In [1]:

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

In [2]:

```
df = pd.read_csv('mxmh_survey_results.csv')
df.head()
```

Out[2]:

	Timestamp	Age	Primary streaming service	Hours per day	While working	Instrumentalist	Composer
0	8/27/2022 19:29:02	18.0	Spotify	3.0	Yes	Yes	Yes
1	8/27/2022 19:57:31	63.0	Pandora	1.5	Yes	No	No
2	8/27/2022 21:28:18	18.0	Spotify	4.0	No	No	No
3	8/27/2022 21:40:40	61.0	YouTube Music	2.5	Yes	No	Yes
4	8/27/2022 21:54:47	18.0	Spotify	4.0	Yes	No	No

5 rows × 33 columns

```
In [3]: for col in df.columns:
            print("Datatype of",col,"is",df[col].dtypes)
            print("")
```

```
Datatype of Timestamp is object
Datatype of Age is float64
Datatype of Primary streaming service is object
Datatype of Hours per day is float64
Datatype of While working is object
Datatype of Instrumentalist is object
Datatype of Composer is object
Datatype of Fav genre is object
Datatype of Exploratory is object
Datatype of Foreign languages is object
Datatype of BPM is float64
Datatype of Frequency [Classical] is object
Datatype of Frequency [Country] is object
Datatype of Frequency [EDM] is object
Datatype of Frequency [Folk] is object
Datatype of Frequency [Gospel] is object
Datatype of Frequency [Hip hop] is object
Datatype of Frequency [Jazz] is object
Datatype of Frequency [K pop] is object
Datatype of Frequency [Latin] is object
Datatype of Frequency [Lofi] is object
Datatype of Frequency [Metal] is object
Datatype of Frequency [Pop] is object
Datatype of Frequency [R&B] is object
Datatype of Frequency [Rap] is object
Datatype of Frequency [Rock] is object

Datatype of Frequency [Video game music] is object

Datatype of Anxiety is float64

Datatype of Depression is float64

Datatype of Insomnia is float64

Datatype of OCD is float64

Datatype of Music effects is object

Datatype of Permissions is object
```

In [4]:

```
df.shape
```

Out[4]:

```
(736, 33)
```

In [5]:

```
df.describe()
```

Out[5]:

	Age	Hours per day	BPM	Anxiety	Depression	Insor
count	735.000000	736.000000	6.290000e+02	736.000000	736.000000	736.00
mean	25.206803	3.572758	1.589948e+06	5.837636	4.796196	3.73
std	12.054970	3.028199	3.987261e+07	2.793054	3.028870	3.08
min	10.000000	0.000000	0.000000e+00	0.000000	0.000000	0.00
25%	18.000000	2.000000	1.000000e+02	4.000000	2.000000	1.00
50%	21.000000	3.000000	1.200000e+02	6.000000	5.000000	3.00
75%	28.000000	5.000000	1.440000e+02	8.000000	7.000000	6.00
max	89.000000	24.000000	1.000000e+09	10.000000	10.000000	10.00

In [6]:

```
df1 = df.copy()
```

Outlier Detection

In [7]:

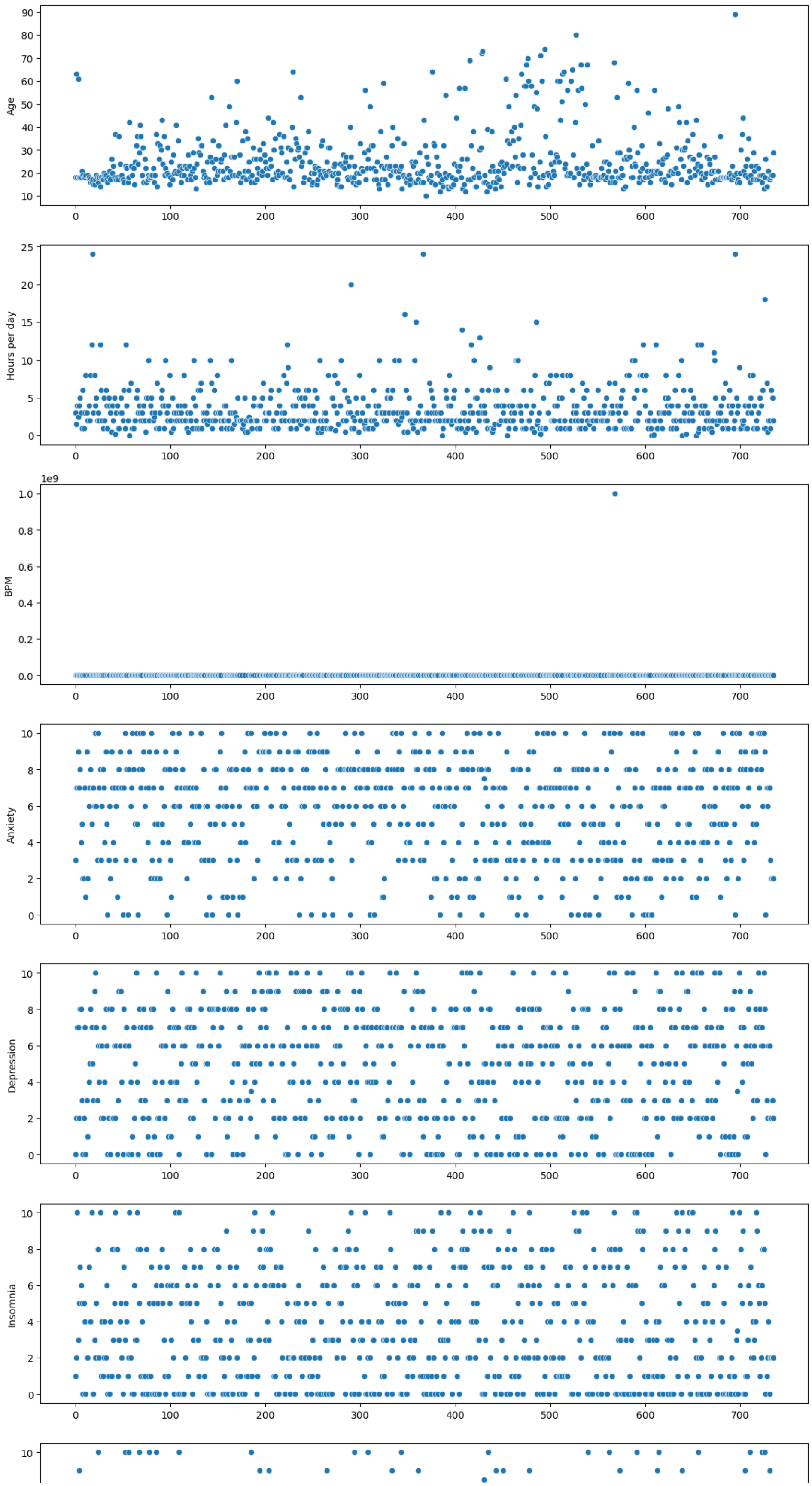
```
num_col = df1.select_dtypes(include = 'number')
num_col.head()
```

