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

df=pd.read_csv("Student Mental health.csv")

df.head()
```

```
        Timestamp Choose your gender   Age What is your course?  \
0  8/7/2020 12:02             Female  18.0          Engineering
1  8/7/2020 12:04               Male  21.0    Islamic education
2  8/7/2020 12:05               Male  19.0                  BIT
3  8/7/2020 12:06             Female  22.0                 Laws
4  8/7/2020 12:13               Male  23.0         Mathemathics

  Your current year of Study What is your CGPA? Marital status  \
0                     year 1        3.00 - 3.49             No
1                     year 2        3.00 - 3.49             No
2                     Year 1        3.00 - 3.49             No
3                     year 3        3.00 - 3.49            Yes
4                     year 4        3.00 - 3.49             No

  Do you have Depression? Do you have Anxiety? Do you have Panic
attack?  \
0                     Yes                   No
Yes
1                      No                  Yes
No
2                     Yes                  Yes
Yes
3                     Yes                   No
No
4                      No                   No
No

  Did you seek any specialist for a treatment?
0                                           No
1                                           No
2                                           No
3                                           No
4                                           No
```

```python
df.tail()
```

```
               Timestamp Choose your gender   Age What is your course?
\
96   13/07/2020 19:56:49             Female  21.0                  BCS

97   13/07/2020 21:21:42               Male  18.0          Engineering
```

```
98   13/07/2020 21:22:56           Female  19.0           Nursing

99   13/07/2020 21:23:57           Female  23.0     Pendidikan Islam

100  18/07/2020 20:16:21             Male  20.0   Biomedical science


    Your current year of Study What is your CGPA? Marital status  \
96                      year 1        3.50 - 4.00             No
97                      Year 2        3.00 - 3.49             No
98                      Year 3        3.50 - 4.00            Yes
99                      year 4        3.50 - 4.00             No
100                     Year 2        3.00 - 3.49             No

    Do you have Depression? Do you have Anxiety? Do you have Panic
attack?  \
96                        No                  Yes
No
97                       Yes                  Yes
No
98                       Yes                   No
Yes
99                        No                   No
No
100                       No                   No
No

    Did you seek any specialist for a treatment?
96                                             No
97                                             No
98                                             No
99                                             No
100                                            No
```

df.info

```
<bound method DataFrame.info of                Timestamp Choose your
gender   Age What is your course?  \
0         8/7/2020 12:02           Female  18.0          Engineering

1         8/7/2020 12:04             Male  21.0    Islamic education

2         8/7/2020 12:05             Male  19.0                  BIT

3         8/7/2020 12:06           Female  22.0                 Laws

4         8/7/2020 12:13             Male  23.0          Mathemathics

..                   ...              ...   ...                  ...

96   13/07/2020 19:56:49           Female  21.0                  BCS
```

```
97   13/07/2020 21:21:42              Male  18.0         Engineering

98   13/07/2020 21:22:56            Female  19.0            Nursing

99   13/07/2020 21:23:57            Female  23.0    Pendidikan Islam

100  18/07/2020 20:16:21              Male  20.0   Biomedical science


    Your current year of Study What is your CGPA? Marital status  \
0                       year 1        3.00 - 3.49             No
1                       year 2        3.00 - 3.49             No
2                       Year 1        3.00 - 3.49             No
3                       year 3        3.00 - 3.49            Yes
4                       year 4        3.00 - 3.49             No
..                         ...                ...            ...
96                      year 1        3.50 - 4.00             No
97                      Year 2        3.00 - 3.49             No
98                      Year 3        3.50 - 4.00            Yes
99                      year 4        3.50 - 4.00             No
100                     Year 2        3.00 - 3.49             No

    Do you have Depression? Do you have Anxiety? Do you have Panic
attack?  \
0                       Yes                   No
Yes
1                        No                  Yes
No
2                       Yes                  Yes
Yes
3                       Yes                   No
No
4                        No                   No
No
..                      ...                  ...
...
96                       No                  Yes
No
97                      Yes                  Yes
No
98                      Yes                   No
Yes
99                       No                   No
No
100                      No                   No
No

    Did you seek any specialist for a treatment?
0                                             No
```

