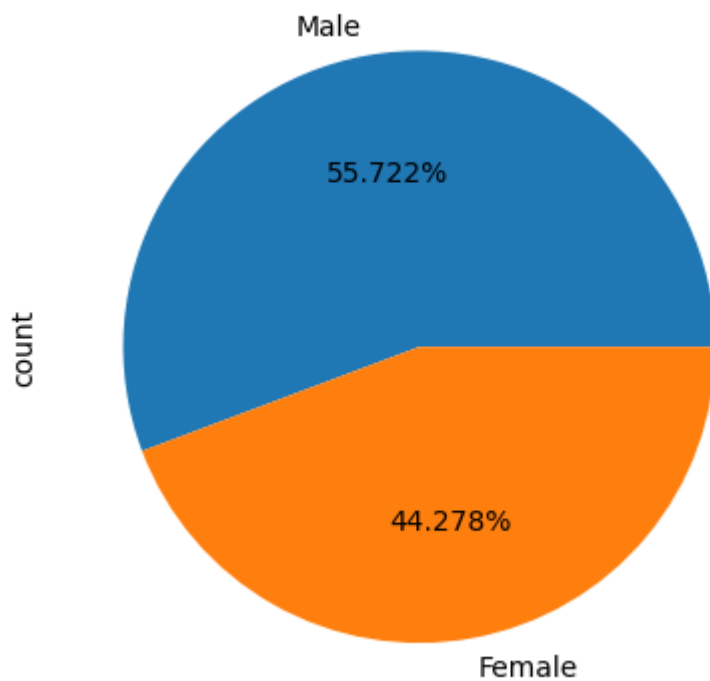
grouped_vals = vals.groupby(grouper)

Out[75]: `<Axes: xlabel='age_groups', ylabel='count'>`

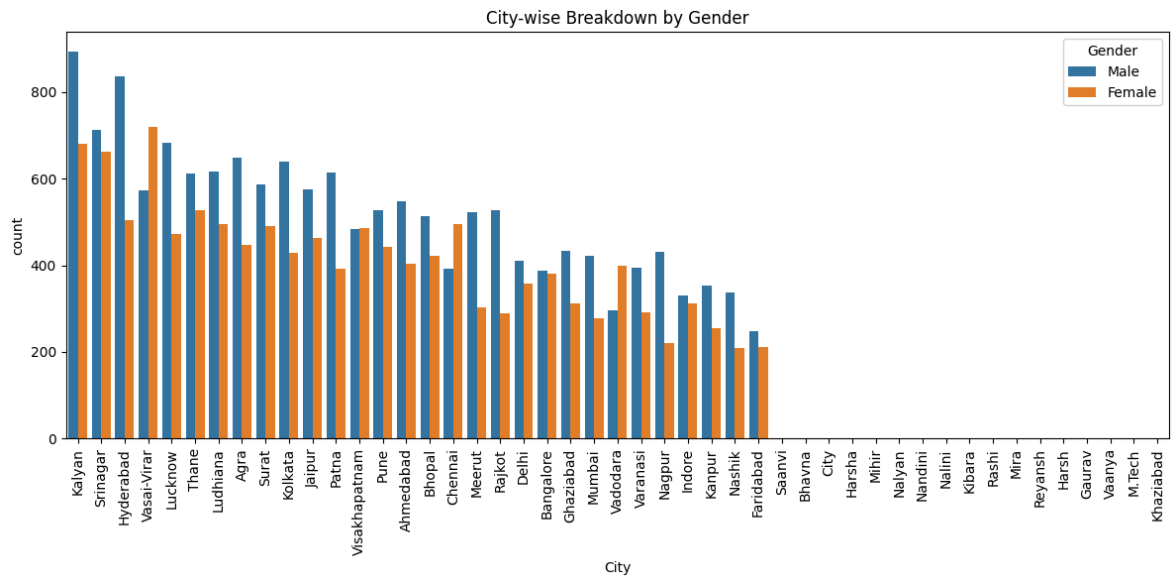


```
In [76]: df.Gender.value_counts().plot.pie(autopct = "%0.3f%")
```

```
Out[76]: <Axes: ylabel='count'>
```

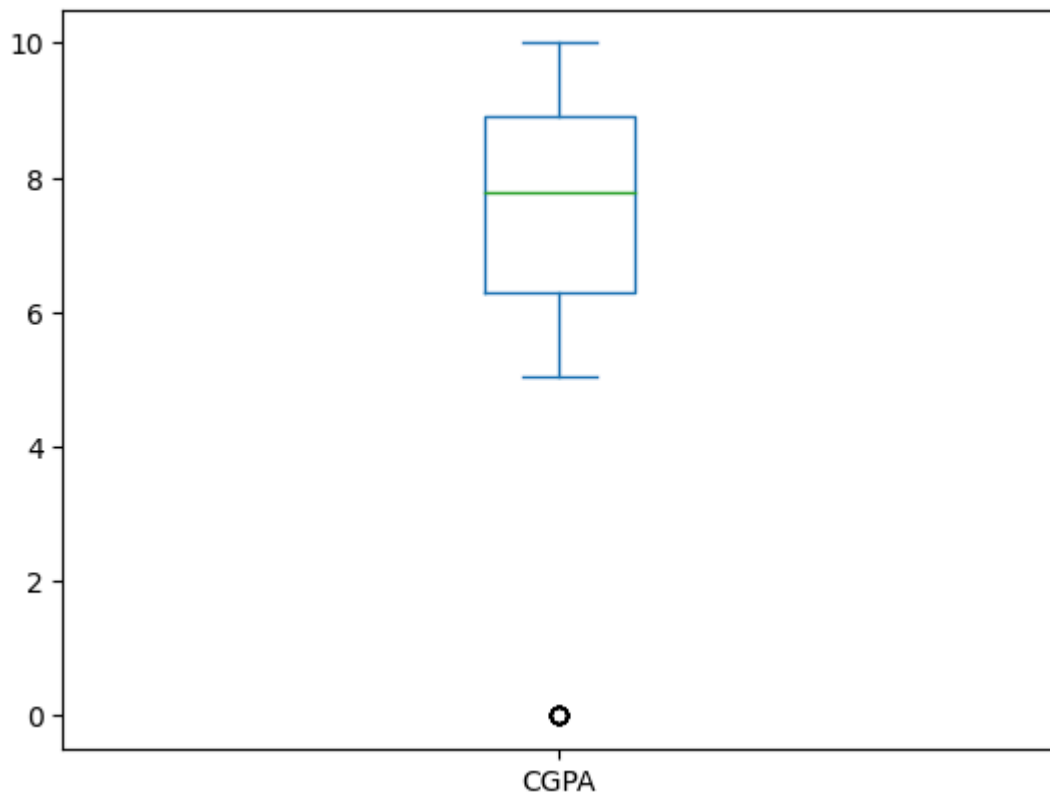


```
In [77]: plt.figure(figsize=(12, 6))
sns.countplot(data=df, x='City', hue='Gender', order=df['City'].value_counts().i
plt.title('City-wise Breakdown by Gender')
plt.xticks(rotation=90)
plt.tight_layout()
plt.show()
```

```
In [78]: df['CGPA'].plot.box()
```

```
Out[78]: <Axes: >
```



```
In [79]: gpalimit = df.CGPA.mean() + 3 * df.CGPA.std()
gpalimit
```

```
Out[79]: 12.068226210516325
```

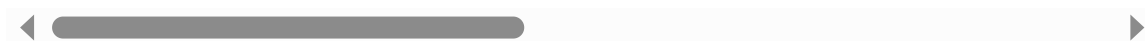
```
In [80]: gpalimit2 = df.CGPA.mean() - 3 * df.CGPA.std()
gpalimit2
```

```
Out[80]: 3.2439821332706344
```

```
In [81]: df[df['CGPA'] < gpalimit2]
```

Out[81]:

	Gender	Age	City	Profession	Academic Pressure	Work Pressure	CGPA	Study Satisfaction
4365	Male	38.0	Chennai	Student	0.0	5.0	0.0	0.0
10407	Male	28.0	Vadodara	Student	3.0	0.0	0.0	4.0
11489	Female	32.0	Vadodara	Student	5.0	0.0	0.0	0.0
13909	Female	20.0	Ahmedabad	Student	0.0	0.0	0.0	0.0
14855	Male	21.0	Lucknow	Student	0.0	2.0	0.0	0.0
20913	Male	18.0	Ahmedabad	Student	0.0	0.0	0.0	0.0
21805	Male	34.0	Pune	Student	1.0	0.0	0.0	5.0
25746	Male	18.0	Rajkot	Student	0.0	5.0	0.0	0.0
26719	Female	24.0	Meerut	Student	0.0	0.0	0.0	0.0

In [82]: `len(df[df['CGPA'] < gpalimit2])`

Out[82]: 9

In [83]: `df.loc[df['CGPA'] < gpalimit2, 'CGPA'] = df['CGPA'].mean()`In [84]: `df[df['CGPA'] < gpalimit2]`

Out[84]:

	Gender	Age	City	Profession	Academic Pressure	Work Pressure	CGPA	Study Satisfaction	Job Satisfaction
--	--------	-----	------	------------	-------------------	---------------	------	--------------------	------------------

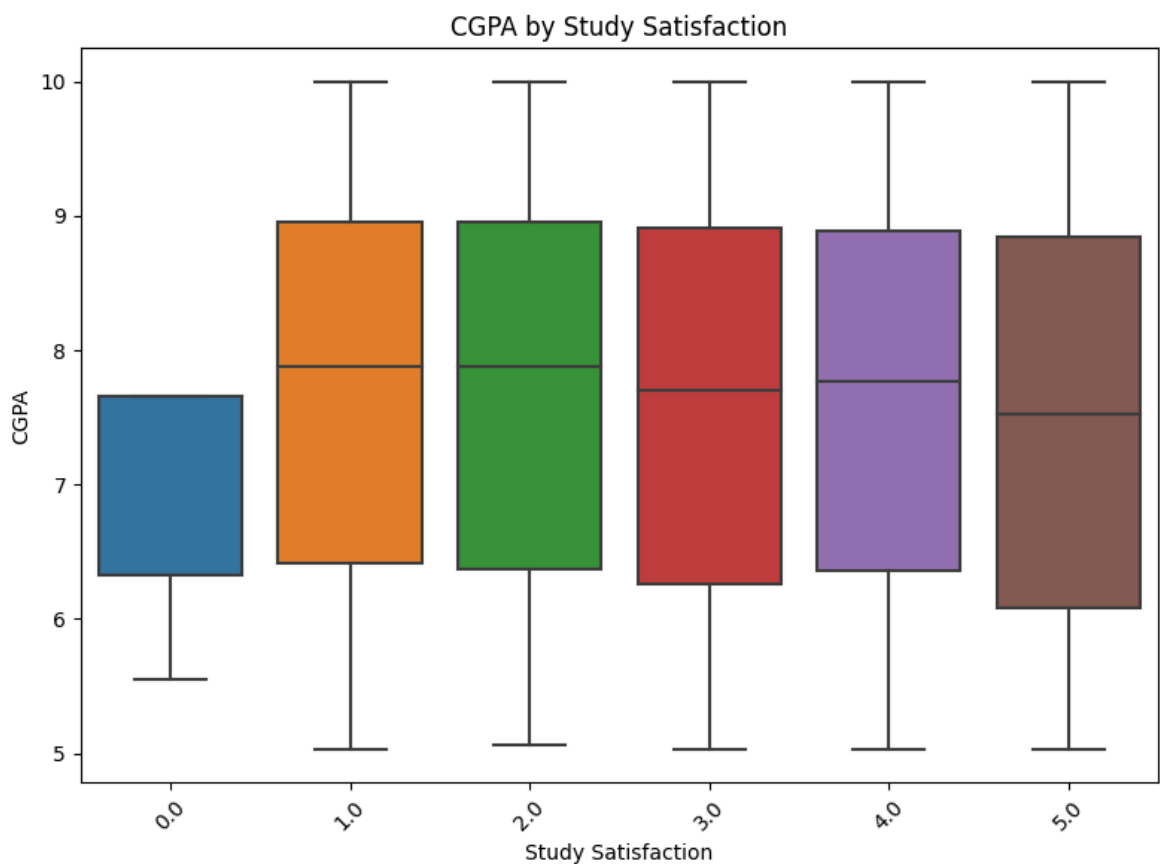
In [85]: `df[df['CGPA'] > gpalimit]`

Out[85]:

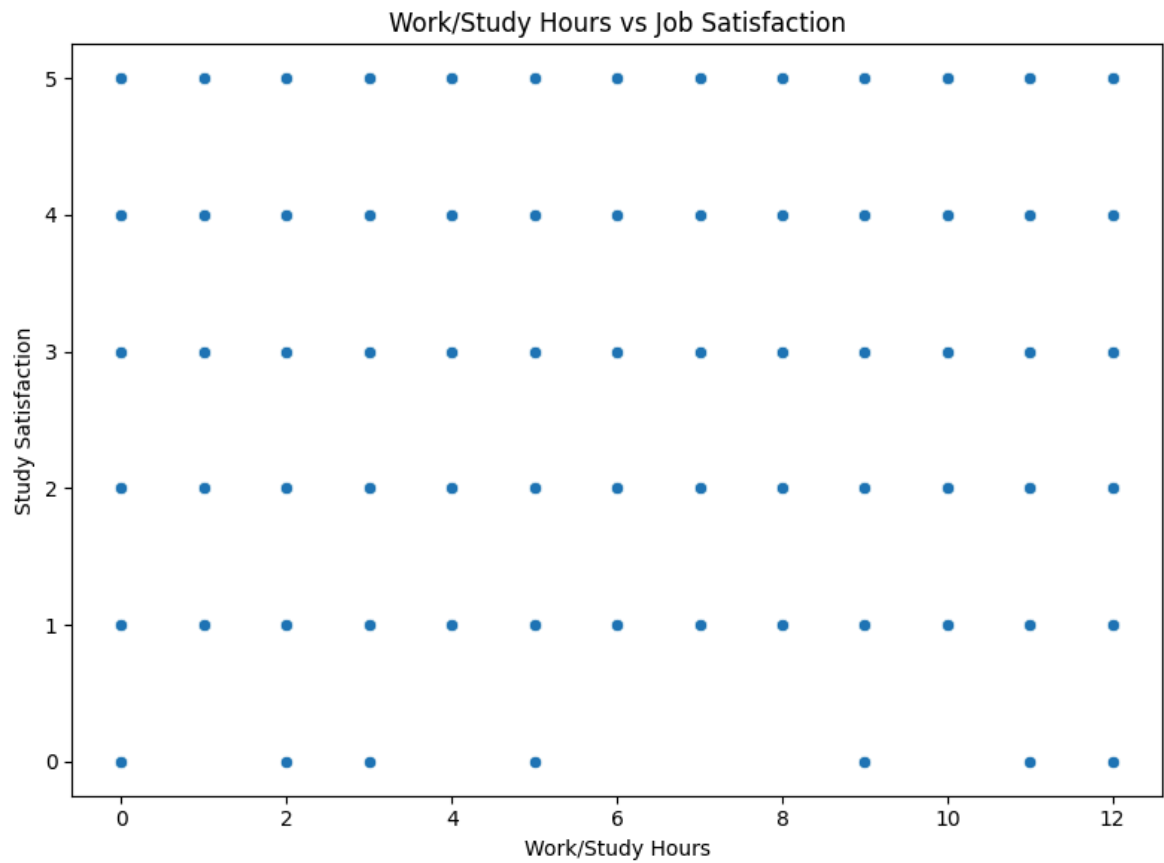
Gender Age City Profession Academic Pressure Work Pressure CGPA Study Satisfaction Job Satisfaction



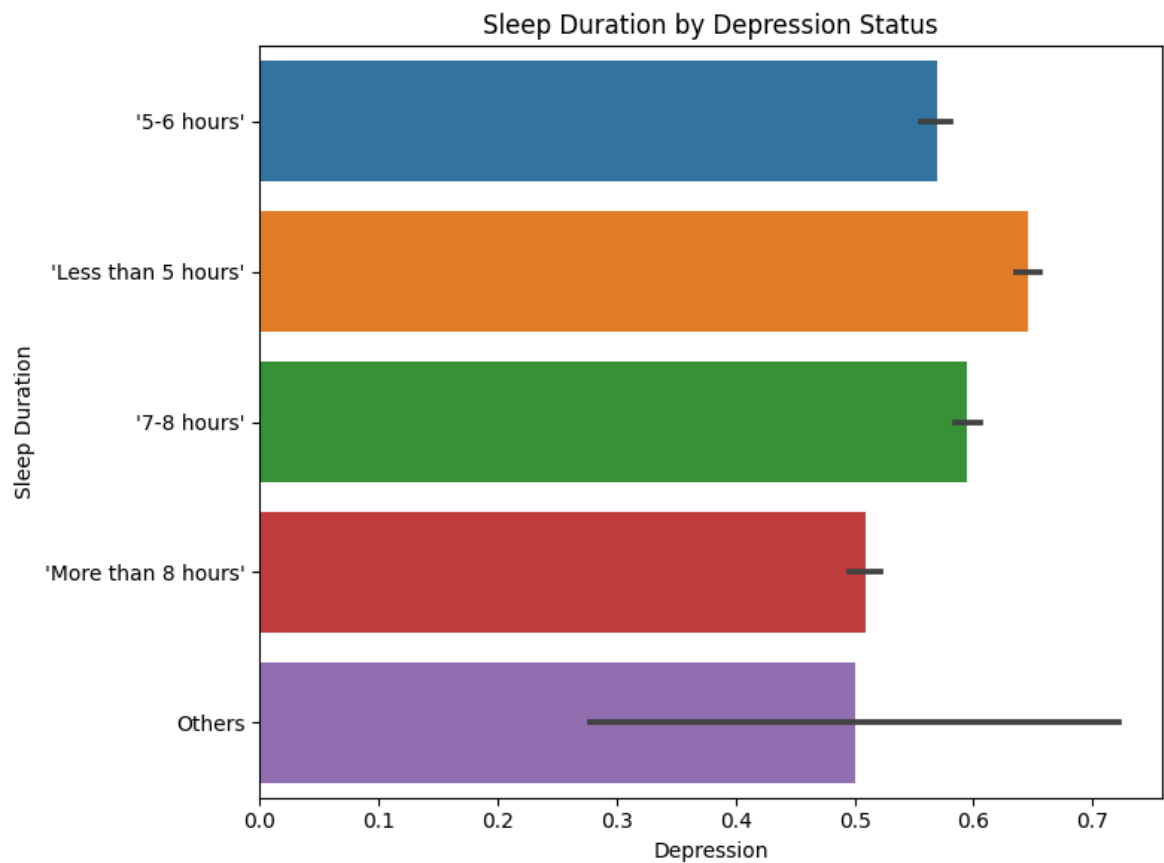
```
In [86]: plt.figure(figsize=(8, 6))
sns.boxplot(data=df, x='Study Satisfaction', y='CGPA')
plt.title('CGPA by Study Satisfaction')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



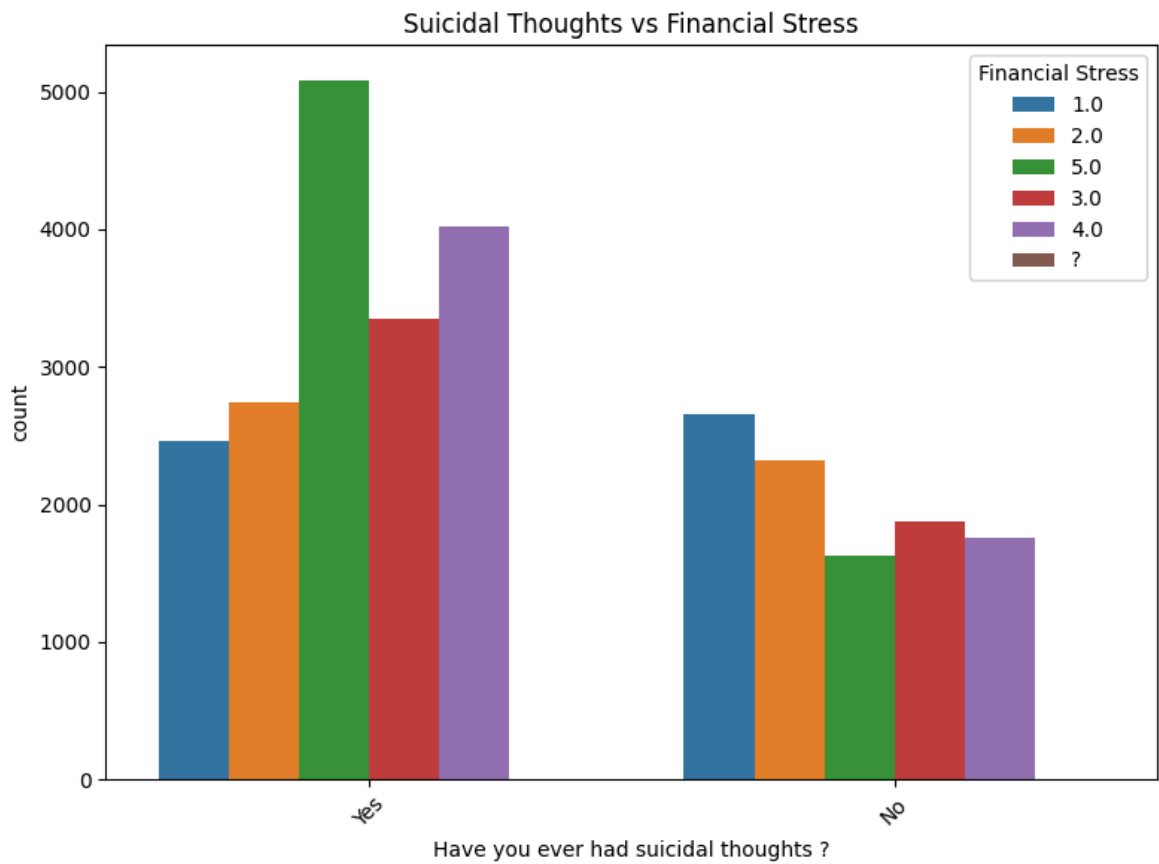
```
In [87]: plt.figure(figsize=(8, 6))
sns.scatterplot(data=df, x='Work/Study Hours', y='Study Satisfaction')
plt.title('Work/Study Hours vs Job Satisfaction')
plt.tight_layout()
plt.show()
```



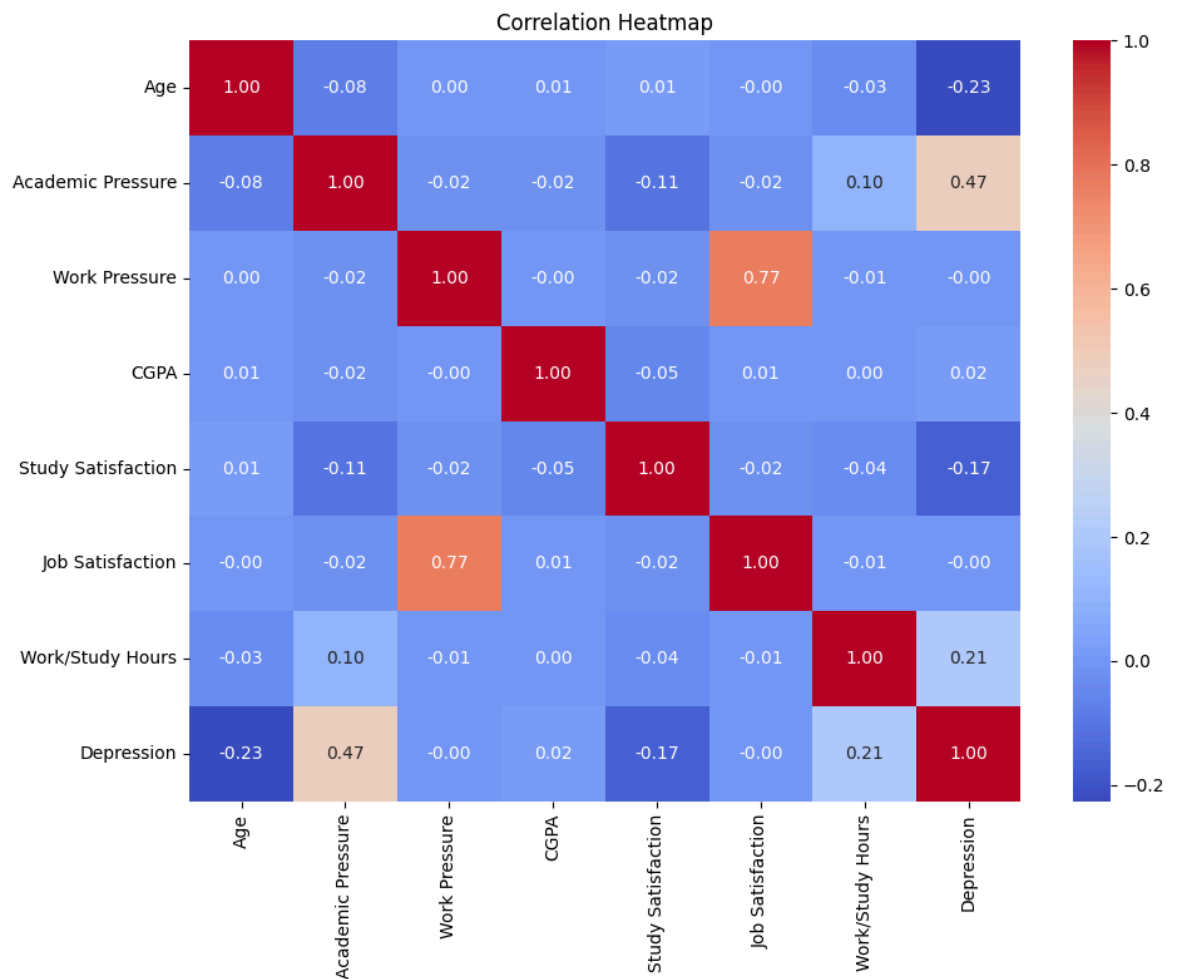
```
In [88]: plt.figure(figsize=(8, 6))
sns.barplot(data=df, x='Depression', y='Sleep Duration')
plt.title('Sleep Duration by Depression Status')
plt.tight_layout()
plt.show()
```



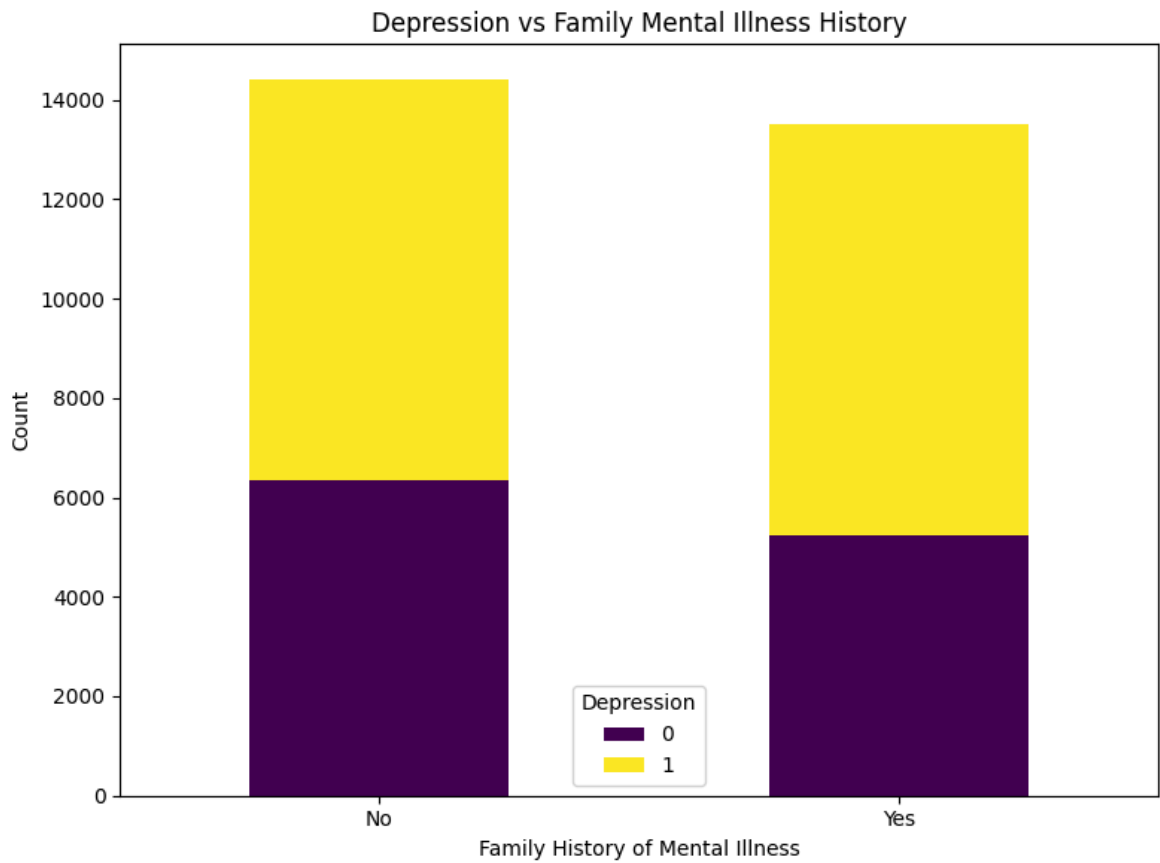
```
In [89]: plt.figure(figsize=(8, 6))
sns.countplot(data=df, x='Have you ever had suicidal thoughts ?', hue='Financial Stress')
plt.title('Suicidal Thoughts vs Financial Stress')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