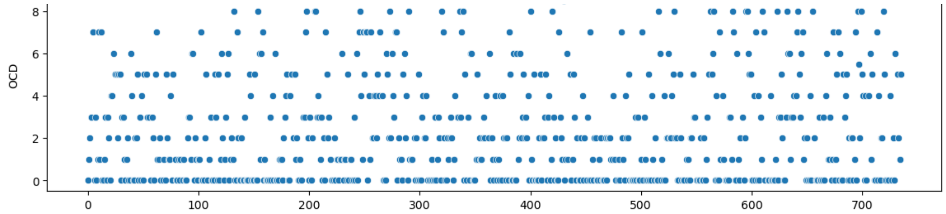
Out[7]:

	Age	Hours per day	BPM	Anxiety	Depression	Insomnia	OCD
0	18.0	3.0	156.0	3.0	0.0	1.0	0.0
1	63.0	1.5	119.0	7.0	2.0	2.0	1.0
2	18.0	4.0	132.0	7.0	7.0	10.0	2.0
3	61.0	2.5	84.0	9.0	7.0	3.0	3.0
4	18.0	4.0	107.0	7.0	2.0	5.0	9.0

In [8]:

```
fig, ax = plt.subplots(7, 1, figsize = (14, 30))

for i, col in enumerate(num_col):
    sns.scatterplot(df1[col], ax = ax[i])
```

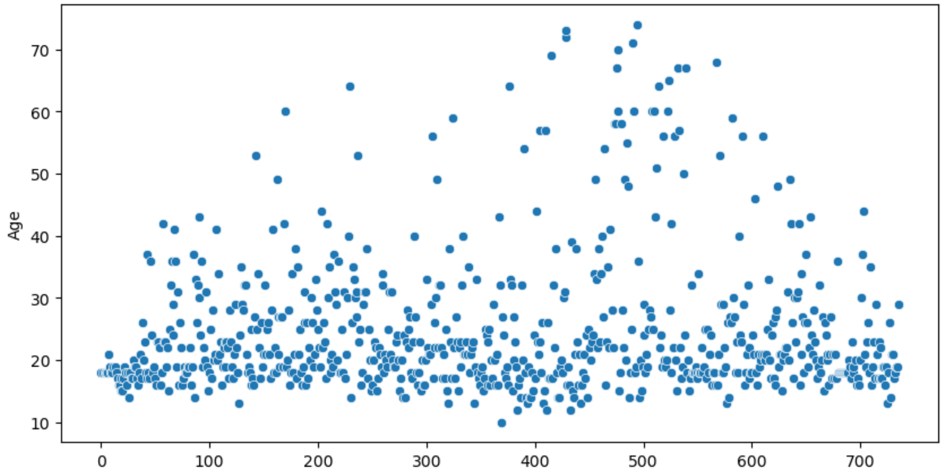

Age
90
80
70
60
50
40
30
20
10
0
100
200
300
400
500
600
700
Hours per day
25
20
15
10
5
0
0
100
200
300
400
500
600
700
1e9
BPM
1.0
0.8
0.6
0.4
0.2
0.0
0
100
200
300
400
500
600
700
Anxiety
10
8
6
4
2
0
0
100
200
300
400
500
600
700
Depression
10
8
6
4
2
0
0
100
200
300
400
500
600
700
Insomnia
10
8
6
4
2
0
0
100
200
300
400
500
600
700
10



In [9]:
```
#we can see outliers in the following columns

#outliers = ['Age', 'Hours per day', 'BPM']
```

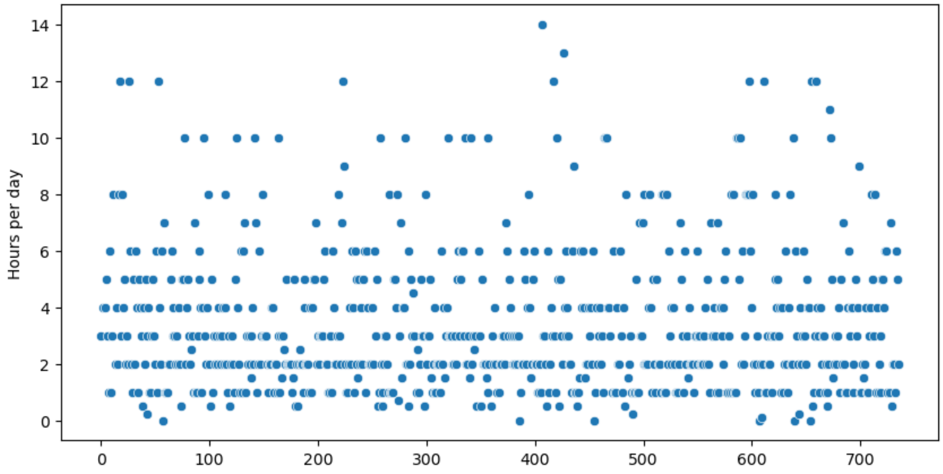
In [10]:
```
#removing the data that is located differently from where the m

df2 = df1[((df1['Age'] < 80) & ~(df1['Age'].isin([61,63])))]
```

In [11]:
```
#checking
plt.figure(figsize=(10,5))
sns.scatterplot(df2['Age']);
```



In [12]:
```
df3 = df2[df2['Hours per day'] < 15]
```

In [13]:
```
#checking
plt.figure(figsize=(10,5))
sns.scatterplot(df3['Hours per day']);
```



In [14]:
```
df3['BPM'].sort_values(ascending = False)
```

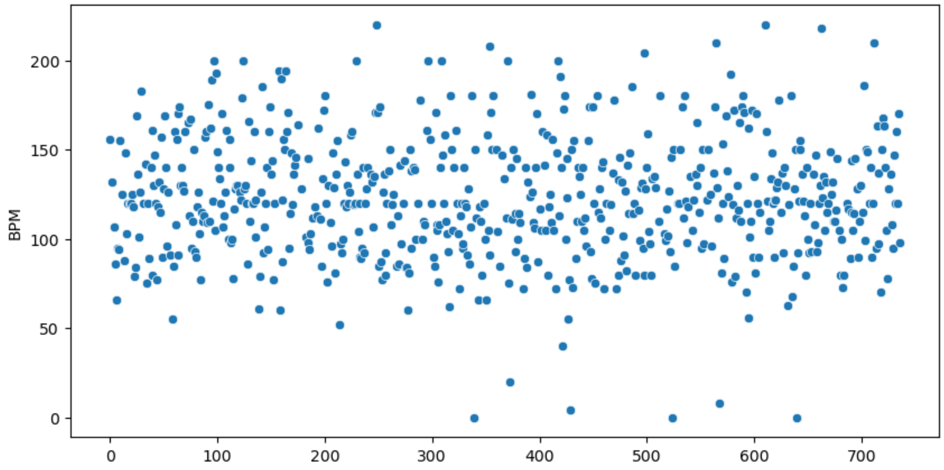
Out[14]:
```
568    999999999.0
644          624.0
610          220.0
248          220.0
662          218.0
          ...
688            NaN
700            NaN
706            NaN
712            NaN
717            NaN
Name: BPM, Length: 721, dtype: float64
```

We can see 2 outliers here for now. lets try to remove those first.

