# Exploratory Data Analysis

Exploratory Data Analysis (EDA) is a process of describing the data by means of statistical and visualization techniques in order to bring important aspects of that data into focus for further analysis. This involves inspecting the dataset from many angles, describing & summarizing it without making any assumptions about its contents.

## Steps in EDA

1. Define your Problem
2. Load your Dataset
3. Explore your Dataset
4. Cleaning your Dataset

## 1. Define your Problem

Here, try to define the problem or research questions that you will be answering.

You are not creating a report, you are trying to understand the problem.

The results are ultimately throw-away, and all that you should be left with is a greater understanding and intuition for the data and a long list of hypotheses to explore when modeling.

## 2. Load your Dataset

```
# Before you load your dataset, Its a advised to import the libraries you will be using.
#If you get moduleNotFoundError run pip install pandas numpy matplotlib seaborn to install the modules

import pandas as pd
 #for data manipulation and analysis. Useful for cleaning, filtering, and transforming datasets.
import numpy as np
# for scientific computing in Python. It provides support for large, multi-dimensional arrays and matrices
import matplotlib.pyplot as plt
#2D plotting library that produces static, animated, and interactive visualizations in Python.
import seaborn as sns
#tatistical data visualization library based on matplotlib. It provides a high-level interface for creatir
```

```
C:\Users\KURIA\AppData\Local\Temp\ipykernel_23172\3814293956.py:4: DeprecationWarning:
Pyarrow will become a required dependency of pandas in the next major release of pandas (pandas 3.0)
(to allow more performant data types, such as the Arrow string type, and better interoperability wit
but was not found to be installed on your system.
If this would cause problems for you,
please provide us feedback at https://github.com/pandas-dev/pandas/issues/54466

  import pandas as pd
```

```
#Use Pandas to load your Dataset

# loading and reading dataset
ds_path = r"D:\APHRC\Projects\Mental Health\data\MH.csv"
df = pd.read_csv(ds_path)
print(df.head)
```

```
<bound method NDFrame.head of       Unnamed: 0       start         end _submission_time FA_Code  S
0              1  2023-06-29  2023-06-29       2023-06-30     ANE    12737
1              2  2023-09-13  2023-09-13       2023-09-14     GYH    14487
2              3  2023-09-28  2023-09-28       2023-09-28     BGO    12027
3              4  2023-07-27  2023-07-27       2023-07-27     MGA     5802
4              5  2023-08-14  2023-08-14       2023-08-21     MLF     7449
...          ...         ...         ...              ...     ...      ...
1809        1810  2023-07-04  2023-07-05       2023-07-05     NAS    15401
1810        1811  2023-08-11  2023-08-11       2023-08-11     BGO     7240
1811        1812  2023-06-26  2023-06-26       2023-06-30     NFL    16808
1812        1813  2023-09-18  2023-09-18       2023-09-18     NPR     3517
1813        1814  2023-08-22  2023-08-22       2023-08-22     KAR    15133

      _index    locationid       villagenam   residence  ...  \
0       1735  I41607189001     NABIDONGHA C  Peri_urban  ...
1       7027  I31501165003      NAKIGO II A       Rural  ...
2       9019  M20901008002         MBALE TC       Rural  ...
3       3667  I31201305001         BUSEYI B       Rural  ...
4       5101  I10503309001  BUSOWOBI CENTRA       Rural  ...
...      ...           ...              ...         ...  ...
1809    2209  I21002157001       NAWANSINGE       Rural  ...
1810    4246  I10503125001  BUSOWOBI CENTRA       Rural  ...
1811    1718  M20704062004            WANTE       Rural  ...
1812    7336  I31401882003  BULUBANDI - CEN  Peri_urban  ...
1813    5371  I10404015001       NAWAMPENDO       Rural  ...

     Thinkpeoplewereagainstyo Havemoodswings  Feelshorttempered  \
0                       Never          Never              Never
1                   Sometimes      Sometimes              Never
2                   Sometimes          Never              Never
3                       Never          Never             Rarely
4                       Never          Never              Never
...                       ...            ...                ...
1809                    Never          Never              Never
1810                    Never          Never              Never
1811                    Never          Never              Never
1812                    Never          Never              Never
1813                   Rarely         Rarely             Rarely

     Thinkabouthurtingyourself  Didyouhaveanurgetodrin  \
0                        Never                   Never
1                        Never                   Never
2                        Never                   Never
3                        Never                   Never
4                        Never                   Never
...                        ...                     ...
1809                     Never                   Never
1810                     Never                   Never
1811                     Never                   Never
1812                     Never                   Never
1813                     Never               Sometimes
```

```
      Didanyonetalktoyouabout Didyoutrytohideyourdri Didyouhaveproblemsfromy  \
0                          NaN                    NaN                      NaN
1                          NaN                    NaN                      NaN
2                          NaN                    NaN                      NaN
3                          NaN                    NaN                      NaN
```

## 3. Explore your Dataset

```
# shape of the data
df.shape
```

```
(1814, 71)
```

```
#data information
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1814 entries, 0 to 1813
Data columns (total 71 columns):
 #   Column                         Non-Null Count  Dtype
---  ------                         --------------  -----
 0   Unnamed: 0                     1814 non-null   int64
 1   start                          1814 non-null   object
 2   end                            1814 non-null   object
 3   _submission_time               1814 non-null   object
 4   FA_Code                        1814 non-null   object
 5   StudyID                        1814 non-null   int64
 6   _index                         1814 non-null   int64
 7   locationid                     1814 non-null   object
 8   villagenam                     1814 non-null   object
 9   residence                      1739 non-null   object
 10  HHHead_id                      1814 non-null   object
 11  gender                         1814 non-null   object
 12  age                            1814 non-null   int64
 13  IndividualId                   1814 non-null   object
 14  ANXIETY                        0 non-null      float64
 15  Overthelast2weekshowoften      0 non-null      float64
 16  Feelingnervousanxiousorone     1814 non-null   object
 17  Notbeingabletostoporcontro     1814 non-null   object
 18  Worryingtoomuchaboutdifferen   1814 non-null   object
 19  Troublerelaxing                1814 non-null   object
 20  Beingsorestlessthatitishar     1814 non-null   object
 21  Becomingeasilyannoyedorirrit   1814 non-null   object
 22  Feelingafraidasifsomethinga    1814 non-null   object
 23  Total                          1814 non-null   int64
 24  DEPRESSION                     0 non-null      float64
 25  Littleinterestorpleasurei      1814 non-null   object
 26  Feelingdowndepressedorh        1814 non-null   object
 27  Troublefallingorstayingas      1814 non-null   object
 28  Feelingtiredorhavinglittl      1814 non-null   object
 29  Poorappetiteorovereating       1814 non-null   object
 30  Feelingbadaboutyourself_or     1814 non-null   object
 31  Troubleconcentratingonthin     1814 non-null   object
 32  MovingorSpeakingsoslowly       1814 non-null   object
 33  Thoughtsthatyouwouldbebe       1814 non-null   object
 34  Total_score                    1814 non-null   int64
 35  PSCHOSIS                       0 non-null      float64
 36  Haveyouhadanystrangeoro        1814 non-null   object
```

```
 37  Doyoueverhearthingsthat       1814 non-null   object
 38  Doyoueverhavevisionsors       1814 non-null   object
 39  Doyoueverfeelthatpeople       1814 non-null   object
 40  Hasiteverseemedlikepeopl      1814 non-null   object
 41  Areyouafraidofanythingor      1814 non-null   object
 42  DuringthePASTWEEKhowmuchd     0 non-null      float64
 43  Managingyourdaytodaylife      1814 non-null   object
 44  Copingwithproblemsinyour      1814 non-null   object
 45  Concentrating                 1814 non-null   object
 46  DuringthePASTWEEKhowmucho     0 non-null      float64
 47  Getalongwithpeopleinyour      1814 non-null   object
 48  Getalongwithpeopleoutside     1814 non-null   object
 49  Getalongwellinsocialsitu      1814 non-null   object
 50  Feelclosetoanotherperson      1814 non-null   object
 51  Feellikeyouhadsomeoneto       1814 non-null   object
 52  Feelconfidentinyourself       1814 non-null   object
```

```
# describing the data
df.describe()
```

| | Unnamed: 0 | StudyID | _index | age | ANXIETY | Overthelast2weekshow |
|---|---|---|---|---|---|---|
| **count** | 1814.000000 | 1814.000000 | 1814.000000 | 1814.000000 | 0.0 | |
| **mean** | 907.500000 | 8431.299338 | 4580.059537 | 52.060088 | NaN | |
| **std** | 523.801012 | 4788.627705 | 2670.713830 | 14.922866 | NaN | |
| **min** | 1.000000 | 37.000000 | 7.000000 | 19.000000 | NaN | |
| **25%** | 454.250000 | 4770.250000 | 2221.250000 | 42.000000 | NaN | |
| **50%** | 907.500000 | 7998.500000 | 4493.000000 | 50.000000 | NaN | |
| **75%** | 1360.750000 | 12746.750000 | 6891.750000 | 61.000000 | NaN | |
| **max** | 1814.000000 | 17013.000000 | 9302.000000 | 105.000000 | NaN | |