```
1                                                        No
2                                                        No
3                                                        No
4                                                        No
..                                                      ...
96                                                       No
97                                                       No
98                                                       No
99                                                       No
100                                                      No

[101 rows x 11 columns]>

df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 101 entries, 0 to 100
Data columns (total 11 columns):
 #   Column                                        Non-Null Count
Dtype
---  ------                                        --------------
-----
 0   Timestamp                                     101 non-null
object
 1   Choose your gender                            101 non-null
object
 2   Age                                           100 non-null
float64
 3   What is your course?                          101 non-null
object
 4   Your current year of Study                    101 non-null
object
 5   What is your CGPA?                            101 non-null
object
 6   Marital status                                101 non-null
object
 7   Do you have Depression?                       101 non-null
object
 8   Do you have Anxiety?                          101 non-null
object
 9   Do you have Panic attack?                     101 non-null
object
 10  Did you seek any specialist for a treatment?  101 non-null
object
dtypes: float64(1), object(10)
memory usage: 8.8+ KB

df.columns #df.columns is the more commonly used attribute to retrieve
the column names.
```

```
Index(['Timestamp', 'Choose your gender', 'Age', 'What is your
course?',
       'Your current year of Study', 'What is your CGPA?', 'Marital
status',
       'Do you have Depression?', 'Do you have Anxiety?',
       'Do you have Panic attack?',
       'Did you seek any specialist for a treatment?'],
      dtype='object')
```

```
print(df.keys())# .df.keys() is the more commonly used attribute to
retrieve the column names.
```

```
Index(['Timestamp', 'Choose your gender', 'Age', 'What is your
course?',
       'Your current year of Study', 'What is your CGPA?', 'Marital
status',
       'Do you have Depression?', 'Do you have Anxiety?',
       'Do you have Panic attack?',
       'Did you seek any specialist for a treatment?'],
      dtype='object')
```

```
null_values = df.isnull()

# Count the number of null values in each column
null_count_per_column = df.isnull().sum()

# Display the DataFrame of null values or the count per column
print(null_values)
print(null_count_per_column)
```

```
     Timestamp  Choose your gender    Age  What is your course?  \
0        False               False  False                 False
1        False               False  False                 False
2        False               False  False                 False
3        False               False  False                 False
4        False               False  False                 False
..         ...                 ...    ...                   ...
96       False               False  False                 False
97       False               False  False                 False
98       False               False  False                 False
99       False               False  False                 False
100      False               False  False                 False

     Your current year of Study  What is your CGPA?  Marital status  \
0                         False               False           False
1                         False               False           False
2                         False               False           False
3                         False               False           False
4                         False               False           False
..                          ...                 ...             ...
```

```
96                        False                False           False
97                        False                False           False
98                        False                False           False
99                        False                False           False
100                       False                False           False

     Do you have Depression?  Do you have Anxiety?  Do you have Panic
attack?  \
0                      False                 False
False
1                      False                 False
False
2                      False                 False
False
3                      False                 False
False
4                      False                 False
False
..                       ...                   ...
...
96                     False                 False
False
97                     False                 False
False
98                     False                 False
False
99                     False                 False
False
100                    False                 False
False

     Did you seek any specialist for a treatment?
0                                           False
1                                           False
2                                           False
3                                           False
4                                           False
..                                            ...
96                                          False
97                                          False
98                                          False
99                                          False
100                                         False

[101 rows x 11 columns]
Timestamp                                       0
Choose your gender                              0
Age                                             1
What is your course?                            0
Your current year of Study                      0
```

```
What is your CGPA?                                0
Marital status                                    0
Do you have Depression?                           0
Do you have Anxiety?                              0
Do you have Panic attack?                         0
Did you seek any specialist for a treatment?      0
dtype: int64
```

Analyzing student mental health data can provide valuable insights into various aspects. Here are some additional analyses you can perform

1 Prevalence of Mental Health Conditions:

now we Calculate the percentage or count of students with depression, anxiety, panic attacks, etc. Visualize the distribution of mental health conditions using bar charts or pie charts.