In [15]:
```
df4 = df3[(df3['BPM'] != 999999999.0) & (df3['BPM'] != 624.0)]
```

In [16]:

```
#checking
plt.figure(figsize=(10,5))
sns.scatterplot(df4['BPM']);
```



Lets clear it further

In [17]:

```
len(df4[df4['BPM'] < 40])
```

Out[17]:

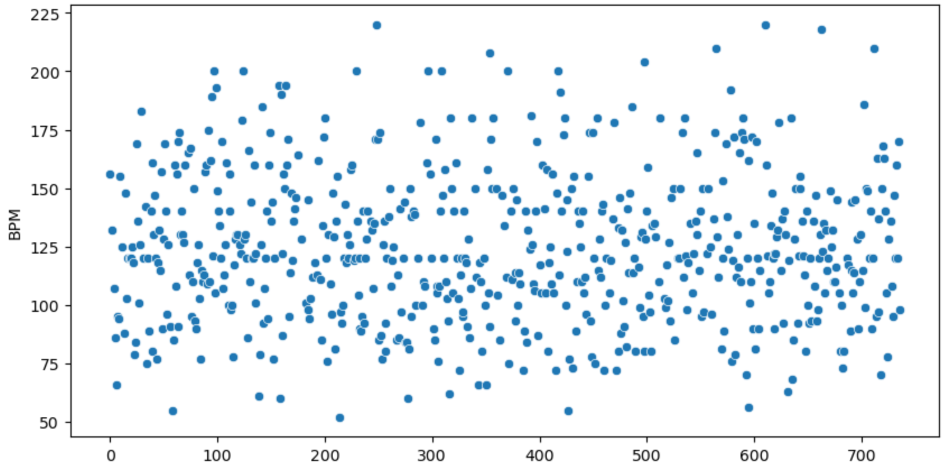
```
6
```

In [18]:

```
#deleting the entries less than 40
df5 = df4[df4['BPM'] > 40]
```

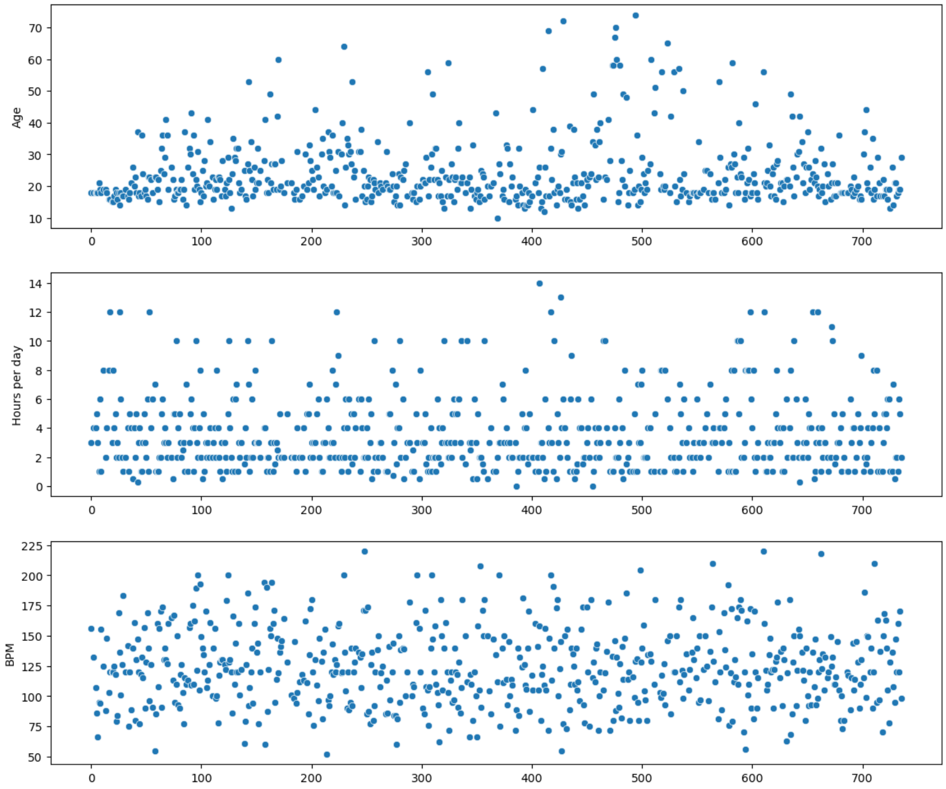
In [19]:

```
#checking
plt.figure(figsize=(10,5))
sns.scatterplot(df5['BPM']);
```



In [20]:

```
fig, ax = plt.subplots(3,1, figsize=(14,12))

outliers = ['Age', 'Hours per day', 'BPM']

for i, c in enumerate(outliers):
    sns.scatterplot(df5[c], ax=ax[i])

plt.show()
```



Handling null values

In [21]:

```
df6 = df5.copy()
```

In [22]:

```
df6.isnull().sum()
```

Out[22]:

```
Timestamp                       0
Age                             0
Primary streaming service       1
Hours per day                   0
While working                   1
Instrumentalist                 3
Composer                        0
Fav genre                       0
Exploratory                     0
Foreign languages               3
BPM                             0
Frequency [Classical]           0
Frequency [Country]             0
Frequency [EDM]                 0
Frequency [Folk]                0
Frequency [Gospel]              0
Frequency [Hip hop]             0
Frequency [Jazz]                0
Frequency [K pop]               0
Frequency [Latin]               0
Frequency [Lofi]                0
Frequency [Metal]               0
Frequency [Pop]                 0
Frequency [R&B]                 0
Frequency [Rap]                 0
Frequency [Rock]                0
Frequency [Video game music]    0
Anxiety                         0
Depression                      0
Insomnia                        0
OCD                             0
Music effects                   4
Permissions                     0
dtype: int64
```

In [23]:

```
#removing the null values

df6.dropna(inplace = True)

df6.isnull().sum()
```

