CALL LIBRARIES AND THEIR DATA

In [46]:
```
import seaborn as sns
sns.get_dataset_names()
```

Out[46]:
```
['anagrams',
 'anscombe',
 'attention',
 'brain_networks',
 'car_crashes',
 'diamonds',
 'dots',
 'dowjones',
 'exercise',
 'flights',
 'fmri',
 'geyser',
 'glue',
 'healthexp',
 'iris',
 'mpg',
 'penguins',
 'planets',
 'seaice',
 'taxis',
 'tips',
 'titanic']
```

1-How to Import Libraries?

In [2]:
```
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
```

How we Read the Data?

In [3]:
```
music = pd.read_csv("mxmh_survey_results.csv")
```

3-How we Call the Data?

In [5]:
```
music.head(10)
```

Out[5]:

	Timestamp	Age	Primary streaming service	Hours per day	While working	Instrumentalist	Composer	Fav genre	Exploratory	F lang
0	8/27/2022 19:29:02	18.0	Spotify	3.0	Yes	Yes	Yes	Latin	Yes	
1	8/27/2022 19:57:31	63.0	Pandora	1.5	Yes	No	No	Rock	Yes	
2	8/27/2022 21:28:18	18.0	Spotify	4.0	No	No	No	Video game music	No	
3	8/27/2022 21:40:40	61.0	YouTube Music	2.5	Yes	No	Yes	Jazz	Yes	
4	8/27/2022 21:54:47	18.0	Spotify	4.0	Yes	No	No	R&B	Yes	
5	8/27/2022 21:56:50	18.0	Spotify	5.0	Yes	Yes	Yes	Jazz	Yes	
6	8/27/2022 22:00:29	18.0	YouTube Music	3.0	Yes	Yes	No	Video game music	Yes	
7	8/27/2022 22:18:59	21.0	Spotify	1.0	Yes	No	No	K pop	Yes	
8	8/27/2022 22:33:05	19.0	Spotify	6.0	Yes	No	No	Rock	No	
9	8/27/2022 22:44:03	18.0	I do not use a streaming service.	1.0	Yes	No	No	R&B	Yes	

10 rows × 33 columns

4-Define the shape of Data?

In [7]:
```
music.shape
```

Out[7]:
```
(736, 33)
```

5-How to get whole data Information?

In [8]:
```
music.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 736 entries, 0 to 735
Data columns (total 33 columns):
 #   Column                        Non-Null Count  Dtype  
---  ------                        --------------  -----  
 0   Timestamp                     736 non-null    object 
 1   Age                           735 non-null    float64
 2   Primary streaming service     735 non-null    object 
 3   Hours per day                 736 non-null    float64
 4   While working                 733 non-null    object 
 5   Instrumentalist               732 non-null    object 
 6   Composer                      735 non-null    object 
 7   Fav genre                     736 non-null    object 
 8   Exploratory                   736 non-null    object 
 9   Foreign languages             732 non-null    object 
 10  BPM                           629 non-null    float64
 11  Frequency [Classical]         736 non-null    object 
 12  Frequency [Country]           736 non-null    object 
 13  Frequency [EDM]               736 non-null    object 
 14  Frequency [Folk]              736 non-null    object 
 15  Frequency [Gospel]            736 non-null    object 
 16  Frequency [Hip hop]           736 non-null    object 
 17  Frequency [Jazz]              736 non-null    object 
 18  Frequency [K pop]             736 non-null    object 
 19  Frequency [Latin]             736 non-null    object 
 20  Frequency [Lofi]              736 non-null    object 
 21  Frequency [Metal]             736 non-null    object 
 22  Frequency [Pop]               736 non-null    object 
 23  Frequency [R&B]               736 non-null    object 
 24  Frequency [Rap]               736 non-null    object 
 25  Frequency [Rock]              736 non-null    object 
 26  Frequency [Video game music]  736 non-null    object 
 27  Anxiety                       736 non-null    float64
 28  Depression                    736 non-null    float64
 29  Insomnia                      736 non-null    float64
 30  OCD                           736 non-null    float64
 31  Music effects                 728 non-null    object 
 32  Permissions                   736 non-null    object 
dtypes: float64(7), object(26)
memory usage: 189.9+ KB
```

6-How to describe Data?

In [11]:
```
music.describe()
```

Out[11]:

	Age	Hours per day	BPM	Anxiety	Depression	Insomnia	OCD
count	735.000000	736.000000	6.290000e+02	736.000000	736.000000	736.000000	736.000000
mean	25.206803	3.572758	1.589948e+06	5.837636	4.796196	3.738451	2.637228
std	12.054970	3.028199	3.987261e+07	2.793054	3.028870	3.088689	2.842017
min	10.000000	0.000000	0.000000e+00	0.000000	0.000000	0.000000	0.000000
25%	18.000000	2.000000	1.000000e+02	4.000000	2.000000	1.000000	0.000000
50%	21.000000	3.000000	1.200000e+02	6.000000	5.000000	3.000000	2.000000
75%	28.000000	5.000000	1.440000e+02	8.000000	7.000000	6.000000	5.000000
max	89.000000	24.000000	1.000000e+09	10.000000	10.000000	10.000000	10.000000

7-How many number of unique values in a Data?