The DataFrame “df” is statistically summarized by the code df.describe(), which gives the count, mean, standard deviation, minimum, and quartiles for each numerical column. The dataset’s central tendencies and spread are briefly summarized.

```
#column to list

df.columns.tolist()
```

```
['Unnamed: 0',
 'start',
 'end',
 '_submission_time',
 'FA_Code',
 'StudyID',
 '_index',
 'locationid',
 'villagenam',
 'residence',
 'HHHead_id',
```

```
'gender',
'age',
'IndividualId',
'ANXIETY',
'Overthelast2weekshowoften',
'Feelingnervousanxiousorone',
'Notbeingabletostoporcontro',
'Worryingtoomuchaboutdifferen',
'Troublerelaxing',
'Beingsorestlessthatitishar',
'Becomingeasilyannoyedorirrit',
'Feelingafraidasifsomethinga',
'Total',
'DEPRESSION',
'Littleinterestorpleasurei',
'Feelingdowndepressedorh',
'Troublefallingorstayingas',
'Feelingtiredorhavinglittl',
'Poorappetiteorovereating',
'Feelingbadaboutyourself_or',
'Troubleconcentratingonthin',
'MovingorSpeakingsoslowly',
'Thoughtsthatyouwouldbebe',
'Total_score',
'PSCHOSIS',
'Haveyouhadanystrangeoro',
'Doyoueverhearthingsthat',
'Doyoueverhavevisionsors',
'Doyoueverfeelthatpeople',
'Hasiteverseemedlikepeopl',
'Areyouafraidofanythingor',
'DuringthePASTWEEKhowmuchd',
'Managingyourdaytodaylife',
'Copingwithproblemsinyour',
'Concentrating',
'DuringthePASTWEEKhowmucho',
'Getalongwithpeopleinyour',
'Getalongwithpeopleoutside',
'Getalongwellinsocialsitu',
'Feelclosetoanotherperson',
'Feellikeyouhadsomeoneto',
'Feelconfidentinyourself',
'Feelsadordepressed',
'Thinkaboutendingyourlife',
'Feelnervous',
'DuringthePASTWEEKhowoften',
'Havethoughtsracingthrough'
```

## 4. Cleaning your Dataset

1. Dealing with Null Values
2. Duplicates
3. changing datatypes
4. Deleting irrerevant columns

```
#Checking for null values

# Check for null values in each column
null_columns = df.columns[df.isnull().any()]

# Create a DataFrame with only columns containing null values
df_null_columns = df[null_columns]

# Calculate the total null values in each column
total_null_values_per_column = df_null_columns.isnull().sum()

# Display the result
print("Columns with null values and their total null values:")
print(total_null_values_per_column)
```

```
Columns with null values and their total null values:
residence                     75
ANXIETY                     1814
Overthelast2weekshowoften   1814
DEPRESSION                  1814
PSCHOSIS                    1814
DuringthePASTWEEKhowmuchd   1814
DuringthePASTWEEKhowmucho   1814
DuringthePASTWEEKhowoften   1814
Didanyonetalktoyouabout     1684
Didyoutrytohideyourdri      1684
Didyouhaveproblemsfromy     1684
Anycomments                 1814
dtype: int64
```

Go back and explore your missing data so that you know how to handle them.

Handling Missing Data

**1. Dropping Missing Values:**

Method: Use dropna() method in pandas.

Pros: Simple and quick. Useful when the missing data is random and removing those rows doesn't significantly affect the analysis.

Cons: May lead to loss of information, especially if the missing data is not entirely random.

**2. Imputation:**

Method: Fill in missing values with a specific value (e.g., mean, median, or mode) or use more advanced imputation methods.

Pros: Retains more data compared to dropping. Can be suitable for datasets with systematic missingness.

Cons: Imputed values may introduce bias, and the choice of imputation method is critical.

**3. Forward or Backward Fill:**

Method: Propagate the last valid observation forward or use the next valid observation to fill gaps.

Pros: Simple and suitable for time-series data.

Cons: The method may not be suitable for all types of data, and it assumes a certain temporal pattern.

**4. Interpolation:**

Method: Use methods like linear interpolation to estimate missing values based on surrounding values.

Pros: More sophisticated than simple imputation methods.

Cons: Requires knowledge of underlying patterns and assumptions about the data.

**5. Indicator/Dummy Variables:**

Method: Introduce a binary indicator variable to denote whether a value was missing.

Pros: Retains information about the presence of missing values.

Cons: Increases the dimensionality of the dataset.

**6 Model-Based Imputation:**

Method: Use machine learning models to predict missing values based on other features.

Pros: Takes into account relationships between variables.

Cons: Requires more computational resources and may be overfitting if not handled carefully.

```
#In our case we can drop columns with 1814 missing values as they are insignificant

# Find columns with 1814 null values
columns_to_drop = df.columns[df.isnull().sum() == 1814]

# Drop the columns
df_dropped = df.drop(columns=columns_to_drop)

# Display the resulting DataFrame
print("DataFrame after dropping columns:")
print(df_dropped.columns)

df_dropped.shape
```

```
DataFrame after dropping columns:
Index(['Unnamed: 0', 'start', 'end', '_submission_time', 'FA_Code', 'StudyID',
       '_index', 'locationid', 'villagenam', 'residence', 'HHHead_id',
       'gender', 'age', 'IndividualId', 'Feelingnervousanxiousorone',
       'Notbeingabletostoporcontro', 'Worryingtoomuchaboutdifferen',
       'Troublerelaxing', 'Beingsorestlessthatitishar',
       'Becomingeasilyannoyedorirrit', 'Feelingafraidasifsomethinga', 'Total',
       'Littleinterestorpleasurei', 'Feelingdowndepressedorh',
       'Troublefallingorstayingas', 'Feelingtiredorhavinglittl',
       'Poorappetiteorovereating', 'Feelingbadaboutyourself_or',
       'Troubleconcentratingonthin', 'MovingorSpeakingsoslowly',
       'Thoughtsthatyouwouldbebe', 'Total_score', 'Haveyouhadanystrangeoro',
       'Doyoueverhearthingsthat', 'Doyoueverhavevisionsors',
       'Doyoueverfeelthatpeople', 'Hasiteverseemedlikepeopl',
       'Areyouafraidofanythingor', 'Managingyourdaytodaylife',
       'Copingwithproblemsinyour', 'Concentrating', 'Getalongwithpeopleinyour',
       'Getalongwithpeopleoutside', 'Getalongwellinsocialsitu',
       'Feelclosetoanotherperson', 'Feellikeyouhadsomeoneto',
       'Feelconfidentinyourself', 'Feelsadordepressed',
       'Thinkaboutendingyourlife', 'Feelnervous', 'Havethoughtsracingthrough',
       'Thinkyouhadspecialpowers', 'Hearvoicesorseethings',
       'Thinkpeoplewerewatchingy', 'Thinkpeoplewereagainstyo',
       'Havemoodswings', 'Feelshorttempered', 'Thinkabouthurtingyourself',
       'Didyouhaveanurgetodrin', 'Didanyonetalktoyouabout',
```

```
       'Didyoutrytohideyourdri', 'Didyouhaveproblemsfromy', '_id'],
      dtype='object')
(1814, 63)
```

```
#For residence, 75 people didnt answer whether they are from Rural or Urban areas. If we explore the dat

# Check for null values in the residence column
null_residence_indices = df_dropped[df_dropped['residence'].isnull()].index

# Iterate over the rows with null residence values
for index in null_residence_indices:
    # Check if the village name starts with "BULUBANDI"
    if df_dropped.loc[index, 'villagenam'].startswith("BULUBANDI"):
        # Fill missing value with the corresponding village name
        df_dropped.loc[index, 'residence'] = df_dropped.loc[index, 'villagenam']

# Display the resulting DataFrame
print("DataFrame after filling missing values in the residence column:")
print(df_dropped)
```

```
DataFrame after filling missing values in the residence column:
      Unnamed: 0       start         end _submission_time FA_Code  StudyID  \
0              1  2023-06-29  2023-06-29       2023-06-30     ANE    12737
1              2  2023-09-13  2023-09-13       2023-09-14     GYH    14487
2              3  2023-09-28  2023-09-28       2023-09-28     BGO    12027
3              4  2023-07-27  2023-07-27       2023-07-27     MGA     5802
4              5  2023-08-14  2023-08-14       2023-08-21     MLF     7449
...          ...         ...         ...              ...     ...      ...
1809        1810  2023-07-04  2023-07-05       2023-07-05     NAS    15401
1810        1811  2023-08-11  2023-08-11       2023-08-11     BGO     7240
1811        1812  2023-06-26  2023-06-26       2023-06-30     NFL    16808
1812        1813  2023-09-18  2023-09-18       2023-09-18     NPR     3517
1813        1814  2023-08-22  2023-08-22       2023-08-22     KAR    15133

      _index    locationid       villagenam   residence  ...  \
0       1735  I41607189001     NABIDONGHA C  Peri_urban  ...
1       7027  I31501165003      NAKIGO II A       Rural  ...
2       9019  M20901008002         MBALE TC       Rural  ...
3       3667  I31201305001         BUSEYI B       Rural  ...
4       5101  I10503309001  BUSOWOBI CENTRA       Rural  ...
...      ...           ...              ...         ...  ...
1809    2209  I21002157001       NAWANSINGE       Rural  ...
1810    4246  I10503125001  BUSOWOBI CENTRA       Rural  ...
1811    1718  M20704062004            WANTE       Rural  ...
1812    7336  I31401882003  BULUBANDI - CEN  Peri_urban  ...
1813    5371  I10404015001       NAWAMPENDO       Rural  ...

     Thinkpeoplewerewatchingy Thinkpeoplewereagainstyo  Havemoodswings  \
0                       Never                   Never           Never
1                      Rarely                Sometimes       Sometimes
2                   Sometimes                Sometimes           Never
3                       Never                   Never           Never
4                       Never                   Never           Never
...                       ...                     ...             ...
1809                    Never                   Never           Never
1810                    Never                   Never           Never
1811                    Never                   Never           Never
1812                    Never                   Never           Never
1813                   Rarely                  Rarely          Rarely

     Feelshorttempered Thinkabouthurtingyourself Didyouhaveanurgetodrin  \
```

```
0               Never                    Never                  Never
1               Never                    Never                  Never
2               Never                    Never                  Never
3              Rarely                    Never                  Never
4               Never                    Never                  Never
...               ...                      ...                    ...
1809            Never                    Never                  Never
1810            Never                    Never                  Never
1811            Never                    Never                  Never
1812            Never                    Never                  Never
1813           Rarely                    Never             Sometimes

     Didanyonetalktoyouabout Didyoutrytohideyourdri Didyouhaveproblemsfromy  \
0                        NaN                    NaN                     NaN
1                        NaN                    NaN                     NaN
2                        NaN                    NaN                     NaN
3                        NaN                    NaN                     NaN
```

```
# Check for null values in the "residence" column
null_values_in_residence = df_dropped['residence'].isnull().sum()

# Display the result
print("Number of null values in the 'residence' column:", null_values_in_residence)

df_dropped.shape

df_dropped
```

```
Number of null values in the 'residence' column: 0
```