Out[23]:

```
Timestamp                       0
Age                             0
Primary streaming service       0
Hours per day                   0
While working                   0
Instrumentalist                 0
Composer                        0
Fav genre                       0
Exploratory                     0
Foreign languages               0
BPM                             0
Frequency [Classical]           0
Frequency [Country]             0
Frequency [EDM]                 0
Frequency [Folk]                0
Frequency [Gospel]              0
Frequency [Hip hop]             0
Frequency [Jazz]                0
Frequency [K pop]               0
Frequency [Latin]               0
Frequency [Lofi]                0
Frequency [Metal]               0
Frequency [Pop]                 0
Frequency [R&B]                 0
Frequency [Rap]                 0
Frequency [Rock]                0
Frequency [Video game music]    0
Anxiety                         0
Depression                      0
Insomnia                        0
OCD                             0
Music effects                   0
Permissions                     0
dtype: int64
```

In [24]:

```
#removing 'Permissions' column as it has just one value

df6.drop('Permissions', axis = 1, inplace = True)
```

In [25]:
```
df7 = df6.copy()
```

Now, taking care of the Timestamp column.

In [26]:
```
df7['Timestamp']
```

Out[26]:
```
2        8/27/2022 21:28:18
4        8/27/2022 21:54:47
5        8/27/2022 21:56:50
6        8/27/2022 22:00:29
7        8/27/2022 22:18:59
                ...        
731     10/30/2022 14:37:28
732      11/1/2022 22:26:42
733      11/3/2022 23:24:38
734      11/4/2022 17:31:47
735       11/9/2022 1:55:20
Name: Timestamp, Length: 595, dtype: object
```

Since the dates are not very useful to the goal of the project, but the hours of the day might be. So here, I am extracting the time from this column

In [27]:
```
df8 = df7.copy()

df8['Time'] = df8['Timestamp'].str[-8:-6]
df8['Time'].unique()
```

Out[27]:
```
array(['21', '22', '23', ' 0', ' 1', ' 3', ' 4', ' 5', ' 8',
'10', '11',
       '12', '13', '14', '15', '16', '17', '18', '19', '20',
' 2', ' 6',
        ' 7', ' 9'], dtype=object)
```

In [28]:
```
df8['Time'] = df8['Time'].astype(int)
df8.Time.unique()
```

Out[28]:
```
array([21, 22, 23,  0,  1,  3,  4,  5,  8, 10, 11, 12, 13, 1
4, 15, 16, 17,
       18, 19, 20,  2,  6,  7,  9])
```

In [29]:

```
#dropping the original column

df8.drop('Timestamp', axis = 1, inplace = True)
```

In [30]:

```
df8.head()
```

Out[30]:

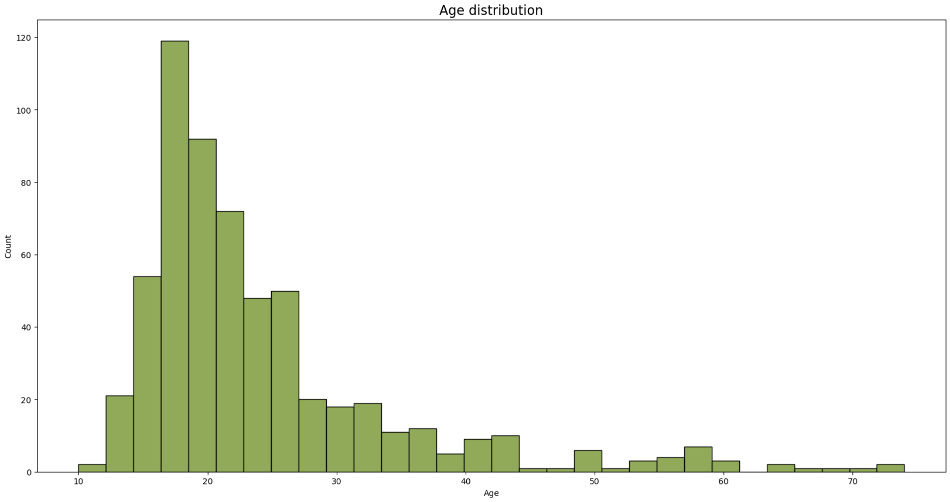
	Age	Primary streaming service	Hours per day	While working	Instrumentalist	Composer	Fav genre	Explo
2	18.0	Spotify	4.0	No	No	No	Video game music	
4	18.0	Spotify	4.0	Yes	No	No	R&B	
5	18.0	Spotify	5.0	Yes	Yes	Yes	Jazz	
6	18.0	YouTube Music	3.0	Yes	Yes	No	Video game music	
7	21.0	Spotify	1.0	Yes	No	No	K pop	

5 rows × 32 columns

EDA

In [31]:

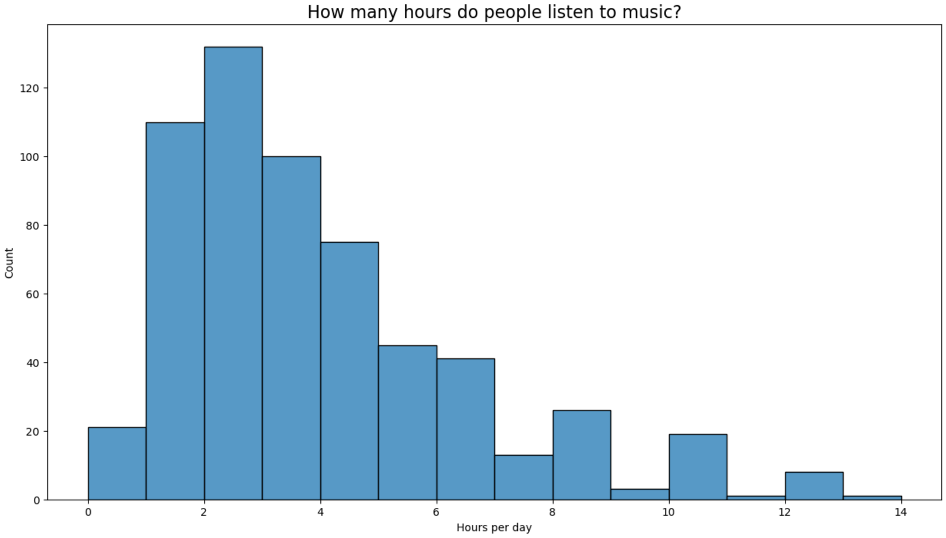
```
plt.figure(figsize=(20,10))
sns.histplot(df8['Age'], color = 'olivedrab')
plt.title("Age distribution", fontsize = 16)
plt.show()
```



Majority of people who have participtaed in this study belong to the age group of 15-25

In [32]:

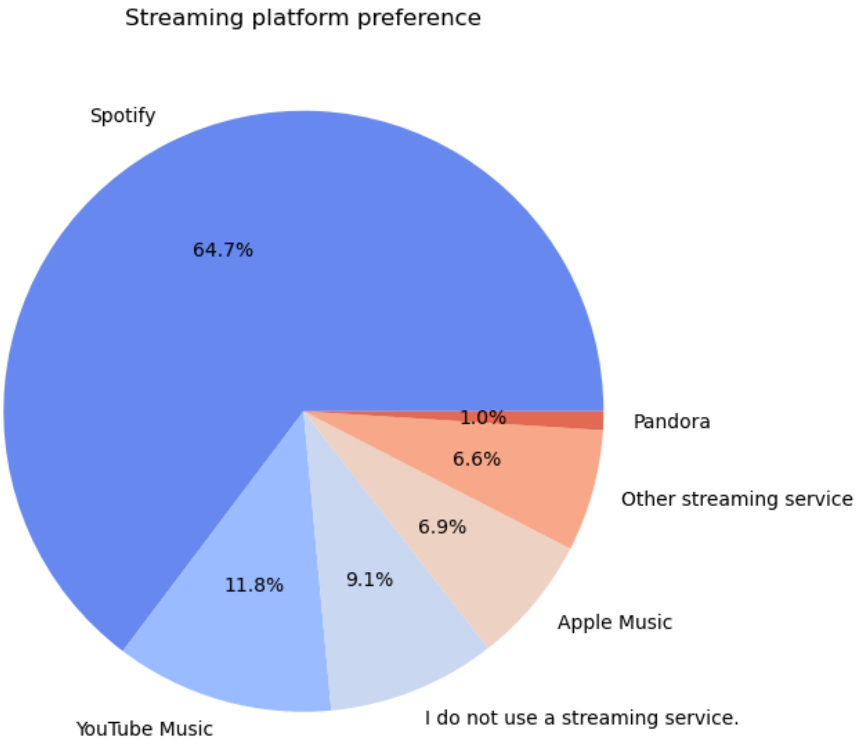
```
plt.figure(figsize=(15,8))
sns.histplot(df8['Hours per day'], bins = 14)
plt.title("How many hours do people listen to music?", fontsize
plt.show()
```



Most people listen to music for 1-4 hours daily, after which the time decreases drastically!

In [33]:

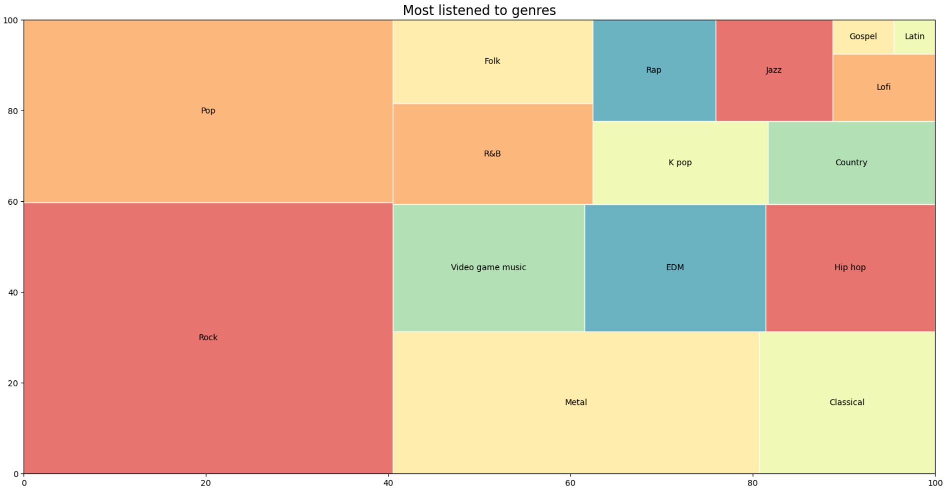
```
plt.figure(figsize = (7,16))
service = df8['Primary streaming service'].value_counts()
plt.pie(service, labels = service.index, colors = sns.color_pal
plt.title("Streaming platform preference")
plt.show()
```



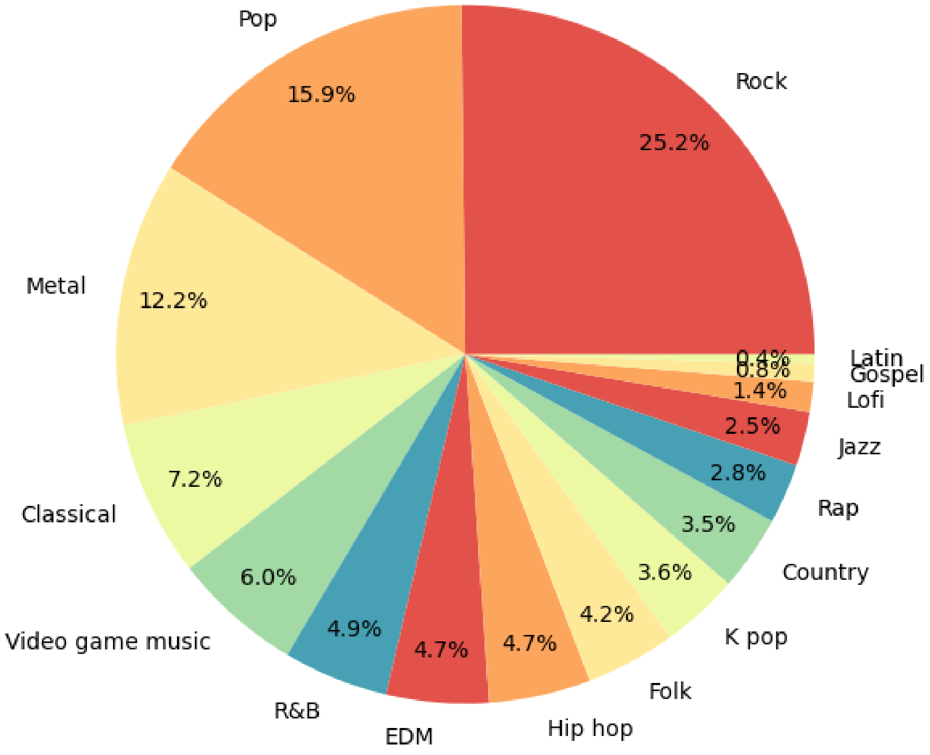
Most people listen to music on Spotify

Now, I want to see how much people listen to different genres of music

In [34]:
```
import squarify
plt.figure(figsize = (20,10))

squarify.plot(df8['Fav genre'].value_counts().values, label = d
              color = sns.color_palette("Spectral"), ec = 'whit

plt.title("Most listened to genres", fontsize = 16);
```