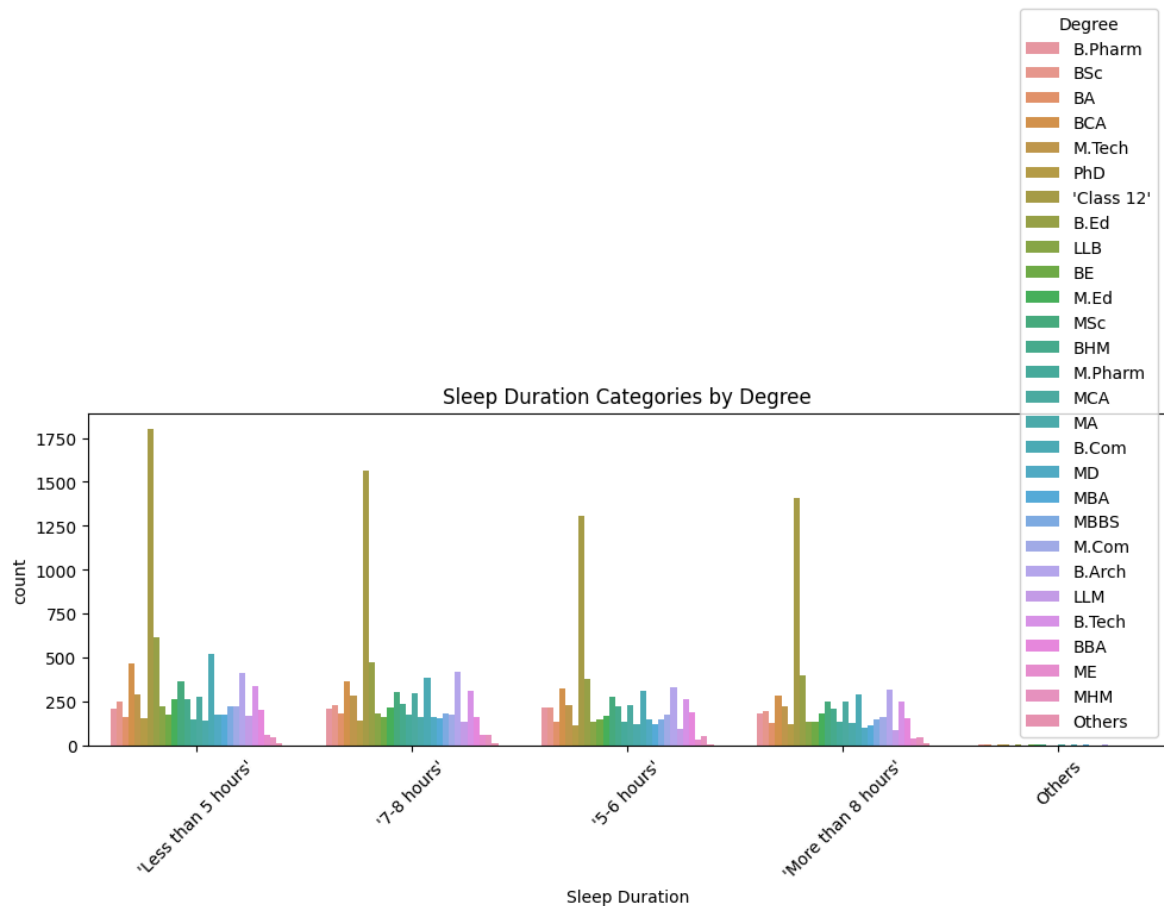
```
In [90]: plt.figure(figsize=(10, 8))
# Select only numeric columns
numeric_df = df.select_dtypes(include='number')
sns.heatmap(numeric_df.corr(), annot=True, cmap='coolwarm', fmt='.2f')
plt.title('Correlation Heatmap')
plt.tight_layout()
plt.show()
```



```
In [91]: crosstab = pd.crosstab(df['Family History of Mental Illness'], df['Depression'])
crosstab.plot(kind='bar', stacked=True, colormap='viridis', figsize=(8, 6))
plt.title('Depression vs Family Mental Illness History')
plt.ylabel('Count')
plt.xticks(rotation=0)
plt.tight_layout()
plt.show()
```



```
In [92]: plt.figure(figsize=(10, 6))
sns.countplot(data=df, x='Sleep Duration', hue='Degree', order=df['Sleep Duration'])
plt.title('Sleep Duration Categories by Degree')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



```
In [93]: df['Degree'].unique()
```

```
Out[93]: array(['B.Pharm', 'BSc', 'BA', 'BCA', 'M.Tech', 'PhD', "'Class 12'",
               'B.Ed', 'LLB', 'BE', 'M.Ed', 'MSc', 'BHM', 'M.Pharm', 'MCA', 'MA',
               'B.Com', 'MD', 'MBA', 'MBBS', 'M.Com', 'B.Arch', 'LLM', 'B.Tech',
               'BBA', 'ME', 'MHM', 'Others'], dtype=object)
```

```
In [94]: degree_map = {
    'B.Pharm': 'Graduate',
    'BSc': 'Graduate',
    'BA': 'Graduate',
    'BCA': 'Graduate',
    'B.Ed': 'Graduate',
    'LLB': 'Graduate',
    'BE': 'Graduate',
    'BHM': 'Graduate',
    'B.Com': 'Graduate',
    'B.Arch': 'Graduate',
    'B.Tech': 'Graduate',
    'BBA': 'Graduate',
    'ME': 'Graduate', # You can reclassify if context implies 'Master of Engine
    "'Class 12'": 'Pre-University',

    'M.Tech': 'Postgraduate',
    'M.Ed': 'Postgraduate',
    'MSc': 'Postgraduate',
    'M.Pharm': 'Postgraduate',
    'MCA': 'Postgraduate',
    'MA': 'Postgraduate',
    'MBA': 'Postgraduate',
    'M.Com': 'Postgraduate',
    'LLM': 'Postgraduate',
```