| | Unnamed: 0 | start | end | _submission_time | FA_Code | StudyID | _index | locationid |
|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 2023-06-29 | 2023-06-29 | 2023-06-30 | ANE | 12737 | 1735 | I41607189001 |
| **1** | 2 | 2023-09-13 | 2023-09-13 | 2023-09-14 | GYH | 14487 | 7027 | I31501165003 |
| **2** | 3 | 2023-09-28 | 2023-09-28 | 2023-09-28 | BGO | 12027 | 9019 | M20901008002 |
| **3** | 4 | 2023-07-27 | 2023-07-27 | 2023-07-27 | MGA | 5802 | 3667 | I31201305001 |
| **4** | 5 | 2023-08-14 | 2023-08-14 | 2023-08-21 | MLF | 7449 | 5101 | I10503309001 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... |
| **1809** | 1810 | 2023-07-04 | 2023-07-05 | 2023-07-05 | NAS | 15401 | 2209 | I21002157001 |
| **1810** | 1811 | 2023-08-11 | 2023-08-11 | 2023-08-11 | BGO | 7240 | 4246 | I10503125001 |
| **1811** | 1812 | 2023-06-26 | 2023-06-26 | 2023-06-30 | NFL | 16808 | 1718 | M20704062004 |
| **1812** | 1813 | 2023-09-18 | 2023-09-18 | 2023-09-18 | NPR | 3517 | 7336 | I31401882003 |
| **1813** | 1814 | 2023-08-22 | 2023-08-22 | 2023-08-22 | KAR | 15133 | 5371 | I10404015001 |

1814 rows × 63 columns

This dataset contains mental health info for Anxiety, depression and Pschosis. For the purposes of this analysis, I will drop columns with pschosis and other irrelevant data.

```
 #List of columns to drop
columns_to_drop = ['Unnamed: 0', '_index','Haveyouhadanystrangeoro',
       'Doyoueverhearthingsthat', 'Doyoueverhavevisionsors',
       'Doyoueverfeelthatpeople', 'Hasiteverseemedlikepeopl',
       'Areyouafraidofanythingor', 'Managingyourdaytodaylife',
       'Copingwithproblemsinyour', 'Concentrating', 'Getalongwithpeopleinyour',
       'Getalongwithpeopleoutside', 'Getalongwellinsocialsitu',
       'Feelclosetoanotherperson', 'Feellikeyouhadsomeoneto',
       'Feelconfidentinyourself', 'Feelsadordepressed',
       'Thinkaboutendingyourlife', 'Feelnervous', 'Havethoughtsracingthrough',
       'Thinkyouhadspecialpowers', 'Hearvoicesorseethings',
       'Thinkpeoplewerewatchingy', 'Thinkpeoplewereagainstyo',
       'Havemoodswings', 'Feelshorttempered', 'Thinkabouthurtingyourself',
       'Didyouhaveanurgetodrin', 'Didanyonetalktoyouabout',
       'Didyoutrytohideyourdri', 'Didyouhaveproblemsfromy', '_id']

# Drop the specified columns
df_dropped1 = df_dropped.drop(columns=columns_to_drop, axis=1)

df_dropped1.shape
```

```
(1814, 30)
```

```
df_dropped1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1814 entries, 0 to 1813
Data columns (total 30 columns):
 #   Column                        Non-Null Count  Dtype 
---  ------                        --------------  ----- 
 0   start                         1814 non-null   object
 1   end                           1814 non-null   object
 2   _submission_time              1814 non-null   object
 3   FA_Code                       1814 non-null   object
 4   StudyID                       1814 non-null   int64 
 5   locationid                    1814 non-null   object
 6   villagenam                    1814 non-null   object
 7   residence                     1814 non-null   object
 8   HHHead_id                     1814 non-null   object
 9   gender                        1814 non-null   object
 10  age                           1814 non-null   int64 
 11  IndividualId                  1814 non-null   object
 12  Feelingnervousanxiousorone    1814 non-null   object
 13  Notbeingabletostoporcontro    1814 non-null   object
 14  Worryingtoomuchaboutdifferen  1814 non-null   object
 15  Troublerelaxing               1814 non-null   object
 16  Beingsorestlessthatitishar    1814 non-null   object
 17  Becomingeasilyannoyedorirrit  1814 non-null   object
 18  Feelingafraidasifsomethinga   1814 non-null   object
 19  Total                         1814 non-null   int64 
 20  Littleinterestorpleasurei     1814 non-null   object
 21  Feelingdowndepressedorh       1814 non-null   object
 22  Troublefallingorstayingas     1814 non-null   object
 23  Feelingtiredorhavinglittl     1814 non-null   object
 24  Poorappetiteorovereating      1814 non-null   object
 25  Feelingbadaboutyourself_or    1814 non-null   object
 26  Troubleconcentratingonthin    1814 non-null   object
 27  MovingorSpeakingsoslowly      1814 non-null   object
 28  Thoughtsthatyouwouldbebe      1814 non-null   object
```

```
 29  Total_score                  1814 non-null   int64
dtypes: int64(4), object(26)
memory usage: 425.3+ KB
```

```
# Display current data types
print(df_dropped1.dtypes)
```

```
start                           object
end                             object
_submission_time                object
FA_Code                         object
StudyID                          int64
locationid                      object
villagenam                      object
residence                       object
HHHead_id                       object
gender                          object
age                              int64
IndividualId                    object
Feelingnervousanxiousorone      object
Notbeingabletostoporcontro      object
Worryingtoomuchaboutdifferen    object
Troublerelaxing                 object
Beingsorestlessthatitishar      object
Becomingeasilyannoyedorirrit    object
Feelingafraidasifsomethinga     object
Total                            int64
Littleinterestorpleasurei       object
Feelingdowndepressedorh         object
Troublefallingorstayingas       object
Feelingtiredorhavinglittl       object
Poorappetiteorovereating        object
Feelingbadaboutyourself_or      object
Troubleconcentratingonthin      object
MovingorSpeakingsoslowly        object
Thoughtsthatyouwouldbebe        object
Total_score                      int64
dtype: object
```

```
print(df_dropped1.dtypes)
```

```
start                           object
end                             object
_submission_time                object
FA_Code                         object
StudyID                          int64
locationid                      object
villagenam                      object
residence                       object
HHHead_id                       object
gender                          object
age                              int64
IndividualId                    object
Feelingnervousanxiousorone      object
Notbeingabletostoporcontro      object
Worryingtoomuchaboutdifferen    object
Troublerelaxing                 object
Beingsorestlessthatitishar      object
Becomingeasilyannoyedorirrit    object
Feelingafraidasifsomethinga     object
Total                            int64
```

```
Littleinterestorpleasurei       object
Feelingdowndepressedorh         object
Troublefallingorstayingas       object
Feelingtiredorhavinglittl       object
Poorappetiteorovereating        object
Feelingbadaboutyourself_or      object
Troubleconcentratingonthin      object
MovingorSpeakingsoslowly        object
Thoughtsthatyouwouldbebe        object
Total_score                      int64
dtype: object
```

```
#Rename the _submission_time column to Submission date
df_dropped1 = df_dropped1.rename(columns={'_submission_time': 'Submission date'})



#Lets work on making sure the data is in correct format



#Convert specific columns to datetime
date_columns = ['start', 'end', 'Submission date']

for col in date_columns:
    df_dropped1[col] = pd.to_datetime(df_dropped1[col], errors='coerce')

# Display data types after the conversion
print("\nData Types After:")
print(df_dropped1.dtypes)
```

```
Data Types After:
start                           datetime64[ns]
end                             datetime64[ns]
Submission date                 datetime64[ns]
FA_Code                                 object
StudyID                                  int64
locationid                              object
villagenam                              object
residence                               object
HHHead_id                               object
gender                                  object
age                                      int64
IndividualId                            object
Feelingnervousanxiousorone              object
Notbeingabletostoporcontro              object
Worryingtoomuchaboutdifferen            object
Troublerelaxing                         object
Beingsorestlessthatitishar              object
Becomingeasilyannoyedorirrit            object
Feelingafraidasifsomethinga             object
Total                                    int64
Littleinterestorpleasurei               object
Feelingdowndepressedorh                 object
Troublefallingorstayingas               object
Feelingtiredorhavinglittl               object
Poorappetiteorovereating                object
Feelingbadaboutyourself_or              object
Troubleconcentratingonthin              object
MovingorSpeakingsoslowly                object
```

```
Thoughtsthatyouwouldbebe               object
Total_score                             int64
dtype: object
```

```
#Save our dataframe as Cleaned_df

clean_df = df_dropped1
```

## 5. Univariate Analysis

# Data Visualization Using Python

Data visualization is the presentation of data in a graphical or visual format. It involves using visual elements such as charts, graphs, and maps to represent and communicate complex information, patterns, and insights from data. The primary goal of data visualization is to make data more accessible, understandable, and interpretable for both technical and non-technical audiences.