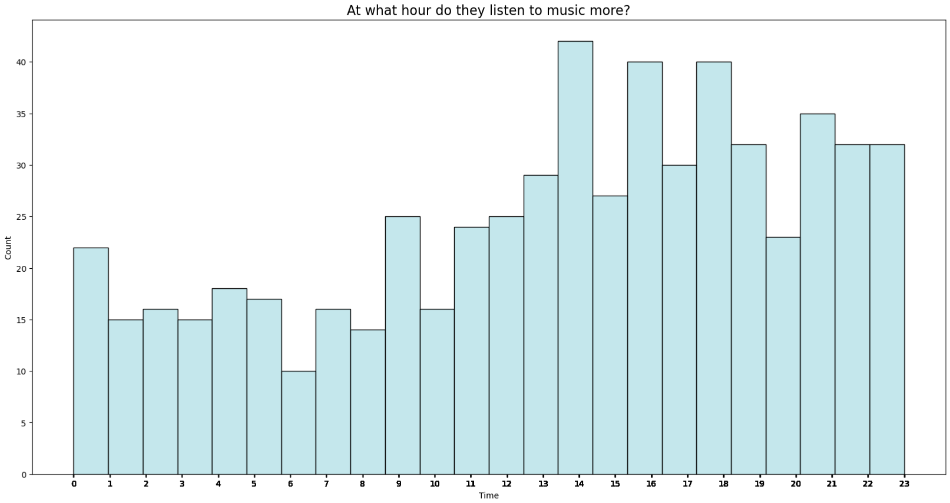
```
In [35]: plt.figure(figsize = (7,18))
         plt.pie(df4['Fav genre'].value_counts(), labels = df4['Fav genr
                pctdistance=0.85, colors = sns.color_palette('Spectral')
```



Rock, Pop and Metal constitutes more than half of the people. On the other hand, Latin and Gospel are listened to by less than 1% of the people.

In [36]:

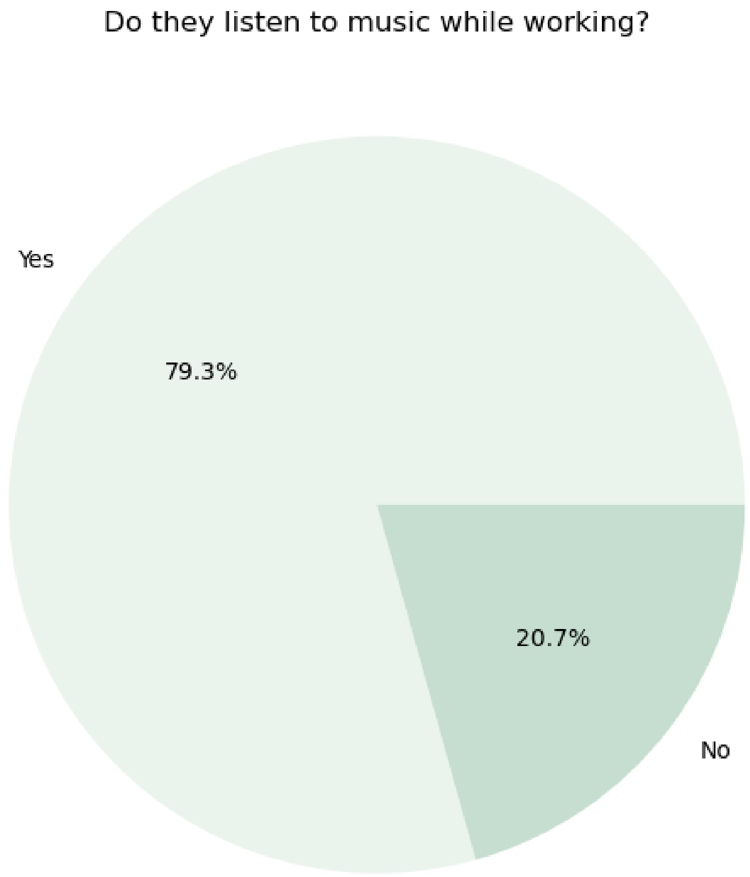
```
labels = list(df8['Time'])

plt.figure(figsize=(20,10))
sns.histplot(df8['Time'], bins = 24, color = 'powderblue')
plt.title("At what hour do they listen to music more?", fontsiz
plt.xticks(labels);
```



Though there isn't a pattern, but we can see an increment towards the second half of the day.

In [37]:

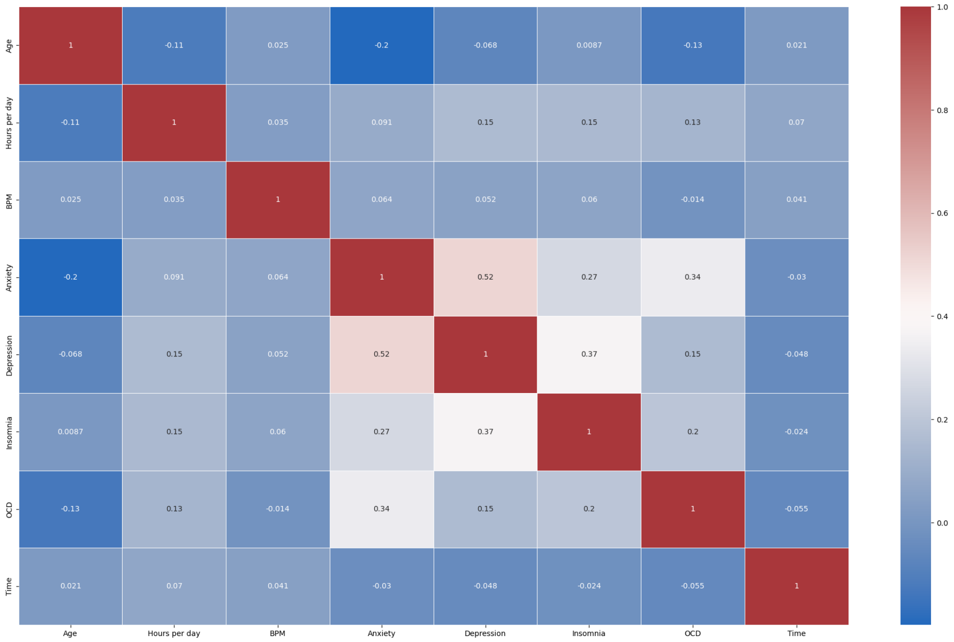
```python
plt.figure(figsize = (7,16))
working = df8['While working'].value_counts()
plt.pie(working, labels = working.index, colors = sns.light_pal
plt.title('Do they listen to music while working?');
```



Checking for correlations in the data

In [38]:

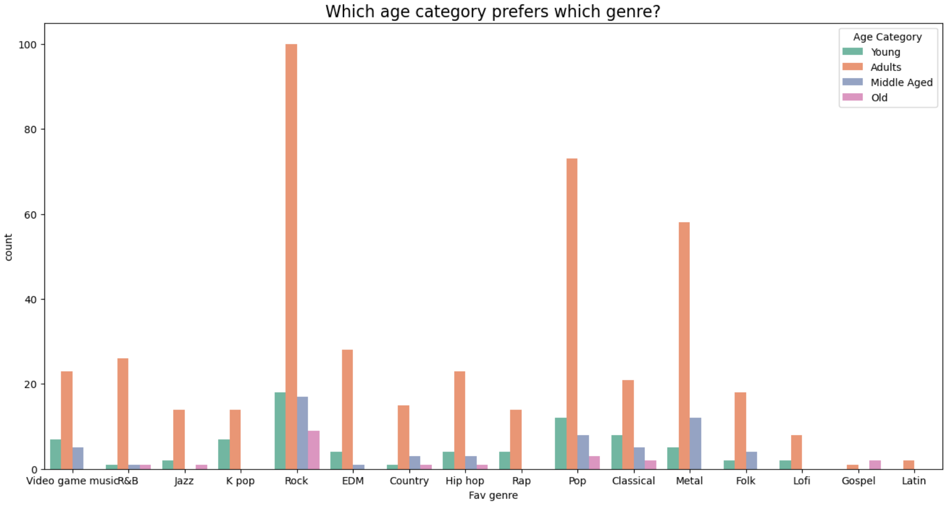
```python
# df4 = df3.copy()

num_col = df8.select_dtypes(include = ['int', 'float'])
mat = num_col.corr()

plt.figure(figsize=(25,15))
cor = sns.heatmap(mat, cmap = sns.color_palette("vlag", as_cmap
```



We can see from here how Anxiety and Depression are correlated with each other

In [39]:

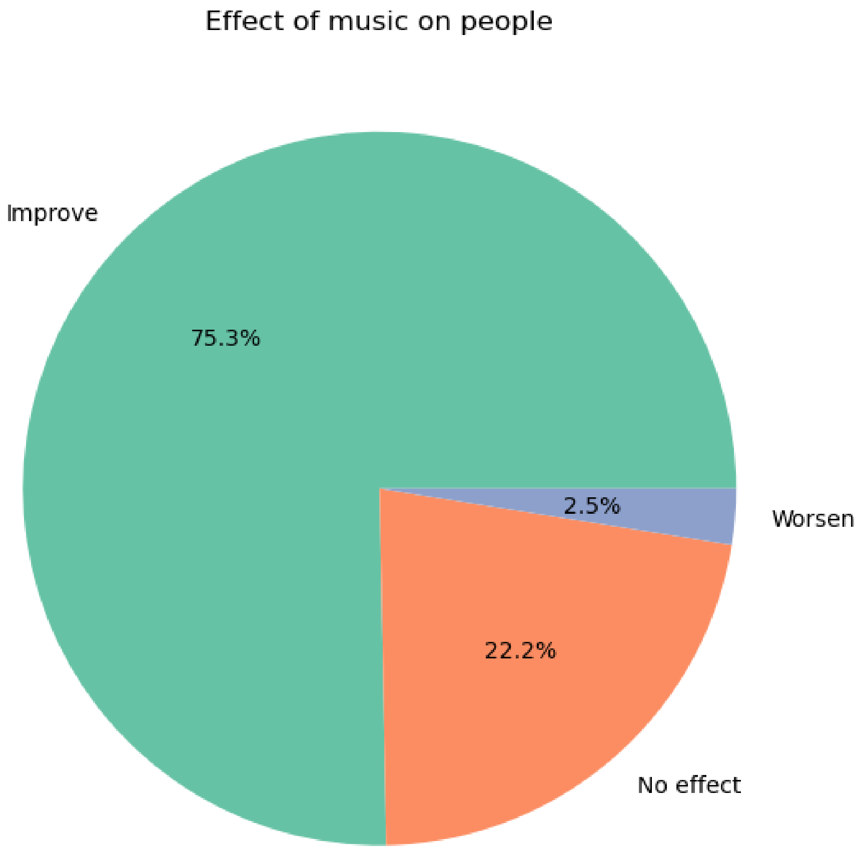
```
# Defining the age bins and labels
age_bins = [0, 17, 34, 54, df4['Age'].max()]
age_labels = ['Young', 'Adults', 'Middle Aged', 'Old']

# Create a new column 'Age Category' based on the bins and labe
df8['Age Category'] = pd.cut(df8['Age'], bins=age_bins, labels=



plt.figure(figsize = (16,8))
sns.countplot(x = df8['Fav genre'], hue = df8['Age Category'],
plt.title("Which age category prefers which genre?", fontsize =
```



- Rock is the most popular music genre, enjoyed most by people belonging to the age group of 18-35 years (Adults).
- Interestingly, Old people enjoy jazz music more than middle aged people.
- Majorly, young and adults are the audiences for K pop music.
- Audiences of Lofi music mostly comprises people aged below 35.
- Gospel music is mostly enjoyed by Old people(above 54 years) and Latin music by adults (18-35 years).

In [40]:

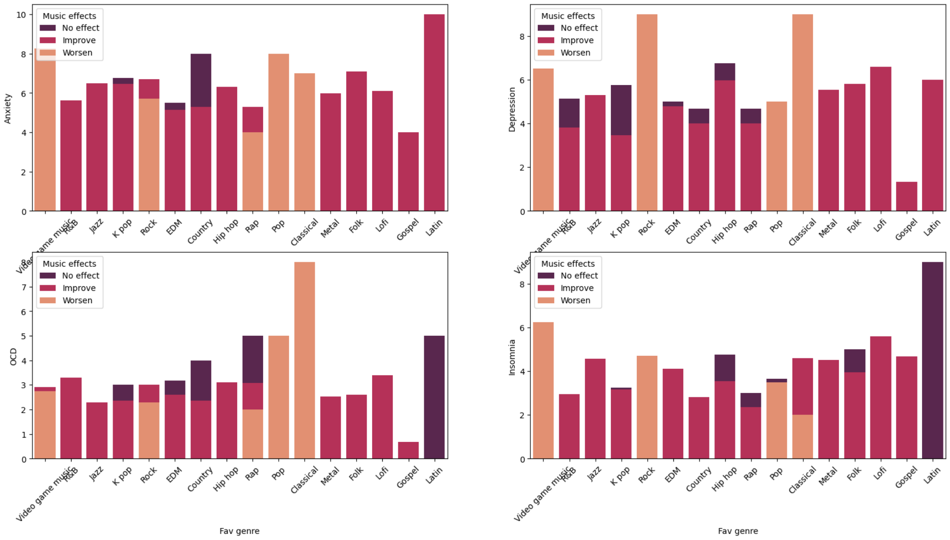
```
plt.figure(figsize = (7,16))
service = df8['Music effects'].value_counts()
plt.pie(service, labels = service.index, colors = sns.color_pal
plt.title("Effect of music on people")
plt.show()
```



As can be seen above, more than 75% of people experience an improvemnt in mood due to music. Let us see in detail the effects music have on people according to their favourite genre and mental health issue.