In [12]:
```
music.nunique()
```

Out[12]:
```
Timestamp                       735
Age                              61
Primary streaming service         6
Hours per day                    27
While working                     2
Instrumentalist                   2
Composer                          2
Fav genre                        16
Exploratory                       2
Foreign languages                 2
BPM                             135
Frequency [Classical]             4
Frequency [Country]               4
Frequency [EDM]                   4
Frequency [Folk]                  4
Frequency [Gospel]                4
Frequency [Hip hop]               4
Frequency [Jazz]                  4
Frequency [K pop]                 4
Frequency [Latin]                 4
Frequency [Lofi]                  4
Frequency [Metal]                 4
Frequency [Pop]                   4
Frequency [R&B]                   4
Frequency [Rap]                   4
Frequency [Rock]                  4
Frequency [Video game music]      4
Anxiety                          12
Depression                       12
Insomnia                         12
OCD                              13
Music effects                     3
Permissions                       1
dtype: int64
```

8-How we do Data cleaning by using pandas and numpy?

Data is already cleaned so there is no need for data cleaning.

PLOTTING

9-Count Plot

NOTE: In count plot there is only one numerical X variable we use.

Import libraries:

In [30]:
```
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
```

Read Data:

In [ ]:
```
music = pd.read_csv("mxmh_survey_results.csv")
```

Read Data of first 10 values:

In [31]:
```
music.head(10)
```

Out[31]:

	Timestamp	Age	Primary streaming service	Hours per day	While working	Instrumentalist	Composer	Fav genre	Exploratory	F lang
0	8/27/2022 19:29:02	18.0	Spotify	3.0	Yes	Yes	Yes	Latin	Yes	
1	8/27/2022 19:57:31	63.0	Pandora	1.5	Yes	No	No	Rock	Yes	
2	8/27/2022 21:28:18	18.0	Spotify	4.0	No	No	No	Video game music	No	
3	8/27/2022 21:40:40	61.0	YouTube Music	2.5	Yes	No	Yes	Jazz	Yes	
4	8/27/2022 21:54:47	18.0	Spotify	4.0	Yes	No	No	R&B	Yes	
5	8/27/2022 21:56:50	18.0	Spotify	5.0	Yes	Yes	Yes	Jazz	Yes	
6	8/27/2022 22:00:29	18.0	YouTube Music	3.0	Yes	Yes	No	Video game music	Yes	
7	8/27/2022 22:18:59	21.0	Spotify	1.0	Yes	No	No	K pop	Yes	
8	8/27/2022 22:33:05	19.0	Spotify	6.0	Yes	No	No	Rock	No	
9	8/27/2022 22:44:03	18.0	I do not use a streaming service.	1.0	Yes	No	No	R&B	Yes	