## Key aspects of data visualization include:

**Communication:** Data visualization serves as a powerful tool for conveying information clearly and concisely. It allows for the effective communication of trends, patterns, and relationships within data.

**Exploration and Analysis:** Visualization enables data analysts, scientists, and decision-makers to explore and analyze large datasets more efficiently. Visual representations can reveal hidden patterns or outliers that might not be immediately apparent in raw data.

**Decision-Making:** Well-designed visualizations facilitate informed decision-making by providing a visual summary of complex information. Decision-makers can quickly grasp insights and trends, leading to more effective and data-driven decisions.

**Storytelling:** Data visualization is often used to tell a story or present a narrative. By structuring data in a visually compelling way, it becomes easier to engage and persuade the audience.

## Common types of data visualizations include:

**Bar charts and histograms:** Used to represent categorical data or the distribution of numerical data.

**Line charts:** Suitable for displaying trends over time or relationships between variables.

**Scatter plots:** Show the relationship between two continuous variables, highlighting patterns or correlations.

**Pie charts:** Represent parts of a whole, illustrating the proportion of different categories.

**Heatmaps:** Visualize the intensity of values in a matrix using colors.

**Maps:** Display geographical data and spatial relationships.

Popular tools for creating data visualizations include Matplotlib, Seaborn, Plotly, Tableau, and Microsoft Power BI, among others. Effective data visualization is not only about choosing the right charts but also involves thoughtful design choices to enhance clarity, accuracy, and the overall user experience.

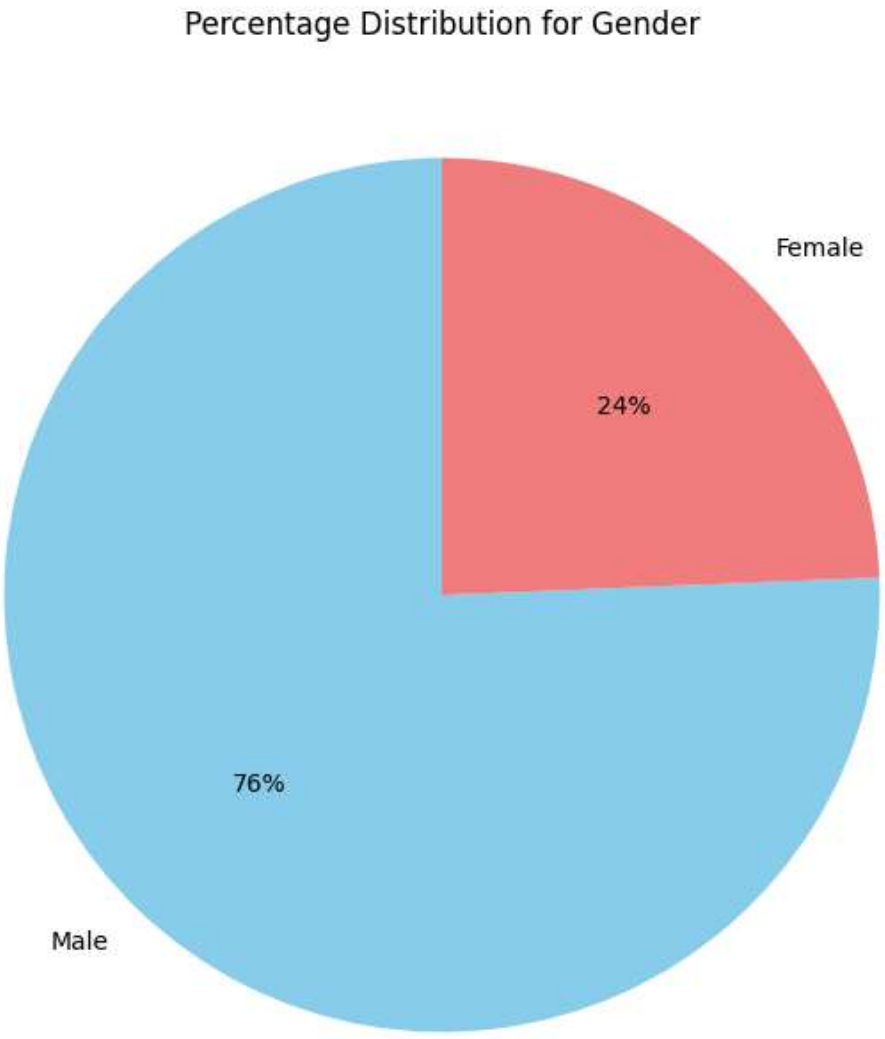
```
print("Descriptive Statistics for Age")
print(clean_df["age"].describe())
```

```
Descriptive Statistics for Age
count    1814.000000
mean       52.060088
std        14.922866
min        19.000000
25%        42.000000
50%        50.000000
75%        61.000000
max       105.000000
Name: age, dtype: float64
```

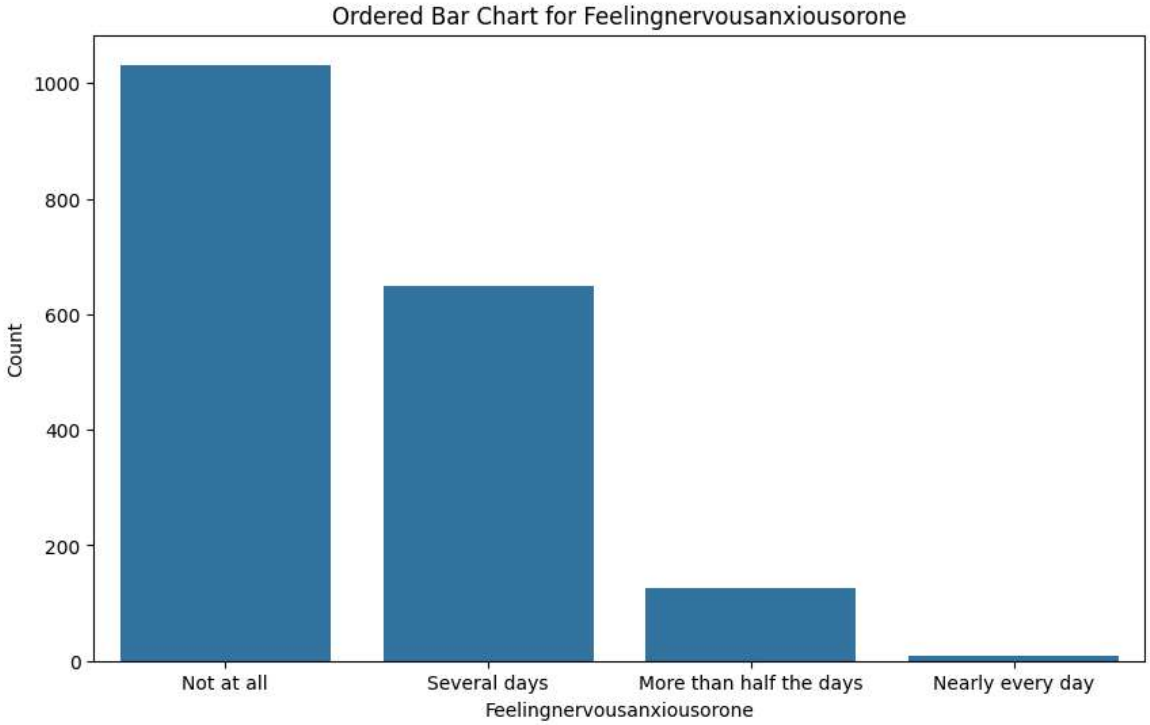
```
#Gender distribution
category_counts = df['gender'].value_counts()

# Calculate percentages and round to 0 decimal places
category_percentages = (category_counts / len(df['gender'])) * 100

# Display the table with counts and rounded percentages
print("Frequency Distribution and Percentages for 'gender':")
print("Count:")
print(category_counts)
print("\nPercentages:")
print(category_percentages.round(0))
```