```

'MHM': 'Postgraduate',

'PhD': 'Doctorate',
'MD': 'Doctorate',
'MBBS': 'Postgraduate', # Or 'Medical Graduate' if separating doctors
}

# Apply to your DataFrame
df['Degree_Level'] = df['Degree'].map(degree_map)

```

In [95]: df

Out[95]:

	Gender	Age	City	Profession	Academic Pressure	Work Pressure	CGPA	Stud Satisfactio
0	Male	33.0	Visakhapatnam	Student	5.0	0.0	8.97	2.
1	Female	24.0	Bangalore	Student	2.0	0.0	5.90	5.
2	Male	31.0	Srinagar	Student	3.0	0.0	7.03	5.
3	Female	28.0	Varanasi	Student	3.0	0.0	5.59	2.
4	Female	25.0	Jaipur	Student	4.0	0.0	8.13	3.
...
27896	Female	27.0	Surat	Student	5.0	0.0	5.75	5.
27897	Male	27.0	Ludhiana	Student	2.0	0.0	9.40	3.
27898	Male	31.0	Faridabad	Student	3.0	0.0	6.61	4.
27899	Female	18.0	Ludhiana	Student	5.0	0.0	6.88	2.
27900	Male	27.0	Patna	Student	4.0	0.0	9.24	1.

27901 rows × 9 columns

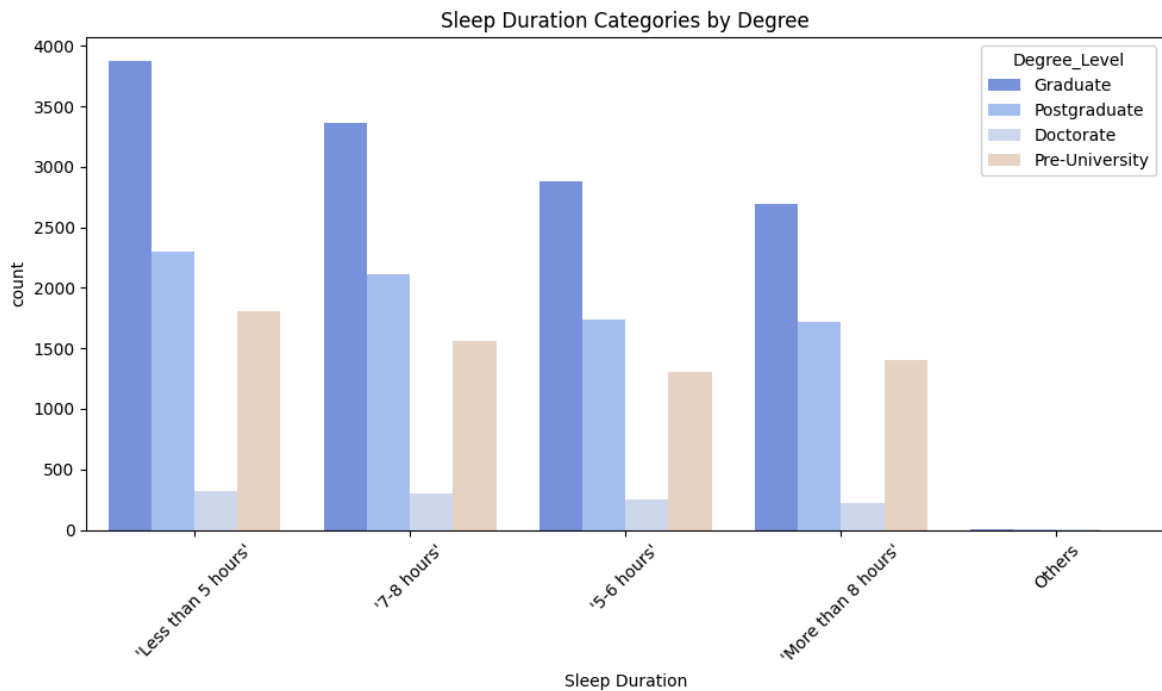


```

In [96]: plt.figure(figsize=(10, 6))
sns.countplot(data=df, x='Sleep Duration', hue='Degree_Level', order=df['Sleep D
plt.title('Sleep Duration Categories by Degree')

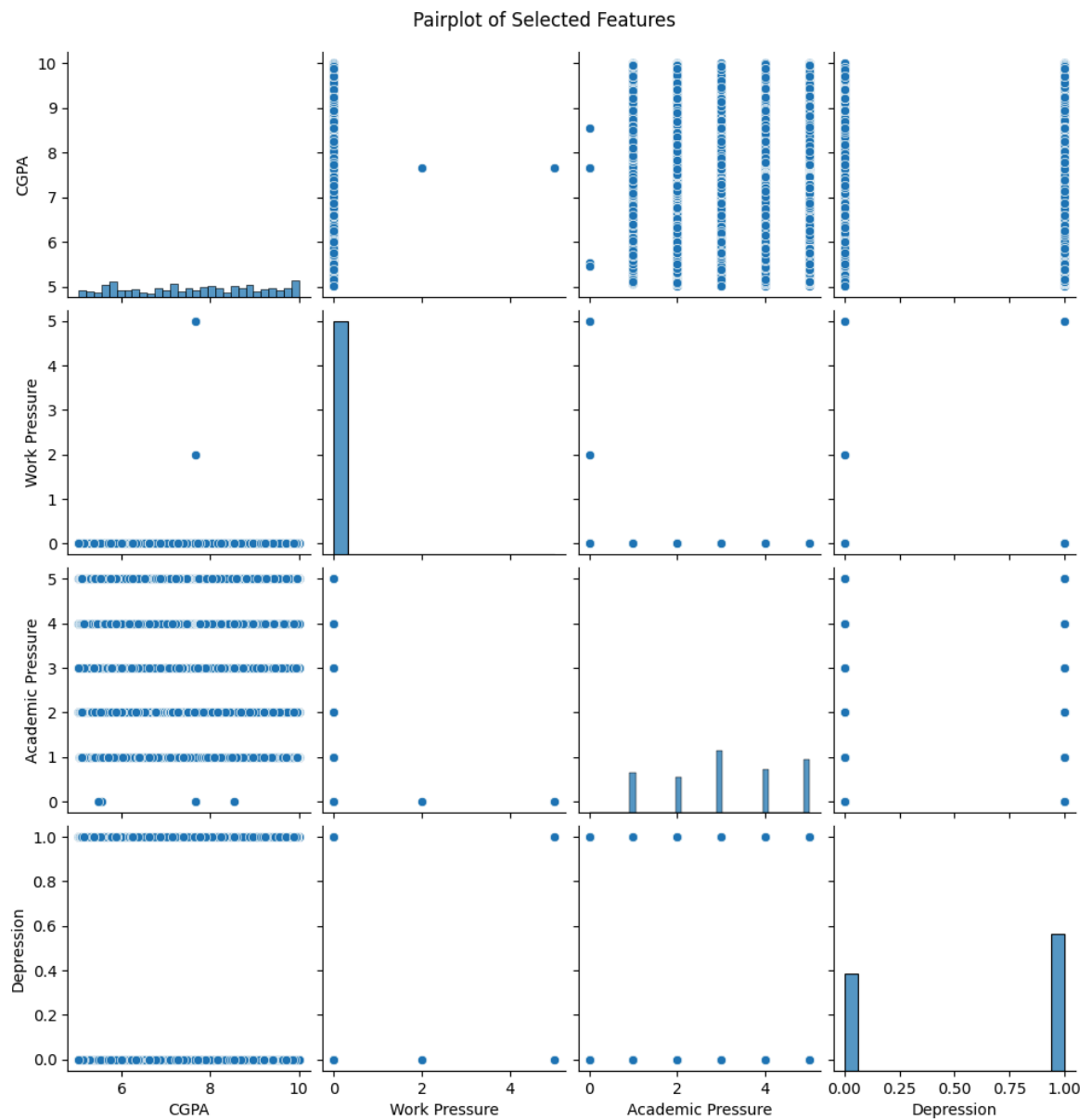
```

```
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

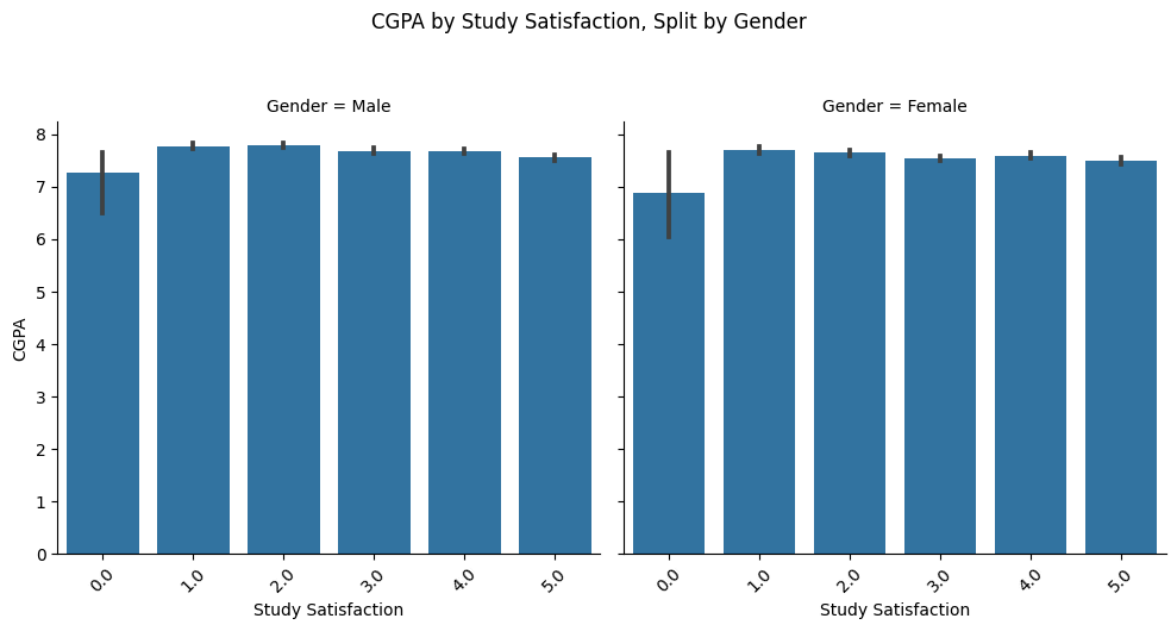


```
In [97]: sns.pairplot(df[['CGPA', 'Work Pressure', 'Academic Pressure', 'Sleep Duration'],
plt.suptitle('Pairplot of Selected Features', y=1.02)
plt.show()
```

```
c:\Users\lenovo\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn
_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be
removed in a future version. Convert inf values to NaN before operating instead.
with pd.option_context('mode.use_inf_as_na', True):
c:\Users\lenovo\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn
_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be
removed in a future version. Convert inf values to NaN before operating instead.
with pd.option_context('mode.use_inf_as_na', True):
c:\Users\lenovo\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn
_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be
removed in a future version. Convert inf values to NaN before operating instead.
with pd.option_context('mode.use_inf_as_na', True):
c:\Users\lenovo\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn
_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be
removed in a future version. Convert inf values to NaN before operating instead.
with pd.option_context('mode.use_inf_as_na', True):
```



```
In [98]: g = sns.FacetGrid(df, col='Gender', height=5)
g.map(sns.barplot, 'Study Satisfaction', 'CGPA', order=sorted(df['Study Satisfac
g.set_xticklabels(rotation=45)
g.fig.suptitle('CGPA by Study Satisfaction, Split by Gender', y=1.05)
plt.tight_layout()
plt.show()
```



```
In [99]: import category_encoders as ce
```

```
In [100... degree_order_map = {
    'Class 12': 1,
    'BCA': 2,
    'BBA': 3,
    'BA': 4,
    'BSc': 5,
    'B.Com': 6,
    'BHM': 7,
    'B.Ed': 8,
    'LLB': 9,
    'B.Pharm': 10,
    'B.Tech': 11,
    'BE': 12,
    'B.Arch': 13,
    'MCA': 14,
    'MBA': 15,
    'MA': 16,
    'MSc': 17,
    'M.Com': 18,
    'M.Ed': 19,
    'LLM': 20,
    'M.Pharm': 21,
    'ME': 22,
    'M.Tech': 23,
    'MHM': 24,
    'MBBS': 25,
    'MD': 26,
    'PhD': 27
}
```

```
In [101... x = df.drop(columns=['Depression'])
y = df['Depression']
```

```
In [102... from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
```

```
In [103... df.isna().sum()
```

```
Out[103... 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  0
           Family History of Mental Illness  0
           Depression    0
           age_groups    1587
           Degree_Level  35
           dtype: int64
```

```
In [104... df[df['age_groups'].isna()]
```

Out[104...