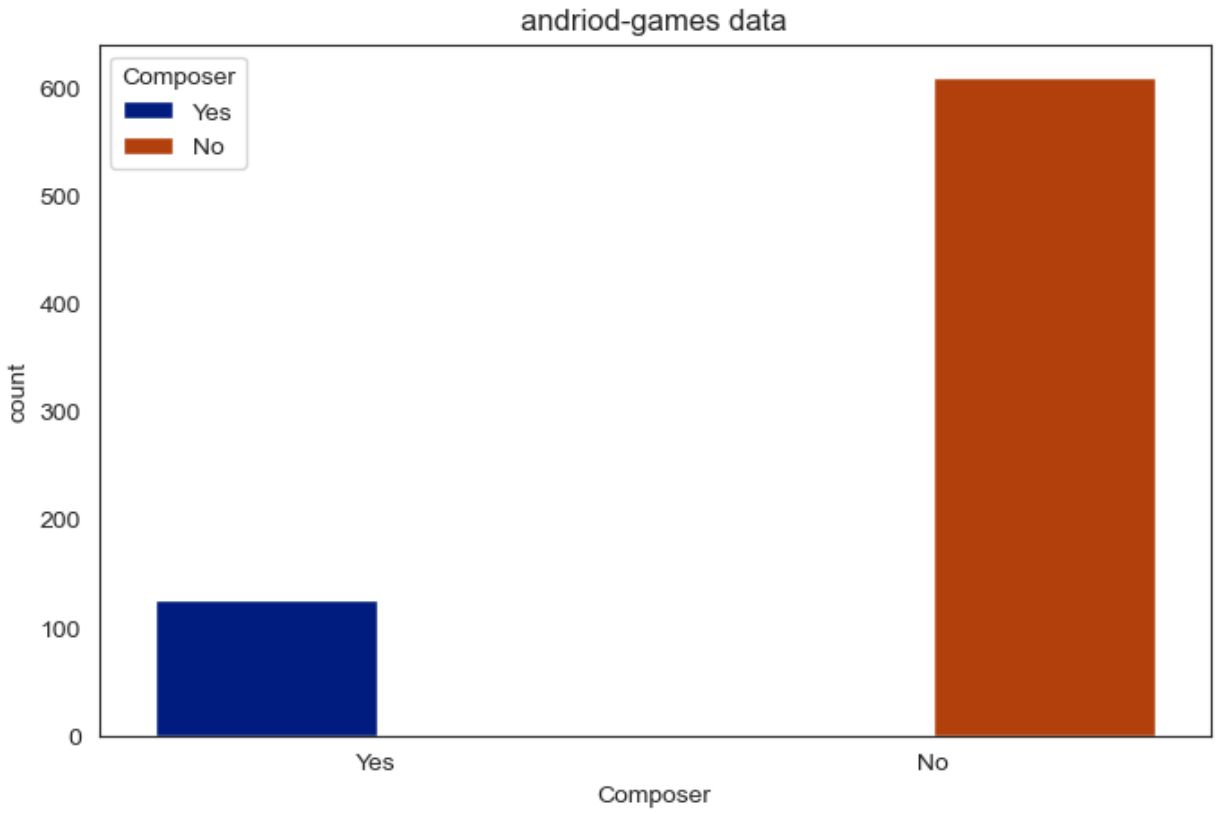
10 rows × 33 columns

Set Style:

In [33]:
```
sns.set_style("white")
```

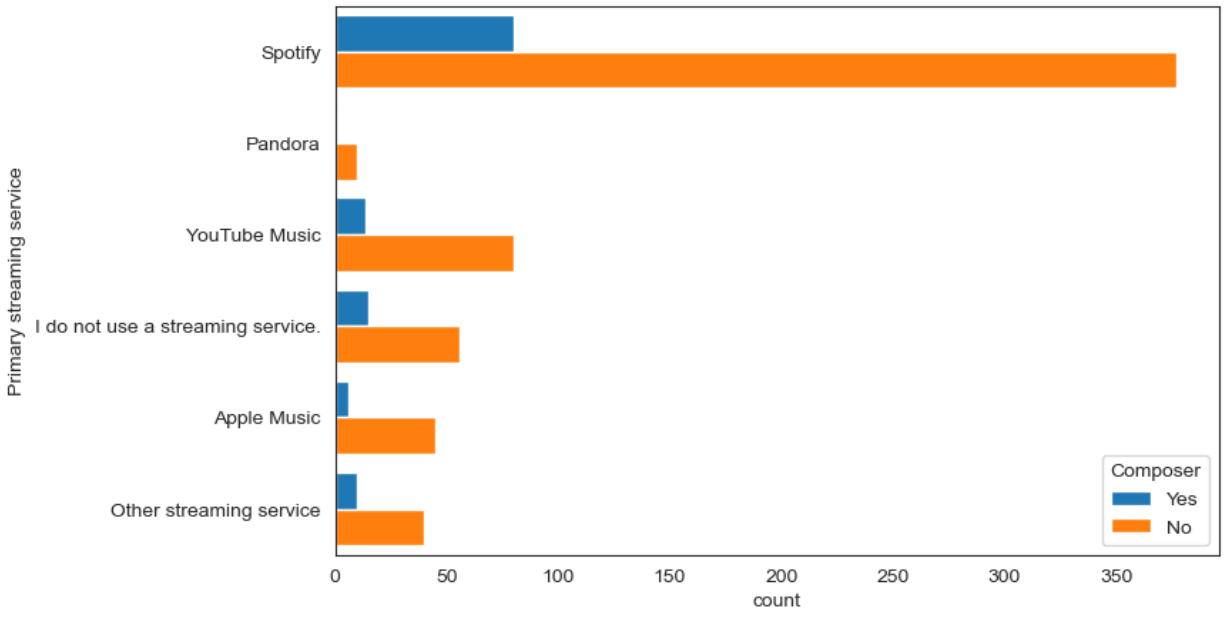
Draw plot:

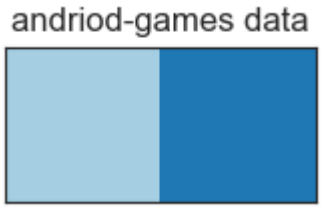
In [34]:
```
# set figure size
plt.figure(figsize=(8,5))

p=sns.countplot(x="Composer", data=music, hue="Composer", saturation=4, palette='dark'
plt.title("andriod-games data")
plt.show()
```



How we select random color & Horizental count plot:

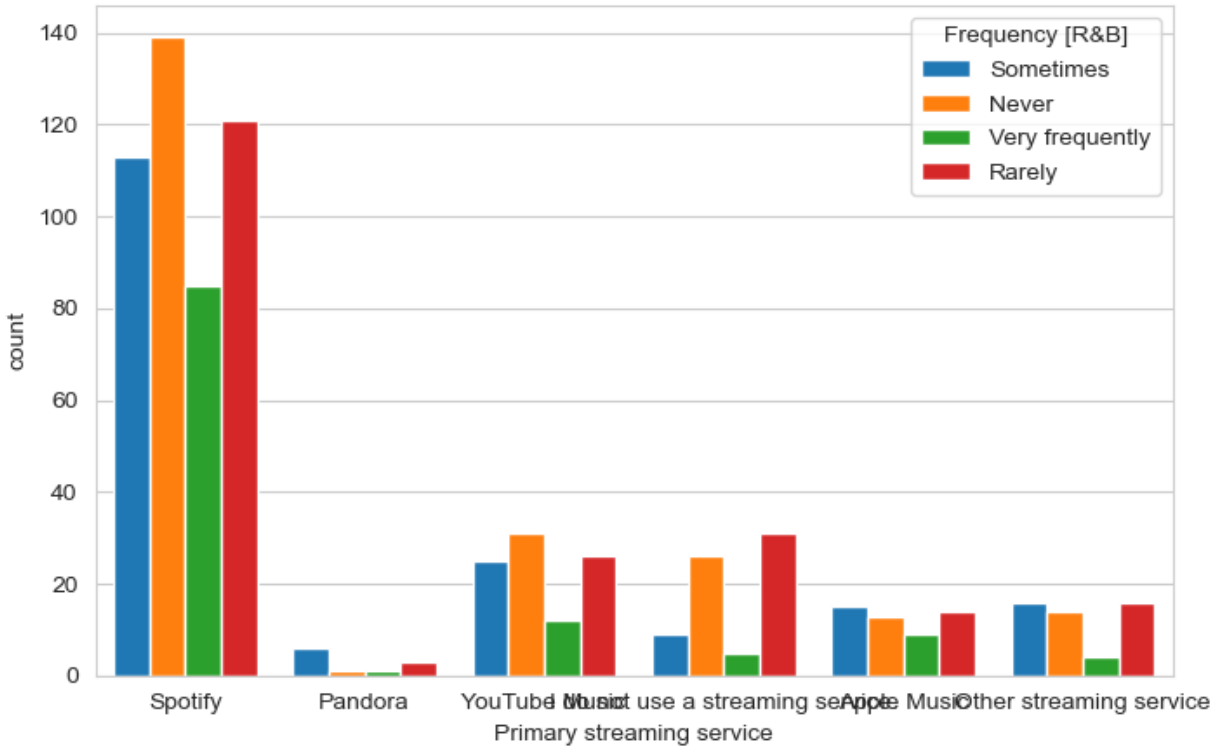
In [35]:
```
plt.figure(figsize=(8,5))

p=sns.countplot(y="Primary streaming service", data=music, hue="Composer", saturation=
custom_palette = sns.color_palette("Paired", 2)
sns.palplot(custom_palette)

plt.title("andriod-games data")
plt.show()
```





Multiple variables in Countplot & with different style:

In [37]:
```
sns.set_style("whitegrid")
music = pd.read_csv("mxmh_survey_results.csv")

plt.figure(figsize=(8,5))

p=sns.countplot(x="Primary streaming service", data=music, hue="Frequency [R&B]", satu
plt.show()
```



10-Box/Boxen Plot.

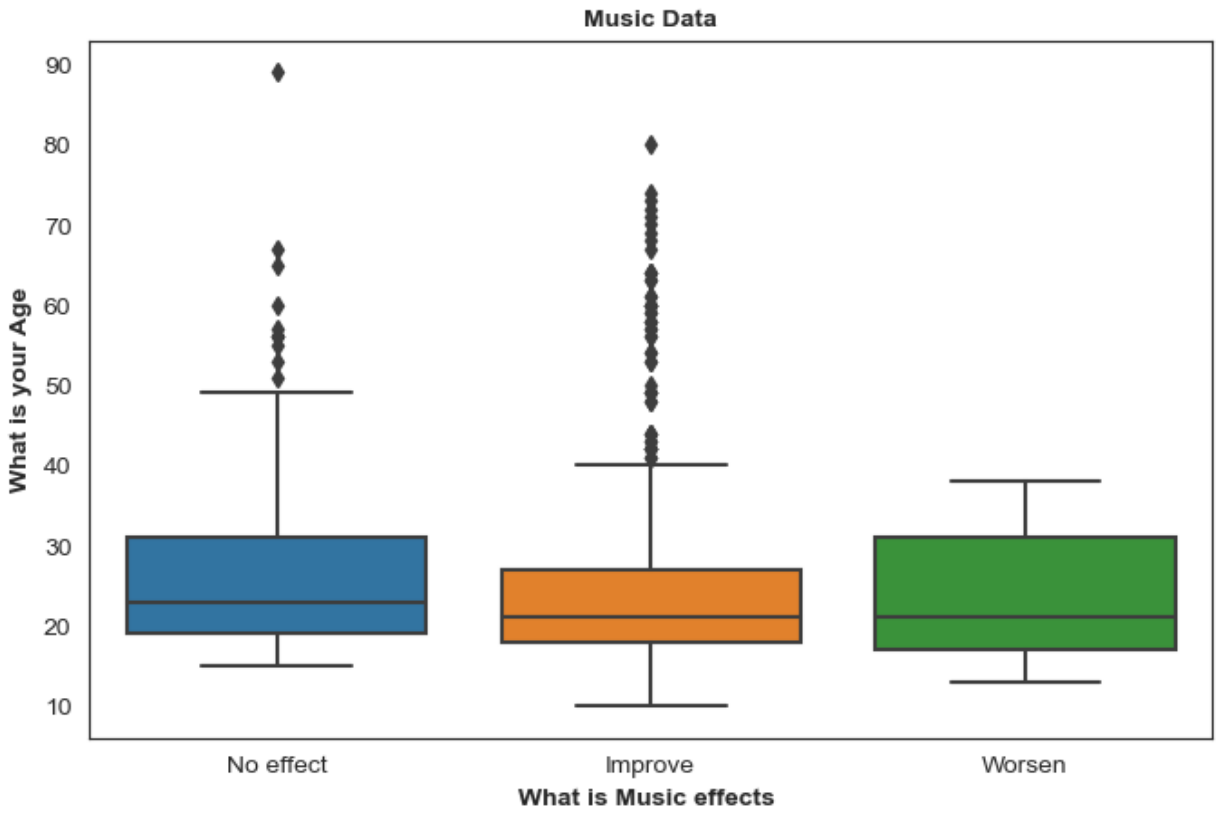
NOTE: In box & Boxen plot we use two X & Y variable may contain one numeric and one cetagorical data. (Quartile, outliyers, median, positive skewed, negative skewed)