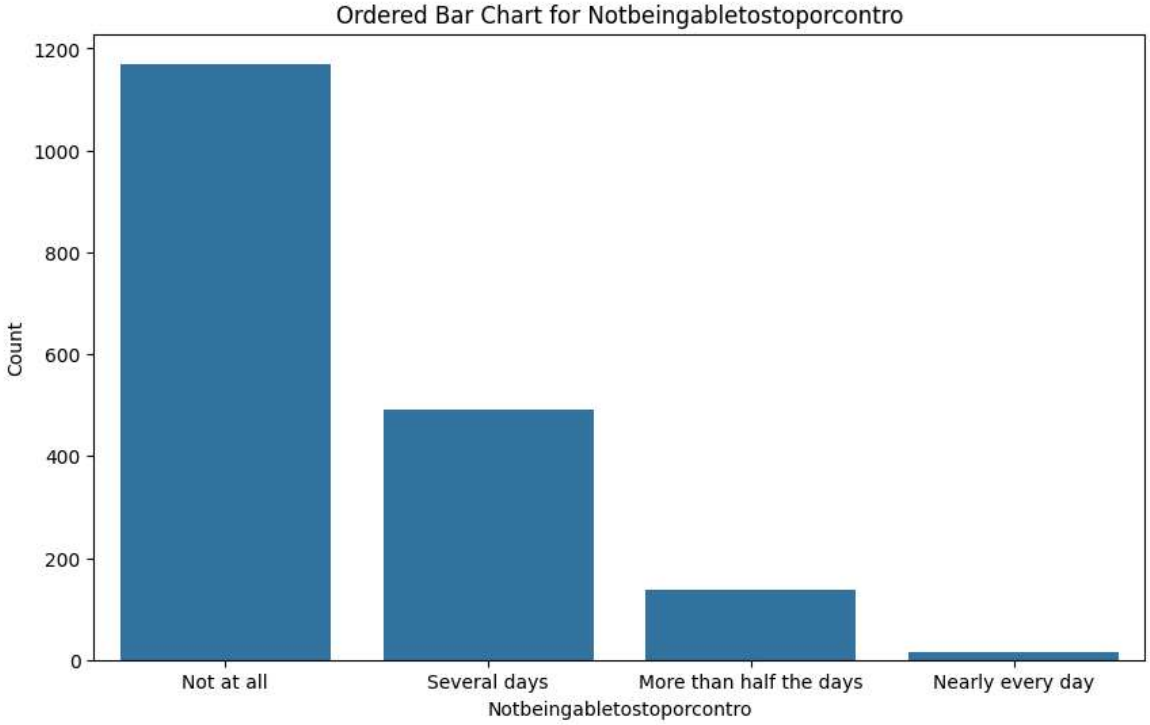
```
Frequency Distribution and Percentages for 'gender':
Count:
gender
Male      1372
Female     442
Name: count, dtype: int64

Percentages:
gender
Male      76.0
Female    24.0
Name: count, dtype: float64
```

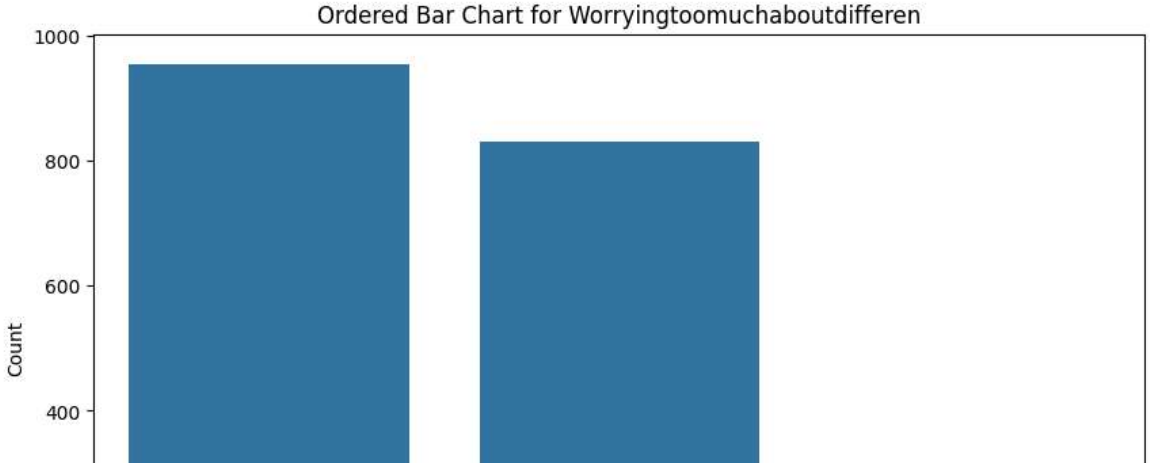
```
# Plot a pie chart
plt.figure(figsize=(8, 8))
plt.pie(category_percentages, labels=['Male', 'Female'], autopct='%1.0f%%', startangle=90, colors=['skyb
plt.title('Percentage Distribution for Gender')
plt.show()
```

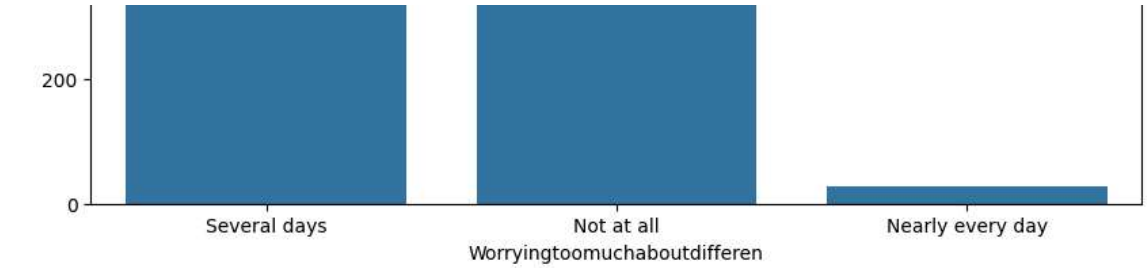


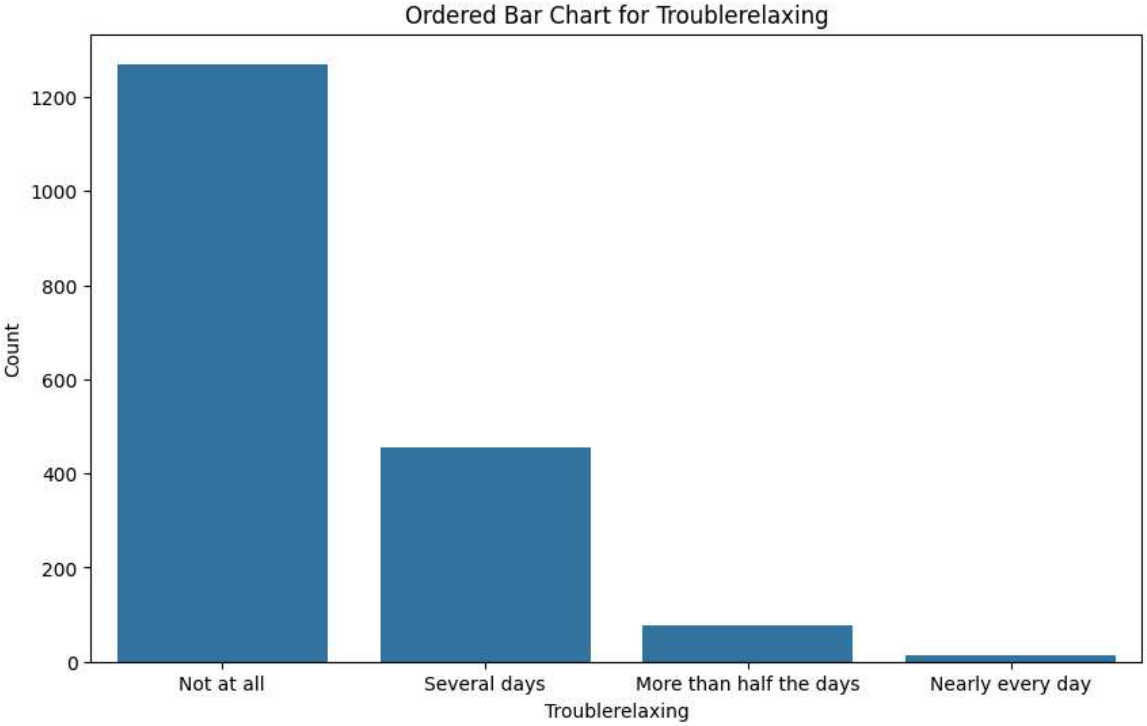

```
anxiety_columns = [
    'Feelingnervousanxiousorone',
    'Notbeingabletostoporcontro',
    'Worryingtoomuchaboutdifferen',
    'Troublerelaxing',
    'Beingsorestlessthatitishar',
    'Becomingeasilyannoyedorirrit',
    'Feelingafraidasifsomethinga'
]

# Plotting ordered bar charts for each anxiety column
for column in anxiety_columns:
    plt.figure(figsize=(10, 6))
    sns.countplot(x=column, data=clean_df, order=clean_df[column].value_counts().index)
    plt.title(f'Ordered Bar Chart for {column}')
    plt.xlabel(column)
    plt.ylabel('Count')
    plt.show()
```