	Gender	Age	City	Profession	Academic Pressure	Work Pressure	CGPA	Stud Satisfactio
24	Male	18.0	Bangalore	Student	4.0	0.0	7.10	3.
30	Male	18.0	Surat	Student	4.0	0.0	6.70	5.
31	Male	18.0	Visakhapatnam	Student	2.0	0.0	6.21	3.
52	Female	18.0	Vadodara	Student	3.0	0.0	5.70	5.
72	Male	18.0	Bhopal	Student	3.0	0.0	6.75	4.
...
27823	Male	18.0	Ahmedabad	Student	2.0	0.0	8.04	4.
27858	Female	18.0	Ludhiana	Student	3.0	0.0	8.24	4.
27867	Male	18.0	Ahmedabad	Student	3.0	0.0	7.88	4.
27884	Male	18.0	Meerut	Student	5.0	0.0	6.25	3.
27899	Female	18.0	Ludhiana	Student	5.0	0.0	6.88	2.

1587 rows × 19 columns



In [105...

```
df[df['Age'] == 18]
```

Out[105...

	Gender	Age	City	Profession	Academic Pressure	Work Pressure	CGPA	Stud Satisfactio
24	Male	18.0	Bangalore	Student	4.0	0.0	7.10	3.
30	Male	18.0	Surat	Student	4.0	0.0	6.70	5.
31	Male	18.0	Visakhapatnam	Student	2.0	0.0	6.21	3.
52	Female	18.0	Vadodara	Student	3.0	0.0	5.70	5.
72	Male	18.0	Bhopal	Student	3.0	0.0	6.75	4.
...
27823	Male	18.0	Ahmedabad	Student	2.0	0.0	8.04	4.
27858	Female	18.0	Ludhiana	Student	3.0	0.0	8.24	4.
27867	Male	18.0	Ahmedabad	Student	3.0	0.0	7.88	4.
27884	Male	18.0	Meerut	Student	5.0	0.0	6.25	3.
27899	Female	18.0	Ludhiana	Student	5.0	0.0	6.88	2.

1587 rows × 19 columns



In [106...

```
df['age_groups'].fillna('teen', inplace=True)
```

C:\Users\lenovo\AppData\Local\Temp\ipykernel_576\1570449849.py:1: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment using an inplace method. The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value) instead, to perform the operation inplace on the original object.

```
df['age_groups'].fillna('teen', inplace=True)
```

```
In [107... df['Degree_Level'].unique()
```

```
Out[107... array(['Graduate', 'Postgraduate', 'Doctorate', 'Pre-University', nan],  
      dtype=object)
```

```
In [108... df[df['Degree_Level'].isna()]
```


Out[108...

	Gender	Age	City	Profession	Academic Pressure	Work Pressure	CGPA	Stud Satisfactio
624	Male	28.0	Varanasi	Student	4.0	0.0	5.66	4.
842	Male	28.0	Visakhapatnam	Student	1.0	0.0	7.94	4.
1897	Female	22.0	Srinagar	Student	1.0	0.0	8.04	4.
2609	Female	20.0	Bhopal	Student	5.0	0.0	5.58	2.
2928	Male	27.0	Kolkata	Student	5.0	0.0	8.04	1.
3189	Male	20.0	Lucknow	Student	2.0	0.0	9.36	3.
3672	Male	18.0	Nashik	Student	4.0	0.0	7.47	3.
4127	Male	19.0	Kolkata	Student	3.0	0.0	7.53	1.
4427	Female	22.0	Vasai-Virar	Student	4.0	0.0	6.03	3.
4516	Female	23.0	Kolkata	Student	3.0	0.0	5.32	4.
7003	Male	33.0	Kalyan	Student	5.0	0.0	8.61	1.
7018	Female	28.0	Mumbai	Student	2.0	0.0	9.88	5.
7480	Female	30.0	Ghaziabad	Student	3.0	0.0	6.56	1.
8017	Female	24.0	Srinagar	Student	3.0	0.0	6.37	1.
8911	Male	26.0	Nagpur	Student	1.0	0.0	6.47	5.
8936	Female	28.0	Patna	Student	5.0	0.0	7.24	1.

	Gender	Age	City	Profession	Academic Pressure	Work Pressure	CGPA	Stud Satisfactio
9256	Male	32.0	Kolkata	Student	3.0	0.0	9.96	2.
11962	Male	27.0	Ahmedabad	Student	5.0	0.0	9.95	2.
12168	Male	23.0	Mumbai	Student	1.0	0.0	6.84	5.
13435	Male	31.0	Kanpur	Student	2.0	0.0	8.95	1.
13515	Male	27.0	Ahmedabad	Student	3.0	0.0	7.27	4.
13588	Male	24.0	Rajkot	Student	3.0	0.0	8.89	3.
15864	Male	29.0	Faridabad	Student	2.0	0.0	9.24	3.
16085	Male	20.0	Vadodara	Student	3.0	0.0	8.24	3.
16604	Female	28.0	Kolkata	Student	3.0	0.0	8.95	5.
17955	Male	29.0	Rajkot	Student	5.0	0.0	5.88	3.
18027	Female	25.0	Patna	Student	4.0	0.0	8.25	4.
19837	Male	30.0	Kolkata	Student	3.0	0.0	5.61	3.
21653	Female	29.0	Hyderabad	Student	3.0	0.0	7.48	2.
21924	Female	23.0	Patna	Student	3.0	0.0	5.58	5.
23118	Female	32.0	Pune	Student	1.0	0.0	5.74	2.
23653	Female	34.0	Srinagar	Student	2.0	0.0	9.89	4.

	Gender	Age	City	Profession	Academic Pressure	Work Pressure	CGPA	Stud Satisfactio
24419	Female	30.0	Nagpur	Student	3.0	0.0	9.49	2.
25860	Female	33.0	Bhopal	Student	1.0	0.0	7.85	3.
26742	Male	32.0	Mumbai	Student	2.0	0.0	5.45	3.

every degree named others, the degree level is null. the number of rows that have these values is 35, so it is very small comared to the number of rows in this dataset so we can drop them

In [109...

```
df[df['Degree'] == 'Others']
```

Out[109...

	Gender	Age	City	Profession	Academic Pressure	Work Pressure	CGPA	Stud Satisfactio
624	Male	28.0	Varanasi	Student	4.0	0.0	5.66	4.
842	Male	28.0	Visakhapatnam	Student	1.0	0.0	7.94	4.
1897	Female	22.0	Srinagar	Student	1.0	0.0	8.04	4.
2609	Female	20.0	Bhopal	Student	5.0	0.0	5.58	2.
2928	Male	27.0	Kolkata	Student	5.0	0.0	8.04	1.
3189	Male	20.0	Lucknow	Student	2.0	0.0	9.36	3.
3672	Male	18.0	Nashik	Student	4.0	0.0	7.47	3.
4127	Male	19.0	Kolkata	Student	3.0	0.0	7.53	1.
4427	Female	22.0	Vasai-Virar	Student	4.0	0.0	6.03	3.
4516	Female	23.0	Kolkata	Student	3.0	0.0	5.32	4.
7003	Male	33.0	Kalyan	Student	5.0	0.0	8.61	1.
7018	Female	28.0	Mumbai	Student	2.0	0.0	9.88	5.
7480	Female	30.0	Ghaziabad	Student	3.0	0.0	6.56	1.
8017	Female	24.0	Srinagar	Student	3.0	0.0	6.37	1.
8911	Male	26.0	Nagpur	Student	1.0	0.0	6.47	5.
8936	Female	28.0	Patna	Student	5.0	0.0	7.24	1.

	Gender	Age	City	Profession	Academic Pressure	Work Pressure	CGPA	Stud Satisfactio
9256	Male	32.0	Kolkata	Student	3.0	0.0	9.96	2.
11962	Male	27.0	Ahmedabad	Student	5.0	0.0	9.95	2.
12168	Male	23.0	Mumbai	Student	1.0	0.0	6.84	5.
13435	Male	31.0	Kanpur	Student	2.0	0.0	8.95	1.
13515	Male	27.0	Ahmedabad	Student	3.0	0.0	7.27	4.
13588	Male	24.0	Rajkot	Student	3.0	0.0	8.89	3.
15864	Male	29.0	Faridabad	Student	2.0	0.0	9.24	3.
16085	Male	20.0	Vadodara	Student	3.0	0.0	8.24	3.
16604	Female	28.0	Kolkata	Student	3.0	0.0	8.95	5.
17955	Male	29.0	Rajkot	Student	5.0	0.0	5.88	3.
18027	Female	25.0	Patna	Student	4.0	0.0	8.25	4.
19837	Male	30.0	Kolkata	Student	3.0	0.0	5.61	3.
21653	Female	29.0	Hyderabad	Student	3.0	0.0	7.48	2.
21924	Female	23.0	Patna	Student	3.0	0.0	5.58	5.
23118	Female	32.0	Pune	Student	1.0	0.0	5.74	2.
23653	Female	34.0	Srinagar	Student	2.0	0.0	9.89	4.

	Gender	Age	City	Profession	Academic Pressure	Work Pressure	CGPA	Stud Satisfactio
24419	Female	30.0	Nagpur	Student	3.0	0.0	9.49	2.
25860	Female	33.0	Bhopal	Student	1.0	0.0	7.85	3.
26742	Male	32.0	Mumbai	Student	2.0	0.0	5.45	3.


In [110...] `df = df[df['Degree_Level'].notna()]`

In [111...] `df.isna().sum()`

Out[111...] 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 0
 Family History of Mental Illness 0
 Depression 0
 age_groups 0
 Degree_Level 0
 dtype: int64

In [112...] `df[df['Degree'] == 'Others']`

Out[112...]

	Gender	Age	City	Profession	Academic Pressure	Work Pressure	CGPA	Study Satisfaction	Job Satisfaction
									

In [113...] `df.info()`

```
<class 'pandas.core.frame.DataFrame'>
Index: 27866 entries, 0 to 27900
Data columns (total 19 columns):
#   Column                                     Non-Null Count  Dtype
---  -
0   Gender                                     27866 non-null  object
1   Age                                       27866 non-null  float64
2   City                                     27866 non-null  object
3   Profession                               27866 non-null  object
4   Academic Pressure                       27866 non-null  float64
5   Work Pressure                           27866 non-null  float64
6   CGPA                                    27866 non-null  float64
7   Study Satisfaction                     27866 non-null  float64
8   Job Satisfaction                       27866 non-null  float64
9   Sleep Duration                         27866 non-null  object
10  Dietary Habits                         27866 non-null  object
11  Degree                                 27866 non-null  object
12  Have you ever had suicidal thoughts ?  27866 non-null  object
13  Work/Study Hours                       27866 non-null  float64
14  Financial Stress                       27866 non-null  object
15  Family History of Mental Illness       27866 non-null  object
16  Depression                             27866 non-null  int64
17  age_groups                             27866 non-null  category
18  Degree_Level                           27866 non-null  object
dtypes: category(1), float64(7), int64(1), object(10)
memory usage: 4.1+ MB
```

```
In [114... dummies = pd.get_dummies(df['Gender'], dtype=int)
```

```
In [115... dummies
```

```
Out[115...
      Female  Male
0          0     1
1          1     0
2          0     1
3          1     0
4          1     0
...        ...   ...
27896       1     0
27897       0     1
27898       0     1
27899       1     0
27900       0     1
```

27866 rows × 2 columns

```
In [116... data = pd.concat([df, dummies], axis=1)
```

```
In [117... data
```

Out[117...