Set Style:

In [38]:
```
sns.set_style("white")
```

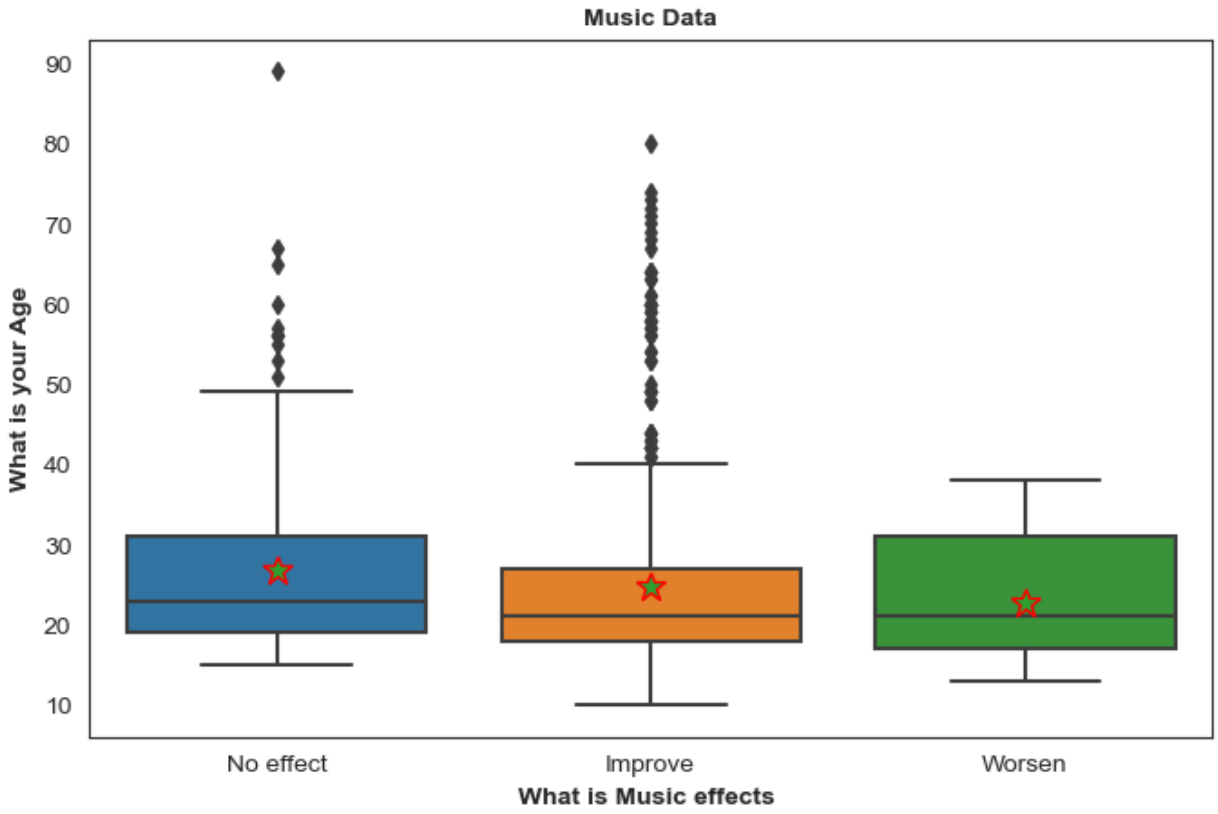
Simple Box Plot with labels:

In [42]:
```
plt.figure(figsize=(8,5))
sns.boxplot(x="Music effects", y="Age", data=music)
plt.xlabel("What is Music effects ", size=10, weight='bold')
plt.ylabel("What is your Age", size=10, weight='bold')
plt.title("Music Data", size=10, weight='bold')
plt.show()
```



Box Plot with Mean:

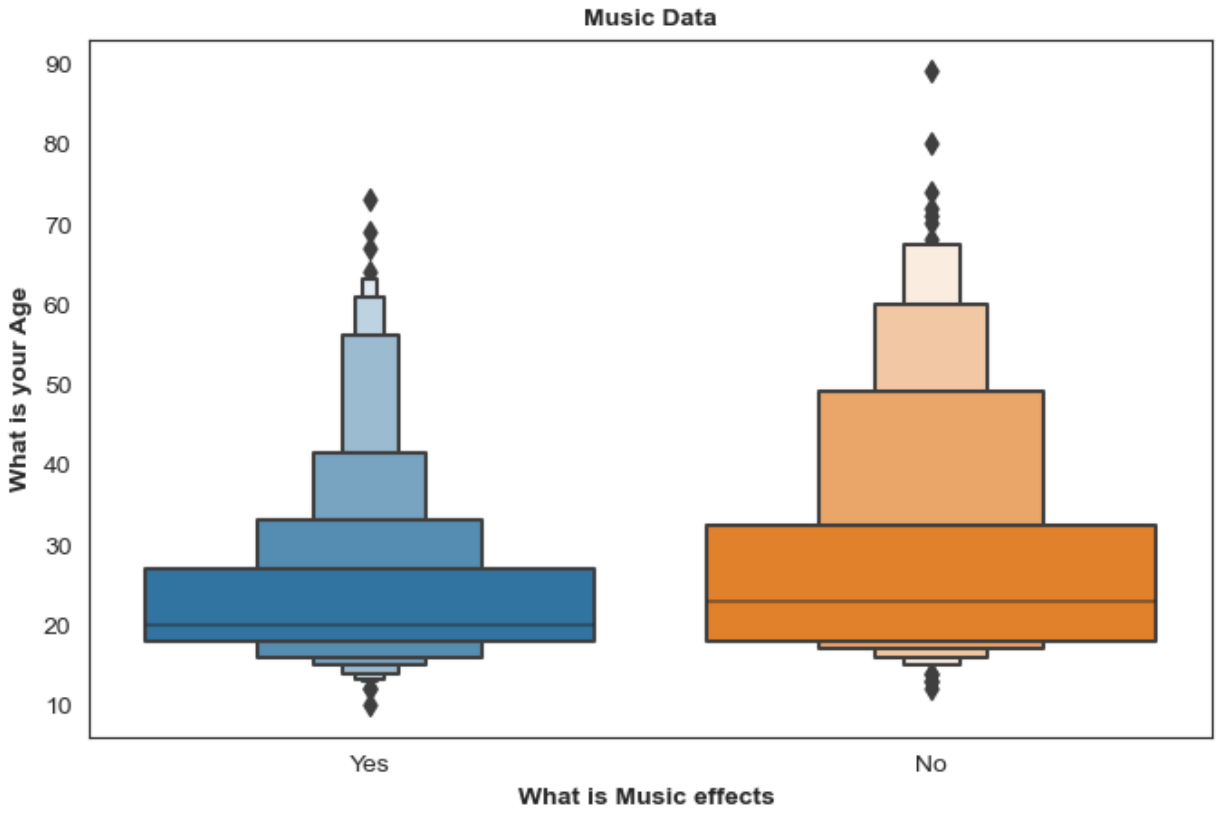
In [44]:
```
plt.figure(figsize=(8,5))
sns.boxplot(x="Music effects", y="Age", data=music, showmeans=True, meanprops= {"marke
plt.xlabel("What is Music effects ", size=10, weight='bold')
plt.ylabel("What is your Age", size=10, weight='bold')
plt.title("Music Data", size=10, weight='bold')
plt.show()
```



In [ ]:

BOXEN PLOT with 2 variable:

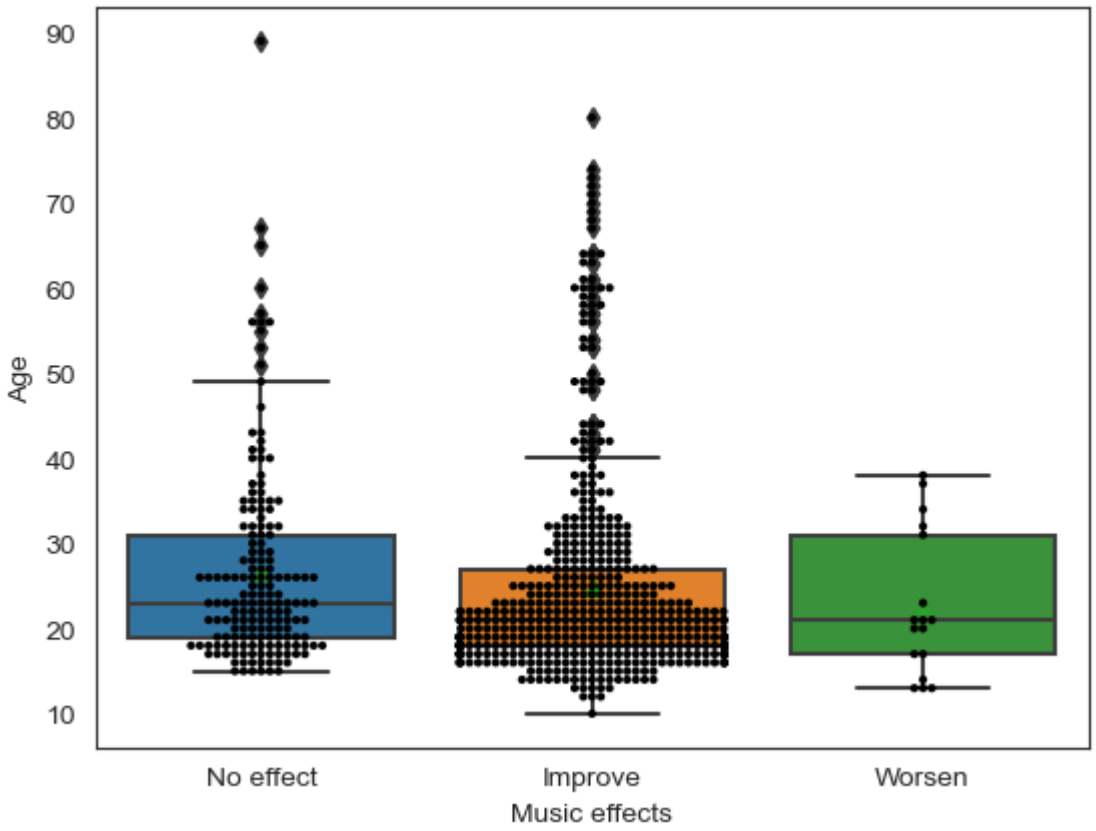
In [48]:
```
plt.figure(figsize=(8,5))
sns.boxenplot(x="Exploratory", y="Age", data=music)
plt.xlabel("What is Music effects ", size=10, weight='bold')
plt.ylabel("What is your Age", size=10, weight='bold')
plt.title("Music Data", size=10, weight='bold')
plt.show()
```