```
# Example code for prevalence analysis
depression_percentage = (df['Do you have Depression?'].value_counts()
/ len(df)) * 100

print(depression_percentage)

Do you have Depression?
No     65.346535
Yes    34.653465
Name: count, dtype: float64

# Example code for demographic analysis
gender_depression_counts = df.groupby('Choose your gender')['Do you
have Depression?'].value_counts().unstack()

print(gender_depression_counts)

Do you have Depression?  No  Yes
Choose your gender
Female                   46   29
Male                     20    6
```

Treatment Seekers Analysis:

Explore characteristics of students who sought specialist treatment for mental health. Compare their demographics and conditions with those who did not seek treatm

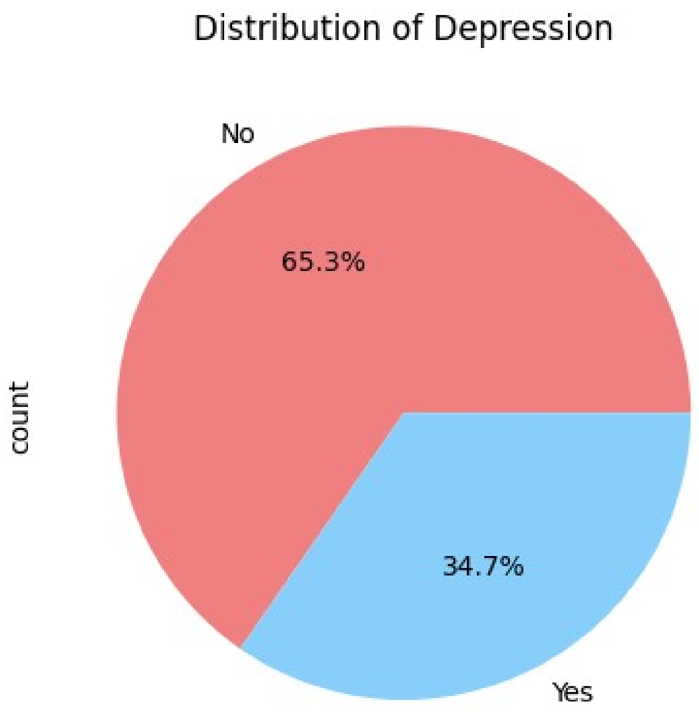
```
# Example code for treatment seekers analysis
treatment_seekers_info = df[df['Did you seek any specialist for a
treatment?'] == 'Yes']
print(treatment_seekers_info)
```

```
              Timestamp Choose your gender   Age What is your course?  \
28       8/7/2020 13:58             Female  24.0                  BIT

33       8/7/2020 14:31               Male  18.0                  BCS

39       8/7/2020 14:56             Female  24.0          Engineering

50       8/7/2020 15:27             Female  23.0                  ALA

54       8/7/2020 15:57             Female  19.0                  BCS

85  13/07/2020 10:33:47             Female  18.0           psychology


   Your current year of Study What is your CGPA? Marital status  \
28                     Year 3        3.50 - 4.00            Yes
33                     Year 2        3.50 - 4.00            Yes
39                     Year 2        2.50 - 2.99            Yes
50                     year 1        2.50 - 2.99            Yes
54                     year 1        3.50 - 4.00             No
85                     year 1        3.50 - 4.00             No

   Do you have Depression? Do you have Anxiety? Do you have Panic attack?  \
28                     Yes                  Yes
Yes
33                     Yes                  Yes
No
39                     Yes                   No
Yes
50                     Yes                   No
Yes
54                     Yes                   No
Yes
85                     Yes                  Yes
No