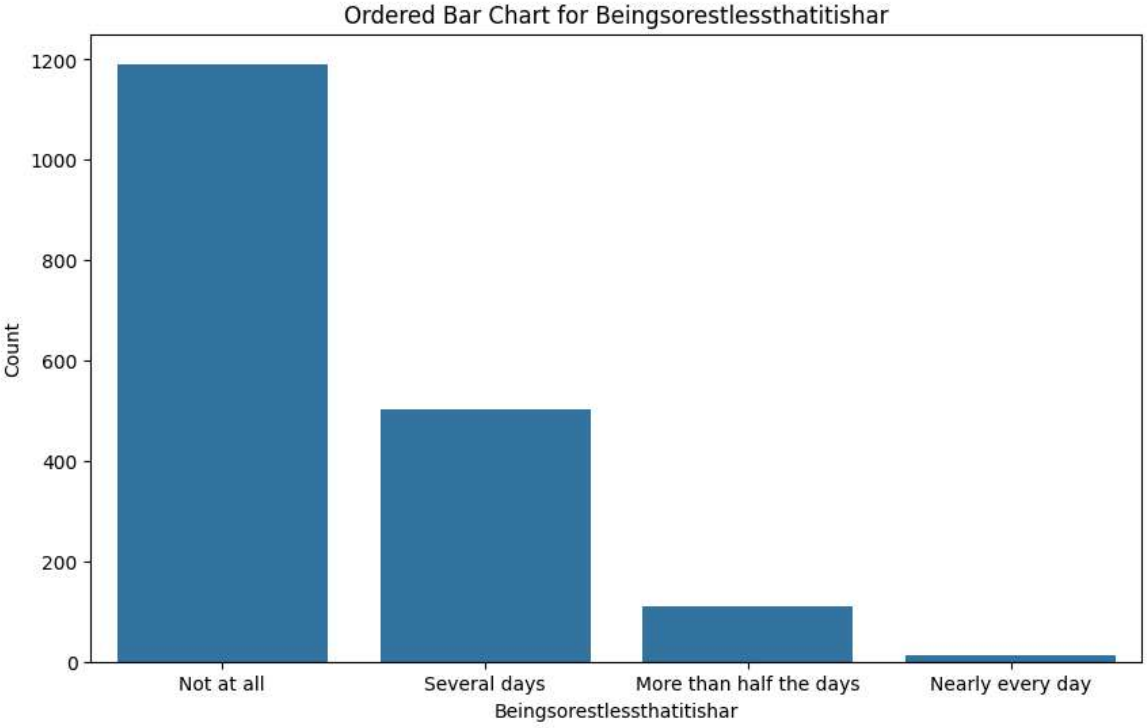

Ordered Bar Chart for Feelingnervousanxiousorone
Count
1000
800
600
400
200
0
Not at all
Several days
More than half the days
Nearly every day
Feelingnervousanxiousorone

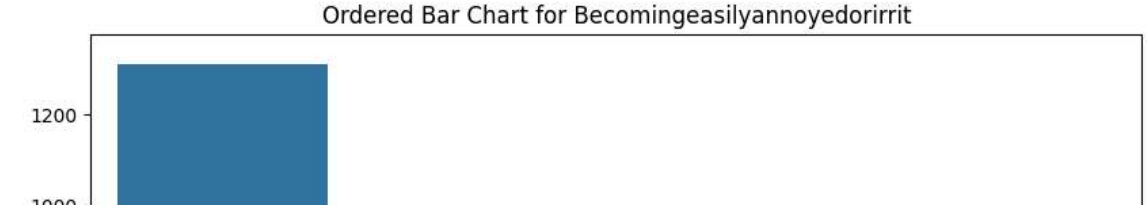

Ordered Bar Chart for Notbeingabletostoporcontro
Count
1200
1000
800
600
400
200
0
Not at all
Several days
More than half the days
Nearly every day
Notbeingabletostoporcontro

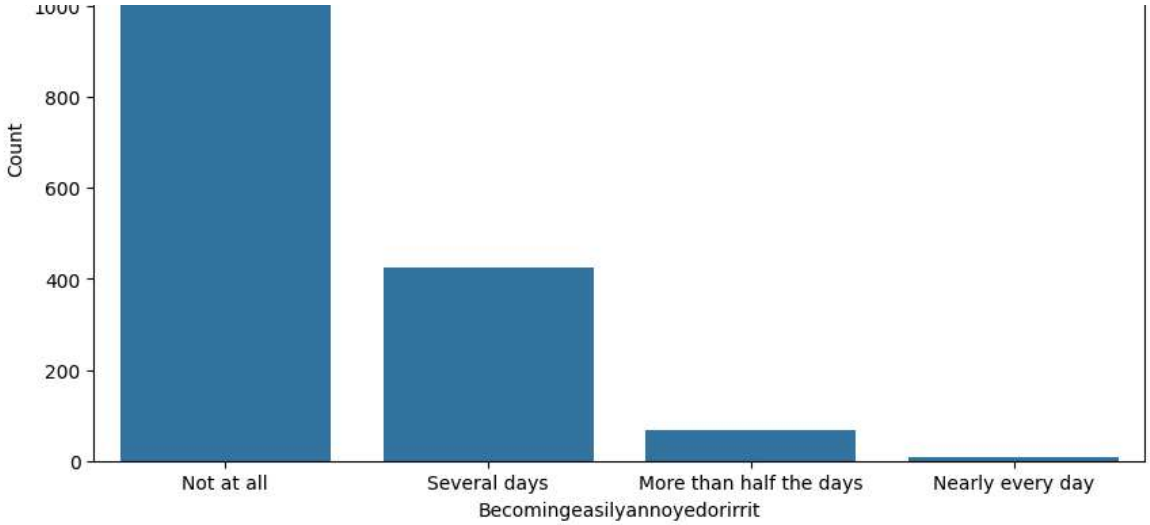

Ordered Bar Chart for Worryingtoomuchaboutdifferen
Count
1000
800
600
400

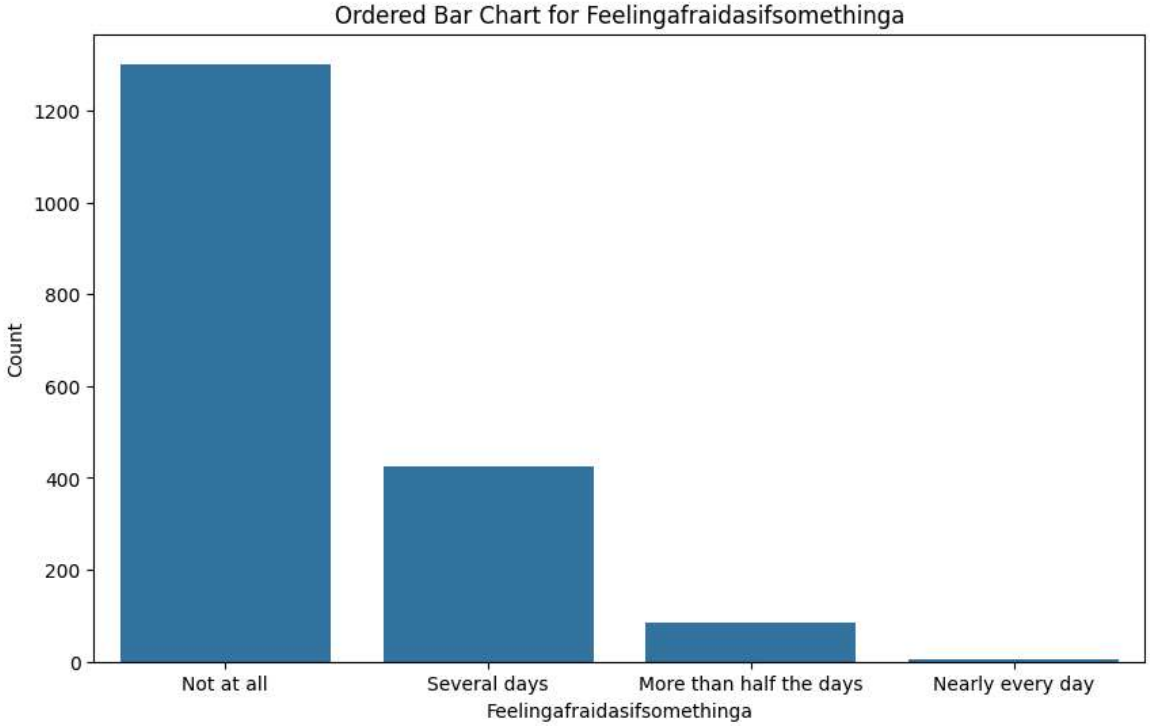

200
0
Several days
Not at all
Nearly every day
Worryingtoomuchaboutdifferen


Ordered Bar Chart for Troublerelaxing
Count
1200
1000
800
600
400
200
0
Not at all
Several days
More than half the days
Nearly every day
Troublerelaxing