Boxplot with swarmplot:

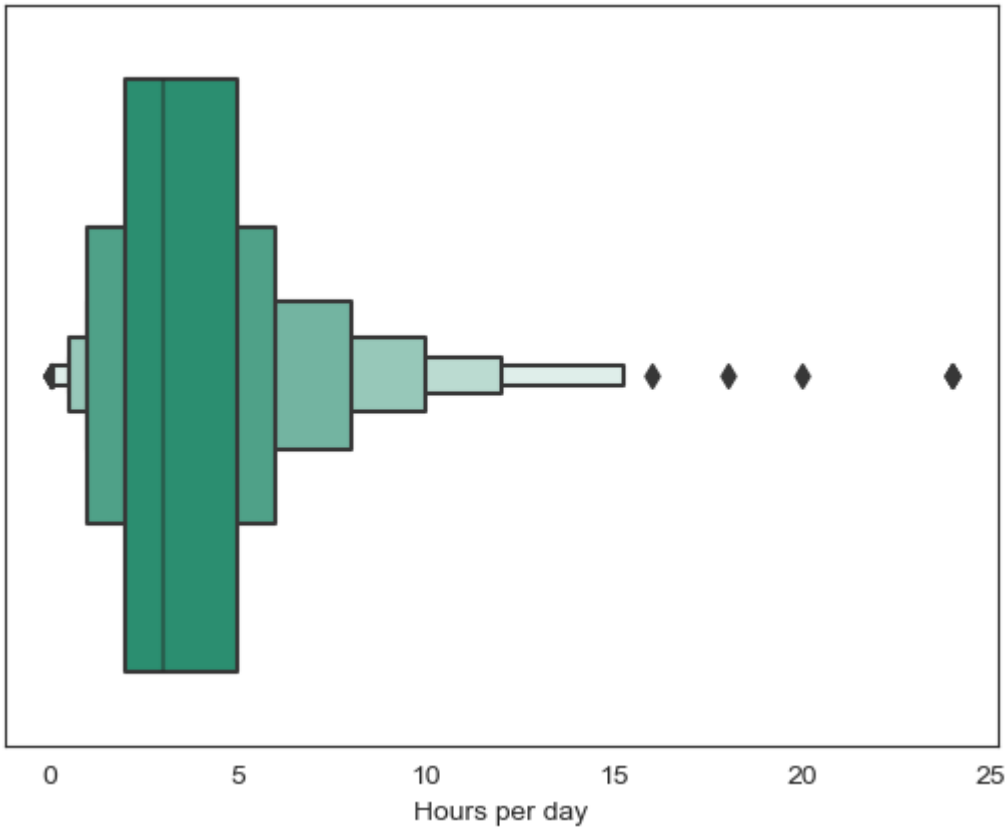
In [56]:
```
sns.boxplot(data =music, x='Music effects',y='Age',showmeans= True,)
sns.swarmplot(data =music, x='Music effects',y='Age', size=3, color='black')
plt.show()
```

```
E:\New folder\lib\site-packages\seaborn\categorical.py:1296: UserWarning: 14.4% of th
e points cannot be placed; you may want to decrease the size of the markers or use st
ripplot.
  warnings.warn(msg, UserWarning)
```