   Did you seek any specialist for a treatment?
28                                          Yes
33                                          Yes
39                                          Yes
50                                          Yes
54                                          Yes
85                                          Yes
```

Pie Chart for Mental Health Conditions:

Create a pie chart to visually represent the distribution of mental health conditions among students.

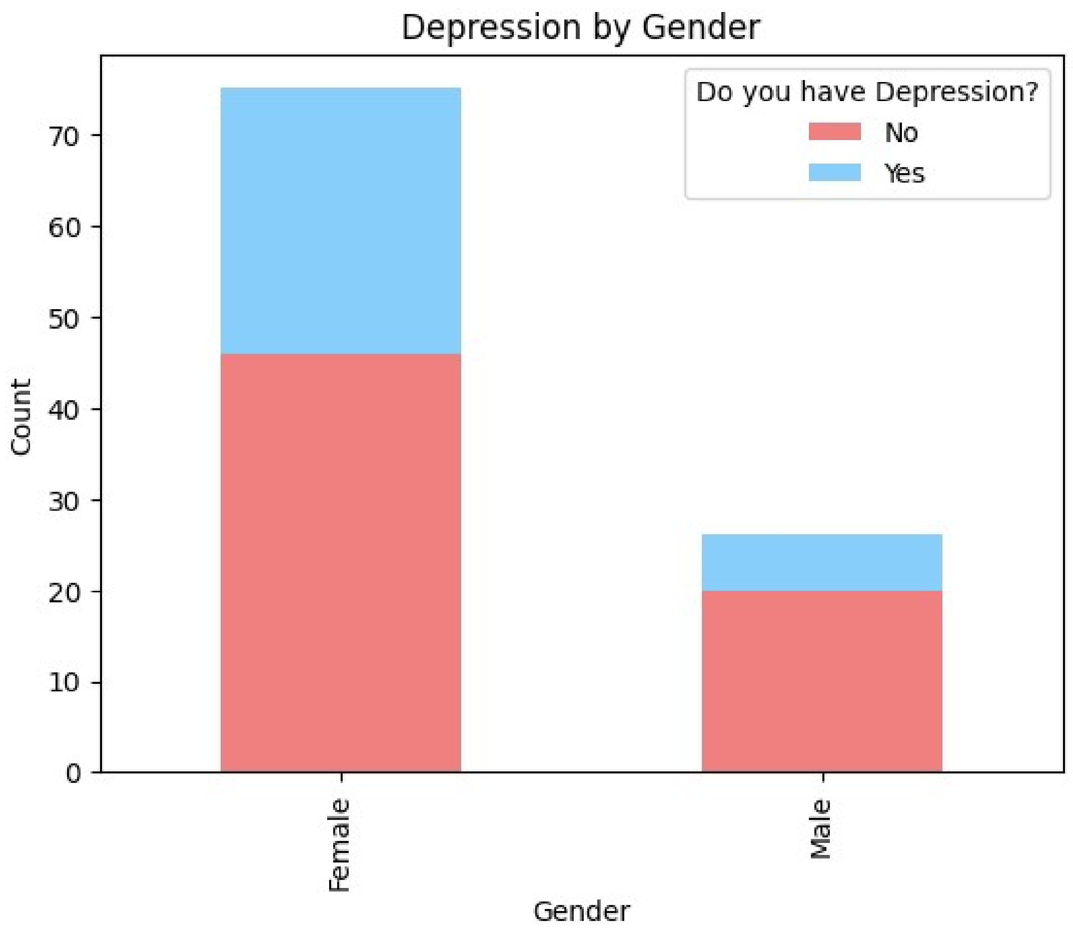
```python
# Example code for pie chart
import matplotlib.pyplot as plt

mental_health_counts = df['Do you have Depression?'].value_counts()
mental_health_counts.plot(kind='pie', autopct='%1.1f%%',
colors=['lightcoral', 'lightskyblue'])
plt.title('Distribution of Depression')
plt.show()
```



Bar Chart for Demographic Analysis:

Use stacked bar charts to visualize the prevalence of mental health conditions based on demographic factors like gender.

```python
# Example code for stacked bar chart
import seaborn as sns

gender_depression_counts = df.groupby('Choose your gender')['Do you
have Depression?'].value_counts().unstack()
gender_depression_counts.plot(kind='bar', stacked=True,
color=['lightcoral', 'lightskyblue'])
plt.title('Depression by Gender')
plt.xlabel('Gender')
plt.ylabel('Count')
plt.show()
```



```
# Example code for box plot
import
sns.boxplot(x='Do you have Depression?', y='CGPA', data=df)
plt.title('CGPA Distribution by Depression Status')
plt.show()

-----------------------------------------------------------------------
-----
ValueError                                Traceback (most recent call
last)
Cell In[41], line 2
      1 # Example code for box plot
----> 2 sns.boxplot(x='Do you have Depression?', y='CGPA', data=df)
      3 plt.title('CGPA Distribution by Depression Status')
      4 plt.show()