	Gender	Age	City	Profession	Academic Pressure	Work Pressure	CGPA	Stud Satisfactio
0	Male	33.0	Visakhapatnam	Student	5.0	0.0	8.97	2.
1	Female	24.0	Bangalore	Student	2.0	0.0	5.90	5.
2	Male	31.0	Srinagar	Student	3.0	0.0	7.03	5.
3	Female	28.0	Varanasi	Student	3.0	0.0	5.59	2.
4	Female	25.0	Jaipur	Student	4.0	0.0	8.13	3.
...
27896	Female	27.0	Surat	Student	5.0	0.0	5.75	5.
27897	Male	27.0	Ludhiana	Student	2.0	0.0	9.40	3.
27898	Male	31.0	Faridabad	Student	3.0	0.0	6.61	4.
27899	Female	18.0	Ludhiana	Student	5.0	0.0	6.88	2.
27900	Male	27.0	Patna	Student	4.0	0.0	9.24	1.

27866 rows × 21 columns



In [118... data.drop(columns=['Gender'], inplace=True)

In [119... data.isna().sum()


```
Out[119...] 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 0
Family History of Mental Illness 0
Depression 0
age_groups 0
Degree_Level 0
Female 0
Male 0
dtype: int64
```

```
In [120...] data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 27866 entries, 0 to 27900
Data columns (total 20 columns):
#   Column                                     Non-Null Count  Dtype
---  -
0   Age                                     27866 non-null  float64
1   City                                    27866 non-null  object
2   Profession                              27866 non-null  object
3   Academic Pressure                       27866 non-null  float64
4   Work Pressure                           27866 non-null  float64
5   CGPA                                    27866 non-null  float64
6   Study Satisfaction                      27866 non-null  float64
7   Job Satisfaction                        27866 non-null  float64
8   Sleep Duration                          27866 non-null  object
9   Dietary Habits                          27866 non-null  object
10  Degree                                  27866 non-null  object
11  Have you ever had suicidal thoughts ?  27866 non-null  object
12  Work/Study Hours                        27866 non-null  float64
13  Financial Stress                        27866 non-null  object
14  Family History of Mental Illness        27866 non-null  object
15  Depression                              27866 non-null  int64
16  age_groups                              27866 non-null  category
17  Degree_Level                            27866 non-null  object
18  Female                                  27866 non-null  int32
19  Male                                    27866 non-null  int32
dtypes: category(1), float64(7), int32(2), int64(1), object(9)
memory usage: 4.1+ MB
```

```
In [121...] data['City'].nunique()
```

```
Out[121...] 47
```

```
In [122...] import category_encoders as ce
binary_encoder = ce.BinaryEncoder(cols= ["City"], return_df= True)
```

```
data = binary_encoder.fit_transform(data)
data
```

Out[122...

	Age	City_0	City_1	City_2	City_3	City_4	City_5	Profession	Academic Pressure	Wk Pressu
0	33.0	0	0	0	0	0	1	Student	5.0	
1	24.0	0	0	0	0	1	0	Student	2.0	
2	31.0	0	0	0	0	1	1	Student	3.0	
3	28.0	0	0	0	1	0	0	Student	3.0	
4	25.0	0	0	0	1	0	1	Student	4.0	
...
27896	27.0	0	1	0	0	1	1	Student	5.0	
27897	27.0	0	1	0	1	0	0	Student	2.0	
27898	31.0	0	1	1	1	0	1	Student	3.0	
27899	18.0	0	1	0	1	0	0	Student	5.0	
27900	27.0	0	1	1	1	0	0	Student	4.0	

27866 rows × 25 columns



In [123...

```
data.isna().sum()
```

```
Out[123... Age 0
City_0 0
City_1 0
City_2 0
City_3 0
City_4 0
City_5 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 0
Family History of Mental Illness 0
Depression 0
age_groups 0
Degree_Level 0
Female 0
Male 0
dtype: int64
```

```
In [124... data.info()
```

```

<class 'pandas.core.frame.DataFrame'>
Index: 27866 entries, 0 to 27900
Data columns (total 25 columns):
#   Column                                     Non-Null Count  Dtype
---  -
0   Age                                         27866 non-null  float64
1   City_0                                     27866 non-null  int64
2   City_1                                     27866 non-null  int64
3   City_2                                     27866 non-null  int64
4   City_3                                     27866 non-null  int64
5   City_4                                     27866 non-null  int64
6   City_5                                     27866 non-null  int64
7   Profession                                 27866 non-null  object
8   Academic Pressure                         27866 non-null  float64
9   Work Pressure                             27866 non-null  float64
10  CGPA                                       27866 non-null  float64
11  Study Satisfaction                        27866 non-null  float64
12  Job Satisfaction                         27866 non-null  float64
13  Sleep Duration                           27866 non-null  object
14  Dietary Habits                           27866 non-null  object
15  Degree                                    27866 non-null  object
16  Have you ever had suicidal thoughts ?    27866 non-null  object
17  Work/Study Hours                         27866 non-null  float64
18  Financial Stress                         27866 non-null  object
19  Family History of Mental Illness         27866 non-null  object
20  Depression                               27866 non-null  int64
21  age_groups                               27866 non-null  category
22  Degree_Level                             27866 non-null  object
23  Female                                   27866 non-null  int32
24  Male                                     27866 non-null  int32
dtypes: category(1), float64(7), int32(2), int64(7), object(8)
memory usage: 5.1+ MB

```

```
In [125... data['Profession'].unique()
```

```
Out[125... array(['Student', "'Civil Engineer'", 'Architect', "'UX/UI Designer'",
      "'Digital Marketer'", "'Content Writer'",
      "'Educational Consultant'", 'Teacher', 'Manager', 'Chef', 'Doctor',
      'Lawyer', 'Entrepreneur', 'Pharmacist'], dtype=object)
```

```
In [126... binary_encoder2 = ce.BinaryEncoder(cols= ["Profession"], return_df= True)
data = binary_encoder2.fit_transform(data)
```

```
In [127... data
```

Out[127...

	Age	City_0	City_1	City_2	City_3	City_4	City_5	Profession_0	Profession_1	F
0	33.0	0	0	0	0	0	1	0	0	
1	24.0	0	0	0	0	1	0	0	0	
2	31.0	0	0	0	0	1	1	0	0	
3	28.0	0	0	0	1	0	0	0	0	
4	25.0	0	0	0	1	0	1	0	0	
...
27896	27.0	0	1	0	0	1	1	0	0	
27897	27.0	0	1	0	1	0	0	0	0	
27898	31.0	0	1	1	1	0	1	0	0	
27899	18.0	0	1	0	1	0	0	0	0	
27900	27.0	0	1	1	1	0	0	0	0	

27866 rows × 28 columns



In [128...

```
data.isna().sum()
```

```
Out[128... Age 0
City_0 0
City_1 0
City_2 0
City_3 0
City_4 0
City_5 0
Profession_0 0
Profession_1 0
Profession_2 0
Profession_3 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 0
Family History of Mental Illness 0
Depression 0
age_groups 0
Degree_Level 0
Female 0
Male 0
dtype: int64
```

```
In [129... data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 27866 entries, 0 to 27900
Data columns (total 28 columns):
#   Column                                     Non-Null Count  Dtype
---  -
0   Age                                         27866 non-null  float64
1   City_0                                     27866 non-null  int64
2   City_1                                     27866 non-null  int64
3   City_2                                     27866 non-null  int64
4   City_3                                     27866 non-null  int64
5   City_4                                     27866 non-null  int64
6   City_5                                     27866 non-null  int64
7   Profession_0                             27866 non-null  int64
8   Profession_1                             27866 non-null  int64
9   Profession_2                             27866 non-null  int64
10  Profession_3                             27866 non-null  int64
11  Academic Pressure                         27866 non-null  float64
12  Work Pressure                             27866 non-null  float64
13  CGPA                                       27866 non-null  float64
14  Study Satisfaction                       27866 non-null  float64
15  Job Satisfaction                         27866 non-null  float64
16  Sleep Duration                           27866 non-null  object
17  Dietary Habits                           27866 non-null  object
18  Degree                                    27866 non-null  object
19  Have you ever had suicidal thoughts ?    27866 non-null  object
20  Work/Study Hours                         27866 non-null  float64
21  Financial Stress                         27866 non-null  object
22  Family History of Mental Illness         27866 non-null  object
23  Depression                               27866 non-null  int64
24  age_groups                               27866 non-null  category
25  Degree_Level                             27866 non-null  object
26  Female                                   27866 non-null  int32
27  Male                                      27866 non-null  int32
dtypes: category(1), float64(7), int32(2), int64(11), object(7)
memory usage: 5.8+ MB
```

```
In [130...] data['Sleep Duration'].unique()
```

```
Out[130...] array(['5-6 hours', 'Less than 5 hours', '7-8 hours',
      'More than 8 hours', 'Others'], dtype=object)
```

```
In [131...] data[data['Sleep Duration'] == 'Others']
```

Out[131...

	Age	City_0	City_1	City_2	City_3	City_4	City_5	Profession_0	Profession_1	F
1963	30.0	0	1	0	1	0	0	0	0	
3807	18.0	0	1	0	1	1	1	0	0	
3984	27.0	0	1	0	1	0	0	0	0	
4535	22.0	0	0	0	0	1	1	0	0	
5526	31.0	0	1	1	1	1	0	0	0	
5786	23.0	0	1	0	0	0	0	0	0	
6740	20.0	0	1	1	0	1	0	0	0	
7696	31.0	0	0	1	1	0	0	0	0	
7875	28.0	0	0	0	0	0	1	0	0	
12933	28.0	0	0	1	0	0	1	0	0	
14382	33.0	0	0	1	1	1	0	0	0	
14988	26.0	0	0	0	0	1	1	0	0	
15103	25.0	0	1	1	0	1	1	0	0	
15958	23.0	0	0	1	0	0	1	0	0	
18224	31.0	0	1	1	1	0	1	0	0	
20315	32.0	0	1	1	0	0	1	0	0	
22903	34.0	0	1	0	1	1	0	0	0	
25204	30.0	0	1	1	1	1	0	0	0	

18 rows × 28 columns



In [132...] `data = data[data['Sleep Duration'] != 'Others']`

In [133...] `dummies2 = pd.get_dummies(data['Sleep Duration'], dtype=int)`

In [134...] `data = pd.concat([data, dummies2], axis=1)`

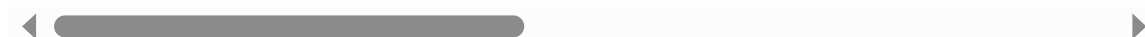
In [135...] `data.drop(columns=['Sleep Duration'], inplace=True)`

In [136...] `data`

Out[136...

	Age	City_0	City_1	City_2	City_3	City_4	City_5	Profession_0	Profession_1	F
0	33.0	0	0	0	0	0	1	0	0	
1	24.0	0	0	0	0	1	0	0	0	
2	31.0	0	0	0	0	1	1	0	0	
3	28.0	0	0	0	1	0	0	0	0	
4	25.0	0	0	0	1	0	1	0	0	
...
27896	27.0	0	1	0	0	1	1	0	0	
27897	27.0	0	1	0	1	0	0	0	0	
27898	31.0	0	1	1	1	0	1	0	0	
27899	18.0	0	1	0	1	0	0	0	0	
27900	27.0	0	1	1	1	0	0	0	0	

27848 rows × 31 columns



In [137...] `object_data = data.select_dtypes(include='object')`

In [138...] `object_data.columns`

Out[138...] `Index(['Dietary Habits', 'Degree', 'Have you ever had suicidal thoughts ?',
'Financial Stress', 'Family History of Mental Illness', 'Degree_Level'],
dtype='object')`

In [139...] `data['Dietary Habits'].unique()`

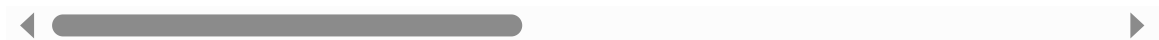
Out[139...] `array(['Healthy', 'Moderate', 'Unhealthy', 'Others'], dtype=object)`

In [140...] `data[data['Dietary Habits'] == 'Others']`

Out[140...