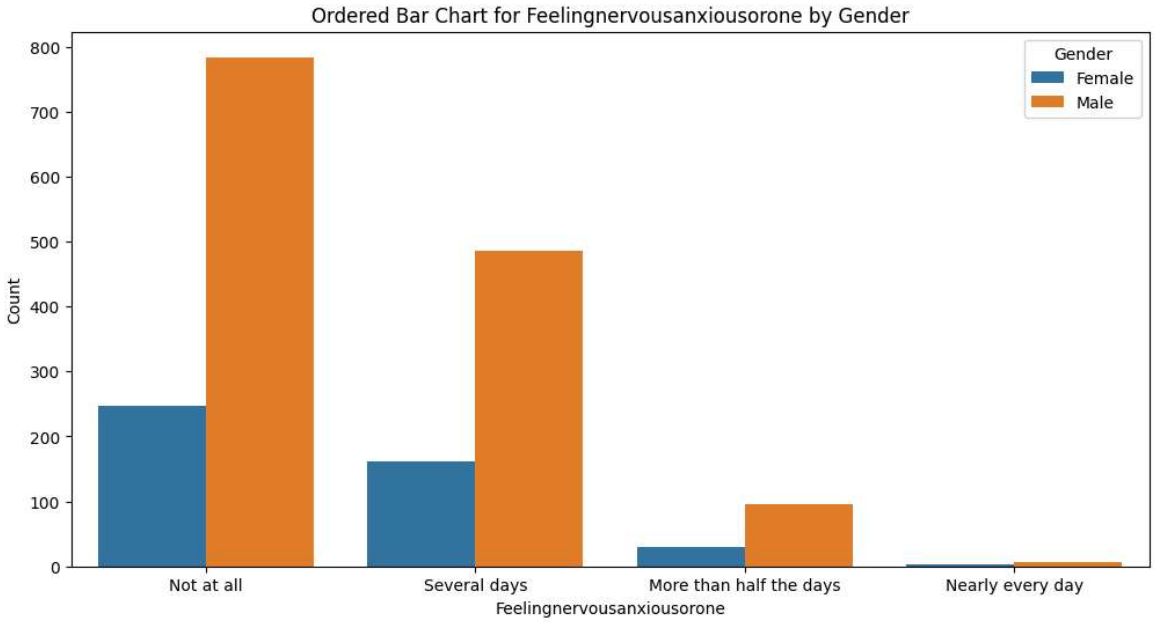

Ordered Bar Chart for Beingsorestlessthatitishar
Count
1200
1000
800
600
400
200
0
Not at all
Several days
More than half the days
Nearly every day
Beingsorestlessthatitishar

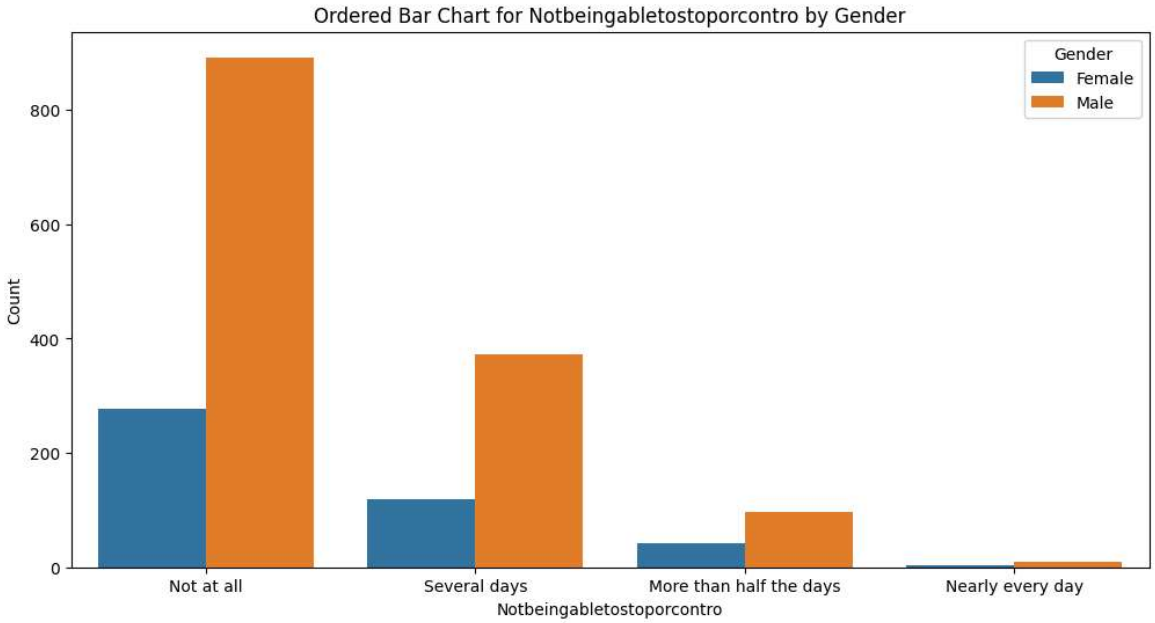

Ordered Bar Chart for Becomingeasilyannoyedorirrit
1200

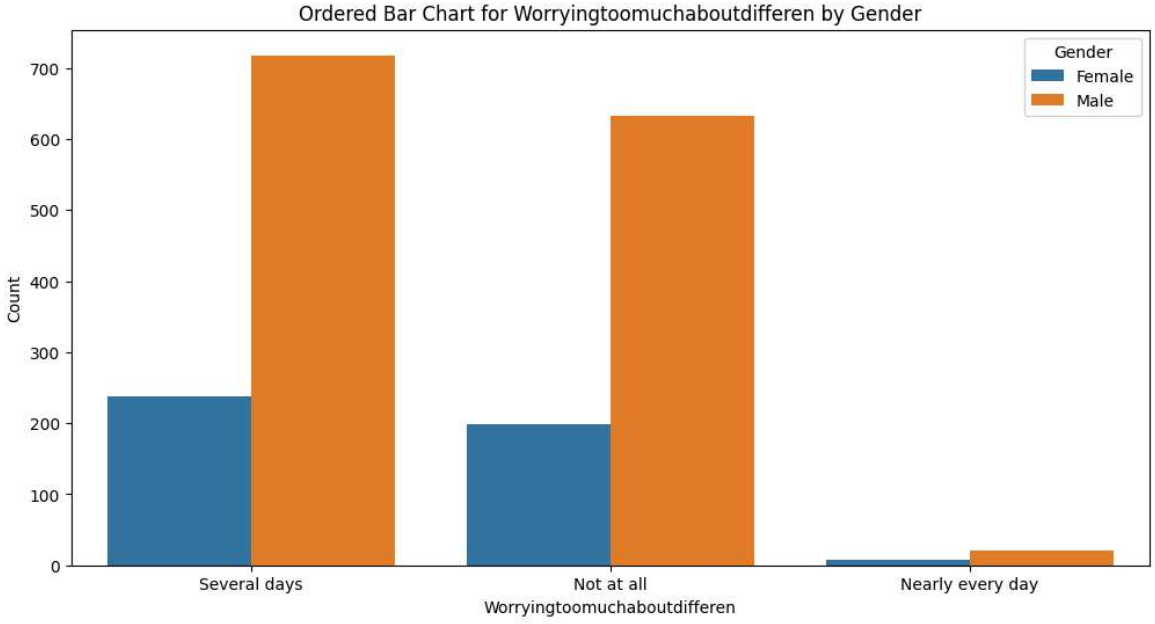

1000
800
600
400
200
0
Count
Not at all
Several days
More than half the days
Nearly every day
Becomingeasilyannoyedorirrit

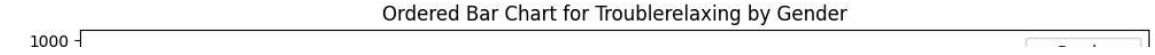

Ordered Bar Chart for Feelingafraidasifsomethinga
1200
1000
800
600
400
200
0
Count
Not at all
Several days
More than half the days
Nearly every day
Feelingafraidasifsomethinga

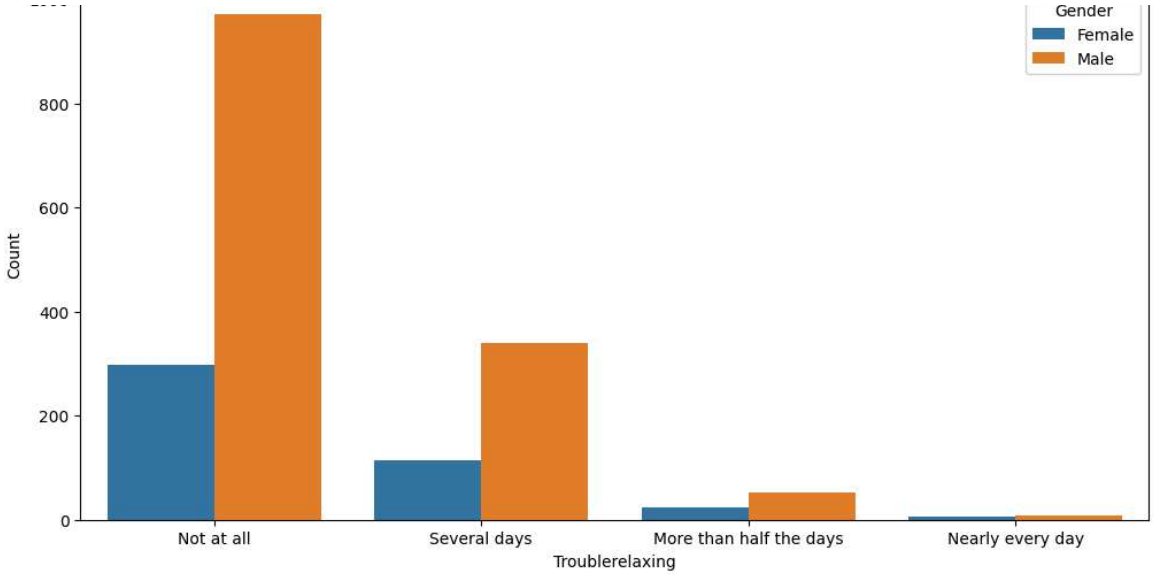
```
# Plotting ordered bar charts for each anxiety column with comparison by gender
for column in anxiety_columns:
    plt.figure(figsize=(12, 6))
    sns.countplot(x=column, hue='gender', data=clean_df, order=clean_df[column].value_counts().index)
    plt.title(f'Ordered Bar Chart for {column} by Gender')
    plt.xlabel(column)
    plt.ylabel('Count')
    plt.legend(title='Gender')
    plt.show()
```