File D:\Iriun Webcam\New folder\Lib\site-packages\seaborn\
categorical.py:1596, in boxplot(data, x, y, hue, order, hue_order,
orient, color, palette, saturation, fill, dodge, width, gap, whis,
linecolor, linewidth, fliersize, hue_norm, native_scale, log_scale,
formatter, legend, ax, **kwargs)
```

```
   1588 def boxplot(
   1589     data=None, *, x=None, y=None, hue=None, order=None,
hue_order=None,
   1590     orient=None, color=None, palette=None, saturation=.75,
fill=True,
   (...)
   1593     legend="auto", ax=None, **kwargs
   1594 ):
-> 1596     p = _CategoricalPlotter(
   1597         data=data,
   1598         variables=dict(x=x, y=y, hue=hue),
   1599         order=order,
   1600         orient=orient,
   1601         color=color,
   1602         legend=legend,
   1603     )
   1605     if ax is None:
   1606         ax = plt.gca()

File D:\Iriun Webcam\New folder\Lib\site-packages\seaborn\
categorical.py:66, in _CategoricalPlotter.__init__(self, data,
variables, order, orient, require_numeric, color, legend)
     55 def __init__(
     56     self,
     57     data=None,
   (...)
     63     legend="auto",
     64 ):
---> 66     super().__init__(data=data, variables=variables)
     68     # This method takes care of some bookkeeping that is
necessary because the
     69     # original categorical plots (prior to the 2021 refactor)
had some rules that
     70     # don't fit exactly into VectorPlotter logic. It may be
wise to have a second
   (...)
     75     # default VectorPlotter rules. If we do decide to make
orient part of the
     76     # _base variable assignment, we'll want to figure out how
to express that.
     77     if self.input_format == "wide" and orient in ["h", "y"]:

File D:\Iriun Webcam\New folder\Lib\site-packages\seaborn\
_base.py:634, in VectorPlotter.__init__(self, data, variables)
    629 # var_ordered is relevant only for categorical axis variables,
and may
    630 # be better handled by an internal axis information object
that tracks
    631 # such information and is set up by the scale_* methods. The
```

```
analogous
    632 # information for numeric axes would be information about log 
scales.
    633 self._var_ordered = {"x": False, "y": False}  # alt., used 
DefaultDict
--> 634 self.assign_variables(data, variables)
    636 # TODO Lots of tests assume that these are called to 
initialize the
    637 # mappings to default values on class initialization. I'd 
prefer to
    638 # move away from that and only have a mapping when explicitly 
called.
    639 for var in ["hue", "size", "style"]:

File D:\Iriun Webcam\New folder\Lib\site-packages\seaborn\
_base.py:679, in VectorPlotter.assign_variables(self, data, variables)
    674 else:
    675     # When dealing with long-form input, use the newer 
PlotData
    676     # object (internal but introduced for the objects 
interface)
    677     # to centralize / standardize data consumption logic.
    678     self.input_format = "long"
--> 679     plot_data = PlotData(data, variables)
    680     frame = plot_data.frame
    681     names = plot_data.names

File D:\Iriun Webcam\New folder\Lib\site-packages\seaborn\_core\
data.py:58, in PlotData.__init__(self, data, variables)
     51 def __init__(
     52     self,
     53     data: DataSource,
     54     variables: dict[str, VariableSpec],
     55 ):
     57     data = handle_data_source(data)
---> 58     frame, names, ids = self._assign_variables(data, 
variables)
     60     self.frame = frame
     61     self.names = names

File D:\Iriun Webcam\New folder\Lib\site-packages\seaborn\_core\
data.py:232, in PlotData._assign_variables(self, data, variables)
    230     else:
    231         err += "An entry with this name does not appear in 
`data`."
--> 232     raise ValueError(err)
    234 else:
    235 
    236     # Otherwise, assume the value somehow represents data
    237 
    238     # Ignore empty data structures
    239     if isinstance(val, Sized) and len(val) == 0:

ValueError: Could not interpret value `CGPA` for `y`. An entry with
this name does not appear in `data`.
```