	Age	City_0	City_1	City_2	City_3	City_4	City_5	Profession_0	Profession_1	F
5684	23.0	0	1	0	1	1	0	0	0	
6892	18.0	0	0	1	0	1	1	0	0	
8263	21.0	0	0	0	1	0	1	0	0	
8456	32.0	0	0	1	1	1	0	0	0	
11746	34.0	0	1	0	1	1	1	0	0	
12763	23.0	0	0	0	0	1	0	0	0	
13758	34.0	0	0	1	1	1	0	0	0	
14504	26.0	0	1	0	0	1	0	0	0	
15087	27.0	0	1	0	0	0	1	0	0	
22515	30.0	0	1	1	0	0	0	0	0	
23506	24.0	0	0	1	1	1	1	0	0	
26061	25.0	0	1	1	0	1	0	0	0	

12 rows × 31 columns



In [141... `data = data[data['Dietary Habits'] != 'Others']`

In [142... `data2 = data.copy()`

In [143... `from sklearn.preprocessing import OrdinalEncoder`
`encoder = OrdinalEncoder(categories=[['Unhealthy', 'Moderate', 'Healthy']])`
`data2['habits_encoded'] = encoder.fit_transform(data2[['Dietary Habits']])`

In [144... `data2`

Out[144...

	Age	City_0	City_1	City_2	City_3	City_4	City_5	Profession_0	Profession_1	F
0	33.0	0	0	0	0	0	1	0	0	
1	24.0	0	0	0	0	1	0	0	0	
2	31.0	0	0	0	0	1	1	0	0	
3	28.0	0	0	0	1	0	0	0	0	
4	25.0	0	0	0	1	0	1	0	0	
...
27896	27.0	0	1	0	0	1	1	0	0	
27897	27.0	0	1	0	1	0	0	0	0	
27898	31.0	0	1	1	1	0	1	0	0	
27899	18.0	0	1	0	1	0	0	0	0	
27900	27.0	0	1	1	1	0	0	0	0	

27836 rows × 32 columns



In [145...

data2.columns

Out[145...

```
Index(['Age', 'City_0', 'City_1', 'City_2', 'City_3', 'City_4', 'City_5',
      'Profession_0', 'Profession_1', 'Profession_2', 'Profession_3',
      'Academic Pressure', 'Work Pressure', 'CGPA', 'Study Satisfaction',
      'Job Satisfaction', 'Dietary Habits', 'Degree',
      'Have you ever had suicidal thoughts ?', 'Work/Study Hours',
      'Financial Stress', 'Family History of Mental Illness', 'Depression',
      'age_groups', 'Degree_Level', 'Female', 'Male', '5-6 hours',
      '7-8 hours', 'Less than 5 hours', 'More than 8 hours',
      'habits_encoded'],
      dtype='object')
```

In [146...

```
data2.drop(columns=['Dietary Habits'], inplace=True)
```

In [147...

```
data2.isna().sum()
```

```

Out[147... Age                                0
           City_0                             0
           City_1                             0
           City_2                             0
           City_3                             0
           City_4                             0
           City_5                             0
           Profession_0                       0
           Profession_1                       0
           Profession_2                       0
           Profession_3                       0
           Academic Pressure                  0
           Work Pressure                      0
           CGPA                              0
           Study Satisfaction                 0
           Job Satisfaction                   0
           Degree                             0
           Have you ever had suicidal thoughts ? 0
           Work/Study Hours                  0
           Financial Stress                   0
           Family History of Mental Illness    0
           Depression                         0
           age_groups                         0
           Degree_Level                       0
           Female                             0
           Male                               0
           '5-6 hours'                        0
           '7-8 hours'                        0
           'Less than 5 hours'                0
           'More than 8 hours'                0
           habits_encoded                     0
           dtype: int64

```

```
In [148... data2['Degree'].unique()
```

```

Out[148... array(['B.Pharm', 'BSc', 'BA', 'BCA', 'M.Tech', 'PhD', "'Class 12'",
           'B.Ed', 'LLB', 'BE', 'M.Ed', 'MSc', 'BHM', 'M.Pharm', 'MCA', 'MA',
           'B.Com', 'MD', 'MBA', 'MBBS', 'M.Com', 'B.Arch', 'LLM', 'B.Tech',
           'BBA', 'ME', 'MHM'], dtype=object)

```

```
In [149... data2['Degree_Level'].unique()
```

```

Out[149... array(['Graduate', 'Postgraduate', 'Doctorate', 'Pre-University'],
           dtype=object)

```

```

In [150... encoder2 = OrdinalEncoder(categories=[["'Class 12'", 'BCA', 'BBA', 'BA', 'BSc',
           'B.Tech', 'BE', 'B.Arch', 'MCA', 'MBA', 'MA', 'MSc', 'M.Com', 'M.Ed', 'LLM',
           'M.Pharm', 'ME', 'M.Tech', 'MHM', 'MBBS', 'MD', 'PhD']])

```

```
In [151... data2['degree_encoded'] = encoder2.fit_transform(data2[['Degree']])
```

```
In [152... data2
```

Out[152...

	Age	City_0	City_1	City_2	City_3	City_4	City_5	Profession_0	Profession_1	F
0	33.0	0	0	0	0	0	1	0	0	
1	24.0	0	0	0	0	1	0	0	0	
2	31.0	0	0	0	0	1	1	0	0	
3	28.0	0	0	0	1	0	0	0	0	
4	25.0	0	0	0	1	0	1	0	0	
...	
27896	27.0	0	1	0	0	1	1	0	0	
27897	27.0	0	1	0	1	0	0	0	0	
27898	31.0	0	1	1	1	0	1	0	0	
27899	18.0	0	1	0	1	0	0	0	0	
27900	27.0	0	1	1	1	0	0	0	0	

27836 rows × 32 columns



```
In [153... data2.drop(columns=['Degree'], inplace=True)
```

```
In [154... encoder3 = OrdinalEncoder(categories=[['Pre-University', 'Graduate', 'Postgradua
data2['degree_level_encoded'] = encoder3.fit_transform(data2[['Degree_Level']])
```

```
In [155... data2.drop(columns=['Degree_Level'], inplace=True)
```

```
In [156... # Index(['Dietary Habits', 'Degree', 'Have you ever had suicidal thoughts ?',
#         'Financial Stress', 'Family History of Mental Illness', 'Degree_Level']
#         dtype='object')
```

```
In [157... data2['Have you ever had suicidal thoughts ?'].unique()
```

```
Out[157... array(['Yes', 'No'], dtype=object)
```

```
In [158... data2['Family History of Mental Illness'].unique()
```

```
Out[158... array(['No', 'Yes'], dtype=object)
```

```
In [159... dummies3 = pd.get_dummies(data2['Have you ever had suicidal thoughts ?'], dtype=
```

```
In [160... dummies4 = pd.get_dummies(data2['Family History of Mental Illness'], dtype=int)
```

```
In [161... data2 = pd.concat([data2, dummies3], axis=1)
```

```
In [162... data2 = pd.concat([data2, dummies4], axis=1)
```

In [163... `data2.drop(columns=['Have you ever had suicidal thoughts ?', 'Family History of`

In [164... `data2`

Out[164...

	Age	City_0	City_1	City_2	City_3	City_4	City_5	Profession_0	Profession_1	F
0	33.0	0	0	0	0	0	1	0	0	
1	24.0	0	0	0	0	1	0	0	0	
2	31.0	0	0	0	0	1	1	0	0	
3	28.0	0	0	0	1	0	0	0	0	
4	25.0	0	0	0	1	0	1	0	0	
...
27896	27.0	0	1	0	0	1	1	0	0	
27897	27.0	0	1	0	1	0	0	0	0	
27898	31.0	0	1	1	1	0	1	0	0	
27899	18.0	0	1	0	1	0	0	0	0	
27900	27.0	0	1	1	1	0	0	0	0	

27836 rows × 33 columns



In [165... `data2[data2['Financial Stress'] == '?']`

Out[165...

	Age	City_0	City_1	City_2	City_3	City_4	City_5	Profession_0	Profession_1	F
4458	32.0	0	0	0	1	0	0	0	0	
13596	29.0	0	1	1	0	0	1	0	0	
19266	20.0	0	0	1	1	1	1	0	0	

3 rows × 33 columns



In [166... `data2 = data2[data2['Financial Stress'] != '?']`

In [167... `data2['Financial Stress'] = data2['Financial Stress'].astype(float)`

C:\Users\lenovo\AppData\Local\Temp\ipykernel_576\4213999668.py:1: SettingWithCopyWarning:
 Warning:
 A value is trying to be set on a copy of a slice from a DataFrame.
 Try using .loc[row_indexer,col_indexer] = value instead

 See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
 data2['Financial Stress'] = data2['Financial Stress'].astype(float)

In [168... data2['Financial Stress'].dtype

Out[168... dtype('float64')

In [169... cat_data = data.select_dtypes(include='category')

In [170... cat_data

Out[170...

age_groups	
0	old man
1	adult
2	old man
3	adult
4	adult
...	...
27896	adult
27897	adult
27898	old man
27899	teen
27900	adult

27836 rows × 1 columns

In [171... cat_data['age_groups'].unique()

Out[171... ['old man', 'adult', 'teen', 'very old']
 Categories (4, object): ['teen' < 'adult' < 'old man' < 'very old']

In [172... final_data = data.drop(columns=['age_groups'])

In [173... final_data.isna().sum()

```
Out[173... Age 0
City_0 0
City_1 0
City_2 0
City_3 0
City_4 0
City_5 0
Profession_0 0
Profession_1 0
Profession_2 0
Profession_3 0
Academic Pressure 0
Work Pressure 0
CGPA 0
Study Satisfaction 0
Job Satisfaction 0
Dietary Habits 0
Degree 0
Have you ever had suicidal thoughts ? 0
Work/Study Hours 0
Financial Stress 0
Family History of Mental Illness 0
Depression 0
Degree_Level 0
Female 0
Male 0
'5-6 hours' 0
'7-8 hours' 0
'Less than 5 hours' 0
'More than 8 hours' 0
dtype: int64
```

```
In [174... final_data.info()
```



```
<class 'pandas.core.frame.DataFrame'>
Index: 27836 entries, 0 to 27900
Data columns (total 30 columns):
#   Column                                     Non-Null Count  Dtype
---  -
0   Age                                         27836 non-null  float64
1   City_0                                     27836 non-null  int64
2   City_1                                     27836 non-null  int64
3   City_2                                     27836 non-null  int64
4   City_3                                     27836 non-null  int64
5   City_4                                     27836 non-null  int64
6   City_5                                     27836 non-null  int64
7   Profession_0                               27836 non-null  int64
8   Profession_1                               27836 non-null  int64
9   Profession_2                               27836 non-null  int64
10  Profession_3                               27836 non-null  int64
11  Academic Pressure                           27836 non-null  float64
12  Work Pressure                               27836 non-null  float64
13  CGPA                                         27836 non-null  float64
14  Study Satisfaction                           27836 non-null  float64
15  Job Satisfaction                           27836 non-null  float64
16  Dietary Habits                             27836 non-null  object
17  Degree                                       27836 non-null  object
18  Have you ever had suicidal thoughts ?      27836 non-null  object
19  Work/Study Hours                           27836 non-null  float64
20  Financial Stress                           27836 non-null  object
21  Family History of Mental Illness           27836 non-null  object
22  Depression                                  27836 non-null  int64
23  Degree_Level                               27836 non-null  object
24  Female                                       27836 non-null  int32
25  Male                                         27836 non-null  int32
26  '5-6 hours'                               27836 non-null  int32
27  '7-8 hours'                               27836 non-null  int32
28  'Less than 5 hours'                        27836 non-null  int32
29  'More than 8 hours'                        27836 non-null  int32
dtypes: float64(7), int32(6), int64(11), object(6)
memory usage: 5.9+ MB
```

```
In [175... final_data.columns
```

```
Out[175... Index(['Age', 'City_0', 'City_1', 'City_2', 'City_3', 'City_4', 'City_5',
      'Profession_0', 'Profession_1', 'Profession_2', 'Profession_3',
      'Academic Pressure', 'Work Pressure', 'CGPA', 'Study Satisfaction',
      'Job Satisfaction', 'Dietary Habits', 'Degree',
      'Have you ever had suicidal thoughts ?', 'Work/Study Hours',
      'Financial Stress', 'Family History of Mental Illness', 'Depression',
      'Degree_Level', 'Female', 'Male', '5-6 hours', '7-8 hours',
      'Less than 5 hours', 'More than 8 hours'],
      dtype='object')
```

```
In [176... final_object_data = final_data.select_dtypes(include='object')
```

```
In [177... final_object_data.columns
```

```
Out[177... Index(['Dietary Habits', 'Degree', 'Have you ever had suicidal thoughts ?',
      'Financial Stress', 'Family History of Mental Illness', 'Degree_Level'],
      dtype='object')
```

```
In [178... final_data.drop(['Dietary Habits', 'Degree', 'Have you ever had suicidal thought',
                  'Financial Stress', 'Family History of Mental Illness', 'Degree_Level'],
```

```
In [179... final_object_data1 = final_data.select_dtypes(include='object')
final_object_data1.columns
```

```
Out[179... Index([], dtype='object')
```

```
In [180... # Mental Illness
# suicidal thoughts
```

```
In [181... final_data.head()
```

```
Out[181...
```