Ordered Bar Chart for Feelingnervousanxiousorone by Gender
Gender
Female
Male
Count
0
100
200
300
400
500
600
700
800
Not at all
Several days
More than half the days
Nearly every day
Feelingnervousanxiousorone

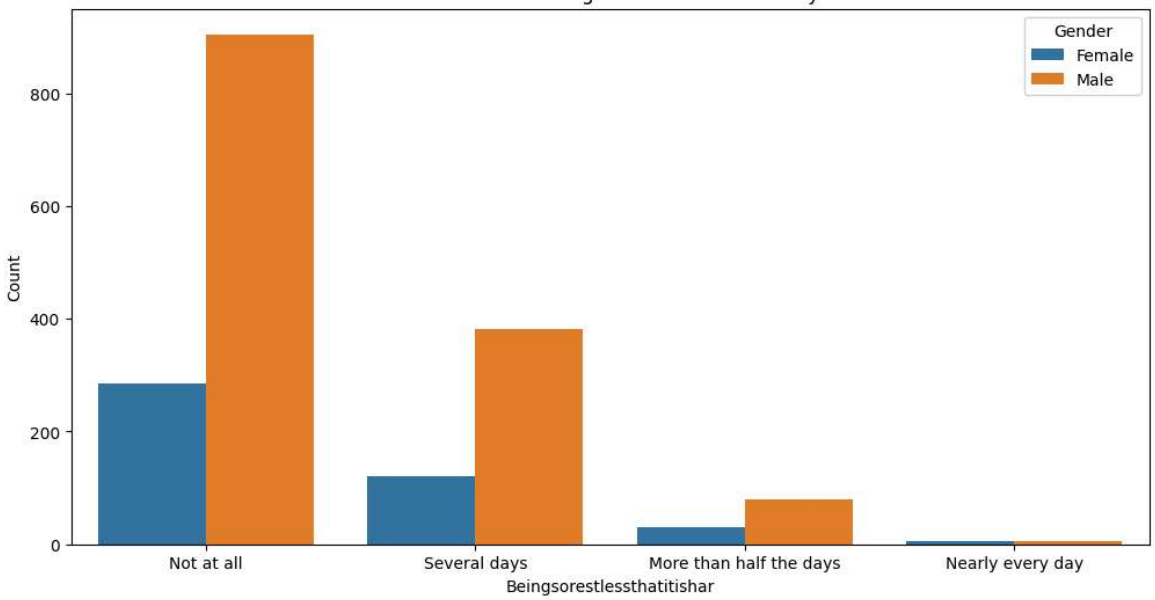

Ordered Bar Chart for Notbeingabletostoporcontro by Gender
Gender
Female
Male
Count
0
200
400
600
800
Not at all
Several days
More than half the days
Nearly every day
Notbeingabletostoporcontro


Ordered Bar Chart for Worryingtoomuchaboutdifferen by Gender
Gender
Female
Male
Count
0
100
200
300
400
500
600
700
Several days
Not at all
Nearly every day
Worryingtoomuchaboutdifferen

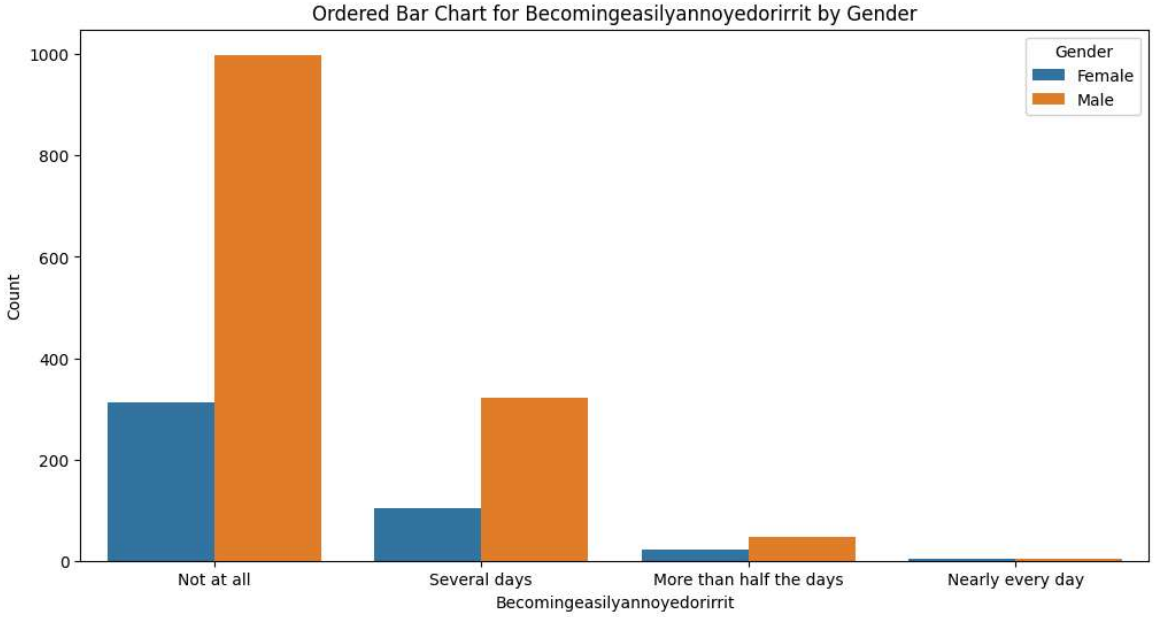

Ordered Bar Chart for Troublerelaxing by Gender
1000



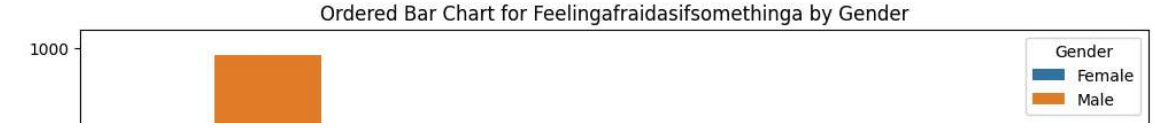

Ordered Bar Chart for Beingsorestlessthatitishar by Gender

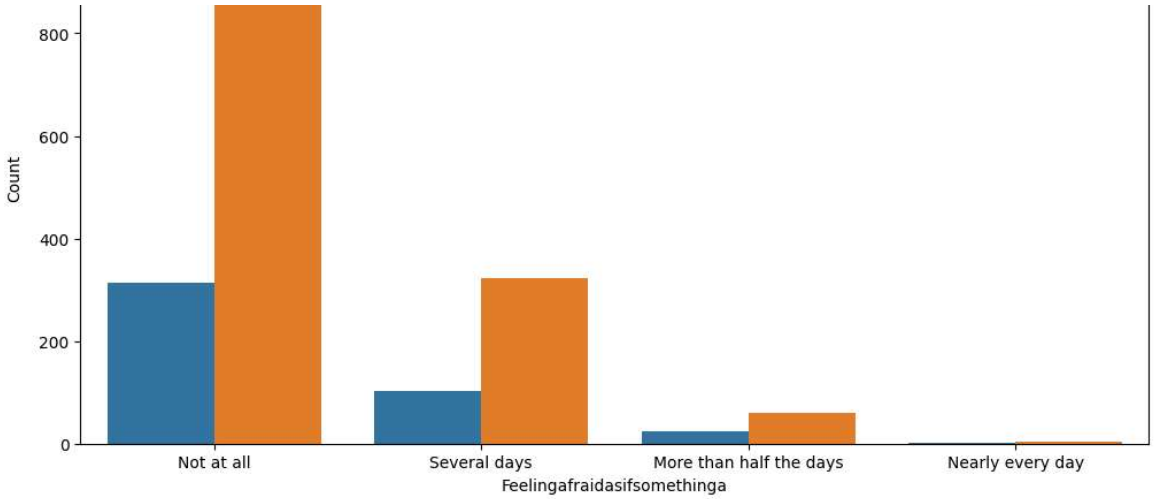



Ordered Bar Chart for Becomingeasilyannoyedorirrit by Gender




Ordered Bar Chart for Feelingafraidasifsomethinga by Gender




800
600
400
200
0
Count
Not at all
Several days
More than half the days
Nearly every day
Feelingafraidasifsomethinga

```
# Create age groups
bins = [0, 18, 30, 45, 60, float('inf')]
labels = ['0-18', '19-30', '31-45', '46-60', '60+']
clean_df['age_group'] = pd.cut(clean_df['age'], bins=bins, labels=labels, right=False)


# Plotting ordered bar charts for each anxiety column with comparison by age group
for column in anxiety_columns:
    plt.figure(figsize=(12, 6))
    sns.countplot(x=column, hue='age_group', data=clean_df, order=clean_df[column].value_counts().index)
    plt.title(f'Ordered Bar Chart for {column} by Age Group')
    plt.xlabel(column)
    plt.ylabel('Count')
    plt.legend(title='Age Group')
    plt.show()
```