In [41]:
```
figure, axes = plt.subplots(2, 2, figsize=(20, 10))

plot1 = sns.barplot(x = df8['Fav genre'], y= df8['Anxiety'], hu
            errorbar = None, dodge = False)
plot1.set_xticklabels(plot1.get_xticklabels(), rotation=45)

plot2 = sns.barplot(x = df8['Fav genre'], y= df8['Depression'],
            errorbar = None, dodge = False)
plot2.set_xticklabels(plot1.get_xticklabels(), rotation=45)

plot3 = sns.barplot(x = df8['Fav genre'], y= df8['OCD'], hue =
            errorbar = None, dodge = False)
plot3.set_xticklabels(plot1.get_xticklabels(), rotation=45)

plot4 = sns.barplot(x = df8['Fav genre'], y= df8['Insomnia'], h
            errorbar = None, dodge = False)
plot4.set_xticklabels(plot1.get_xticklabels(), rotation=45)

figure.suptitle("Favourite genre of different age groups accord

plt.show()
```

Favourite genre of different age groups according to their mental health

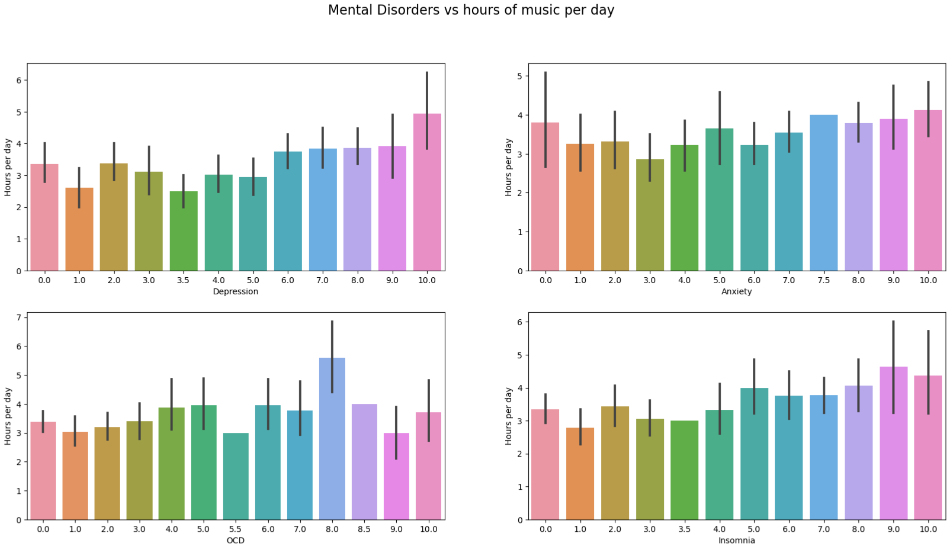


It can be concluded that generally music is shown to have a positive affect on the listener. In addition to that,

- 'Video game music', 'Pop music' and 'Classical music' generally have a negative affect on the moods of the listener.
- 'Latin music' is seen to improve the mood for people with Anxiety and depression and has no affect on people with OCD and insomnia

In [42]:

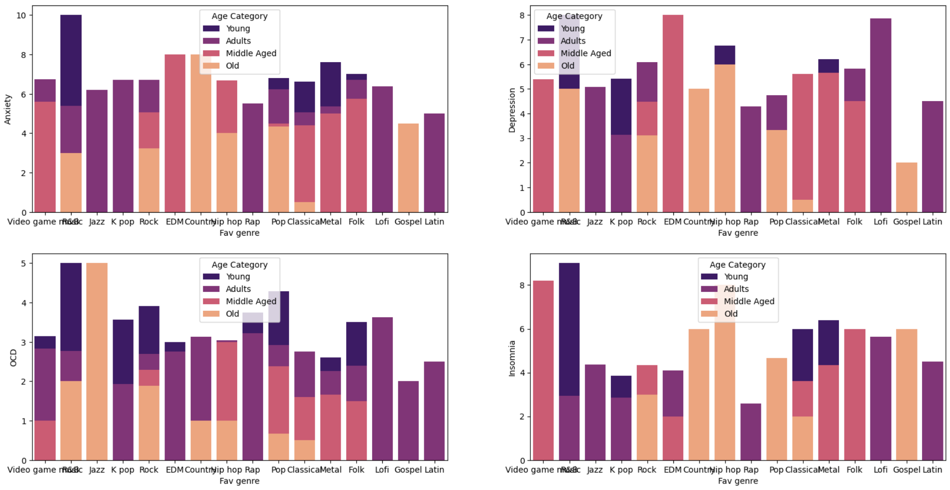
```
figure, axes = plt.subplots(2, 2, figsize=(20, 10))

sns.barplot(data=df8, x='Depression', y='Hours per day', ax=axe
sns.barplot(data=df8, x='Anxiety', y='Hours per day', ax=axes[0
sns.barplot(data=df8, x='OCD', y='Hours per day', ax=axes[1,0])
sns.barplot(data=df8, x='Insomnia', y='Hours per day', ax=axes[

figure.suptitle("Mental Disorders vs hours of music per day", f

plt.show()
```



We can see a relatively consistent duration of music listening in people with different levels of Anxiety. And it can be seen here that people with higher (self-reported) Depression, OCD and Insomnia tend to listen to music more.

In [43]:
```
figure, axes = plt.subplots(2, 2, figsize=(20, 10))

sns.barplot(x = df8['Fav genre'], y= df8['Anxiety'], hue = df8[
            errorbar = None, dodge = False)
sns.barplot(x = df8['Fav genre'], y= df8['Depression'], hue = d
            errorbar = None, dodge = False)
sns.barplot(x = df8['Fav genre'], y= df8['OCD'], hue = df8['Age
            errorbar = None, dodge = False)
sns.barplot(x = df8['Fav genre'], y= df8['Insomnia'], hue = df8
            errorbar = None, dodge = False)

figure.suptitle("Favourite genre of different age groups accord

plt.show()
```

Favourite genre of different age groups according to their mental health



It can be concluded that:

- R&B is preferred among all 4 mental health issues
- Other than that, people with depression enjoy EDM and Lofi as well
- People with OCD prefers jazz and pop
- Old people with insomnia are inclined towards Country, Hip hop and Gospel