	Age	City_0	City_1	City_2	City_3	City_4	City_5	Profession_0	Profession_1	Profession_2
0	33.0	0	0	0	0	0	1	0	0	0
1	24.0	0	0	0	0	1	0	0	0	0
2	31.0	0	0	0	0	1	1	0	0	0
3	28.0	0	0	0	1	0	0	0	0	0
4	25.0	0	0	0	1	0	1	0	0	0

5 rows × 24 columns



```
In [182... final_data.columns
```

```
Out[182... Index(['Age', 'City_0', 'City_1', 'City_2', 'City_3', 'City_4', 'City_5',
        'Profession_0', 'Profession_1', 'Profession_2', 'Profession_3',
        'Academic Pressure', 'Work Pressure', 'CGPA', 'Study Satisfaction',
        'Job Satisfaction', 'Work/Study Hours', 'Depression', 'Female', 'Male',
        ''5-6 hours'', ''7-8 hours'', ''Less than 5 hours'',
        ''More than 8 hours''],
        dtype='object')
```

```
In [183... df = df.rename(columns={'old_name1': 'new_name1', 'old_name2': 'new_name2'})
```

```
In [184... data2.head()
```

Out[184...

	Age	City_0	City_1	City_2	City_3	City_4	City_5	Profession_0	Profession_1	Profession_2
0	33.0	0	0	0	0	0	1	0	0	0
1	24.0	0	0	0	0	1	0	0	0	0
2	31.0	0	0	0	0	1	1	0	0	0
3	28.0	0	0	0	1	0	0	0	0	0
4	25.0	0	0	0	1	0	1	0	0	0

5 rows × 33 columns

In [185...] `final_object_data1 = data2.select_dtypes(include='object')`In [186...] `final_object_data1.columns`Out[186...] `Index([], dtype='object')`In [187...] `data2.columns`

Out[187...] `Index(['Age', 'City_0', 'City_1', 'City_2', 'City_3', 'City_4', 'City_5',
 'Profession_0', 'Profession_1', 'Profession_2', 'Profession_3',
 'Academic Pressure', 'Work Pressure', 'CGPA', 'Study Satisfaction',
 'Job Satisfaction', 'Work/Study Hours', 'Financial Stress',
 'Depression', 'age_groups', 'Female', 'Male', '5-6 hours',
 '7-8 hours', 'Less than 5 hours', 'More than 8 hours',
 'habits_encoded', 'degree_encoded', 'degree_level_encoded', 'No', 'Yes',
 'No', 'Yes'],
 dtype='object')`

In [188...] `catdata = data2.select_dtypes(include='category')`In [189...] `catdata`

Out[189...

age_groups	
0	old man
1	adult
2	old man
3	adult
4	adult
...	...
27896	adult
27897	adult
27898	old man
27899	teen
27900	adult

27833 rows × 1 columns

In [190...

```
final = data2.drop(columns=['age_groups'])
```

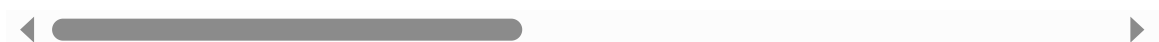
In [191...

```
final.head()
```

Out[191...

	Age	City_0	City_1	City_2	City_3	City_4	City_5	Profession_0	Profession_1	Profession_2
0	33.0	0	0	0	0	0	1	0	0	0
1	24.0	0	0	0	0	1	0	0	0	0
2	31.0	0	0	0	0	1	1	0	0	0
3	28.0	0	0	0	1	0	0	0	0	0
4	25.0	0	0	0	1	0	1	0	0	0

5 rows × 32 columns



In [192...

```
final = final.rename(columns={'Yes': 'suicidal_thoughts_yes', 'No': 'suicidal_thoughts_no'})
```

In [193...

```
final.head()
```

Out[193...

	Age	City_0	City_1	City_2	City_3	City_4	City_5	Profession_0	Profession_1	Profession_2
0	33.0	0	0	0	0	0	1	0	0	0
1	24.0	0	0	0	0	1	0	0	0	0
2	31.0	0	0	0	0	1	1	0	0	0
3	28.0	0	0	0	1	0	0	0	0	0
4	25.0	0	0	0	1	0	1	0	0	0

5 rows × 32 columns



In [194...

`final.columns`

Out[194...

```
Index(['Age', 'City_0', 'City_1', 'City_2', 'City_3', 'City_4', 'City_5',
      'Profession_0', 'Profession_1', 'Profession_2', 'Profession_3',
      'Academic Pressure', 'Work Pressure', 'CGPA', 'Study Satisfaction',
      'Job Satisfaction', 'Work/Study Hours', 'Financial Stress',
      'Depression', 'Female', 'Male', "'5-6 hours'", "'7-8 hours'",
      "'Less than 5 hours'", "'More than 8 hours'", 'habits_encoded',
      'degree_encoded', 'degree_level_encoded', 'suicidal_thoughts_no',
      'suicidal_thoughts_yes', 'suicidal_thoughts_no',
      'suicidal_thoughts_yes'],
      dtype='object')
```

In [195...

```
final.columns.values[-1] = 'mental_illness_yes'
final.columns.values[-2] = 'mental_illness_no'
```

In [196...

`final.head()`

Out[196...

	Age	City_0	City_1	City_2	City_3	City_4	City_5	Profession_0	Profession_1	Profession_2
0	33.0	0	0	0	0	0	1	0	0	0
1	24.0	0	0	0	0	1	0	0	0	0
2	31.0	0	0	0	0	1	1	0	0	0
3	28.0	0	0	0	1	0	0	0	0	0
4	25.0	0	0	0	1	0	1	0	0	0

5 rows × 32 columns



In [197...

```
x = final.drop(columns=['Depression'])
y = final['Depression']
```

In [198...

```
x_train, x_test, y_train, y_test = train_test_split(x, y, random_state=42, test_
```

```
In [199... lg = LogisticRegression()
```

```
In [200... lg.fit(x_train, y_train)
```

c:\Users\lenovo\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\linear_model_logistic.py:458: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:

<https://scikit-learn.org/stable/modules/preprocessing.html>

Please also refer to the documentation for alternative solver options:

https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
n_iter_i = _check_optimize_result(

```
Out[200... ▼ LogisticRegression  
LogisticRegression()
```

```
In [201... lg.score(x_test, y_test)
```

```
Out[201... 0.8476737919885037
```

```
In [202... lg.score(x_train, y_train)
```

```
Out[202... 0.8483337824485763
```

```
In [203... from sklearn.svm import SVC
```

```
In [204... sv = SVC()
```

```
In [205... sv.fit(x_train, y_train)
```

```
Out[205... ▼ SVC  
SVC()
```

```
In [206... sv.score(x_test, y_test)
```

```
Out[206... 0.8469552721393928
```

```
In [207... sv.score(x_train, y_train)
```

```
Out[207... 0.848872720740142
```

```
In [208... from sklearn.tree import DecisionTreeClassifier  
from sklearn.ensemble import RandomForestClassifier  
from xgboost import XGBClassifier
```

```
In [210... dt = DecisionTreeClassifier()  
rf = RandomForestClassifier()  
xg = XGBClassifier()
```

```
In [211... dt_params = {  
    'criterion': ['gini', 'entropy'],
```

```
'max_depth': [5, 10, 15, 20],
'min_samples_split': [2, 5, 10],
'min_samples_leaf': [1, 2, 4]
}
```

```
In [212... from sklearn.model_selection import GridSearchCV
```

```
In [213... gs = GridSearchCV(dt, dt_params)
```

```
In [214... gs.fit(x_train, y_train)
```

```
Out[214...
└─ GridSearchCV
  └─ estimator: DecisionTreeClassifier
    └─ DecisionTreeClassifier
```

```
In [215... gs.best_params_
```

```
Out[215... {'criterion': 'entropy',
'max_depth': 5,
'min_samples_leaf': 1,
'min_samples_split': 2}
```

```
In [216... gs.best_score_
```

```
Out[216... 0.8240814732044944
```

```
In [217... rf_params = {
    'n_estimators': [100, 200, 300]
}
```

```
In [218... gs2 = GridSearchCV(rf, rf_params)
```

```
In [219... gs2.fit(x_train, y_train)
```

```
Out[219...
└─ GridSearchCV
  └─ estimator: RandomForestClassifier
    └─ RandomForestClassifier
```

```
In [220... gs2.best_params_
```

```
Out[220... {'n_estimators': 300}
```

```
In [221... gs2.best_score_
```

```
Out[221... 0.842090835267839
```

```
In [222... xg_params = {
    'n_estimators': [100, 200],
    'max_depth': [3, 6, 9],
    'learning_rate': [0.01, 0.1, 0.2]
}
```

```
In [223... gs3 = GridSearchCV(xg, xg_params)
```

```
In [224... gs3.fit(x_train, y_train)
```

```
Out[224...
  ▸ GridSearchCV
  ▸ estimator: XGBClassifier
    ▸ XGBClassifier
```

```
In [225... gs3.best_params_
```

```
Out[225... {'learning_rate': 0.2, 'max_depth': 3, 'n_estimators': 100}
```

```
In [226... gs3.best_score_
```

```
Out[226... 0.847525162020004
```

```
In [227... from sklearn.metrics import confusion_matrix
```

```
In [228... xg1 = XGBClassifier()
```

```
In [229... xg1.fit(x_train, y_train)
```

```
Out[229...
  ▾ XGBClassifier
  XGBClassifier(base_score=None, booster=None, callbacks=None,
                colsample_bylevel=None, colsample_bynode=None,
                colsample_bytree=None, device=None, early_stopping_rounds=None,
                enable_categorical=False, eval_metric=None, feature_types=None,
                gamma=None, grow_policy=None, importance_type=None,
                interaction_constraints=None, learning_rate=None, max_bin=None,
```

```
In [230... y_pred = xg1.predict(x_test)
```

```
In [231... cm = confusion_matrix(y_test, y_pred)
```

```
In [232... plt.figure(figsize=(6, 4))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=xg1.classes_, yticklabels=xg1.classes_)
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix')
plt.show()
```