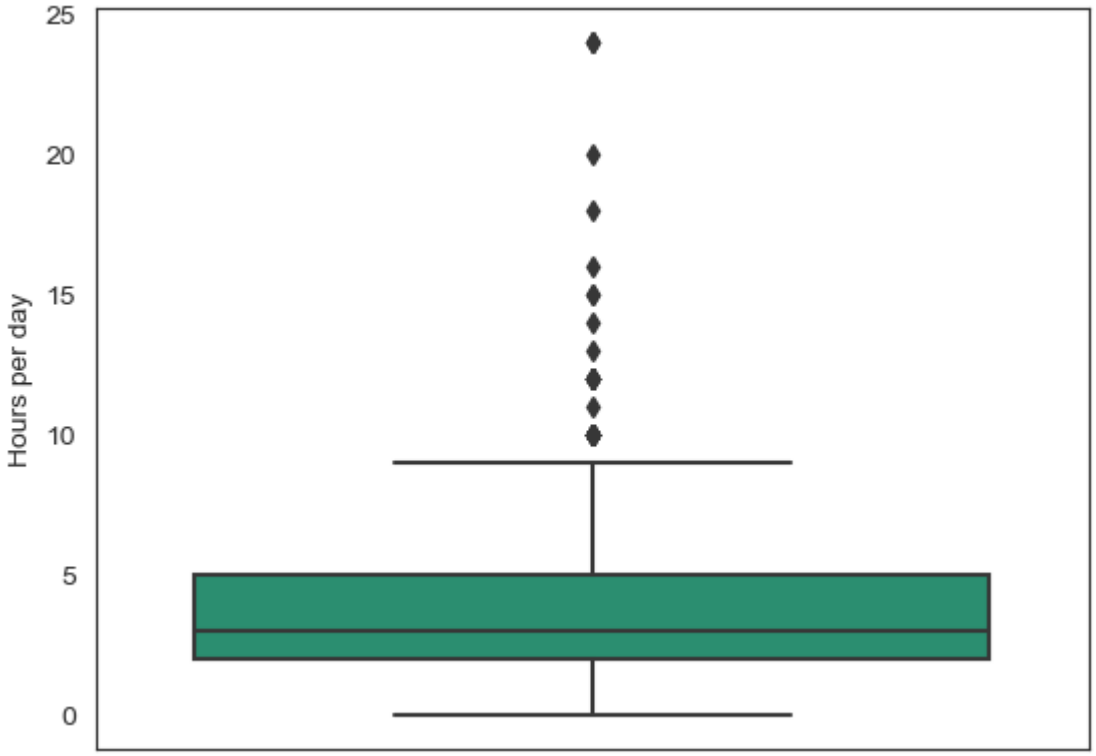
Single Boxen Plot:

In [60]:
```
sns.boxenplot(x="Hours per day",data=music, palette='Dark2')
plt.show()
```



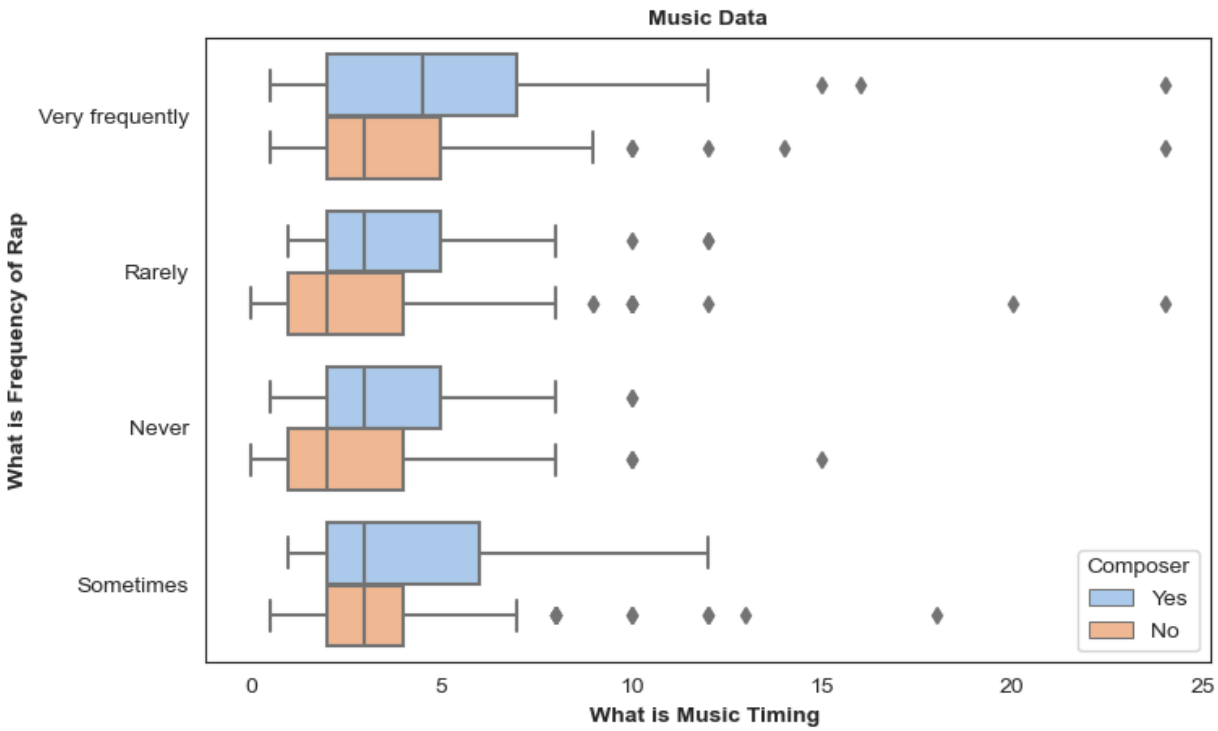
Single Box Plot:

In [62]:
```
sns.boxplot(y="Hours per day",data=music, palette='Dark2')
plt.show()
```



Boxplot with Dodge:

In [67]:
```
plt.figure(figsize=(8,5))
p=sns.boxplot(x="Hours per day", y="Frequency [Rap]", hue="Composer", data=music, pale
plt.xlabel("What is Music Timing ", size=10, weight='bold')
plt.ylabel("What is Frequency of Rap", size=10, weight='bold')
plt.title("Music Data", size=10, weight='bold')
plt.show()
```

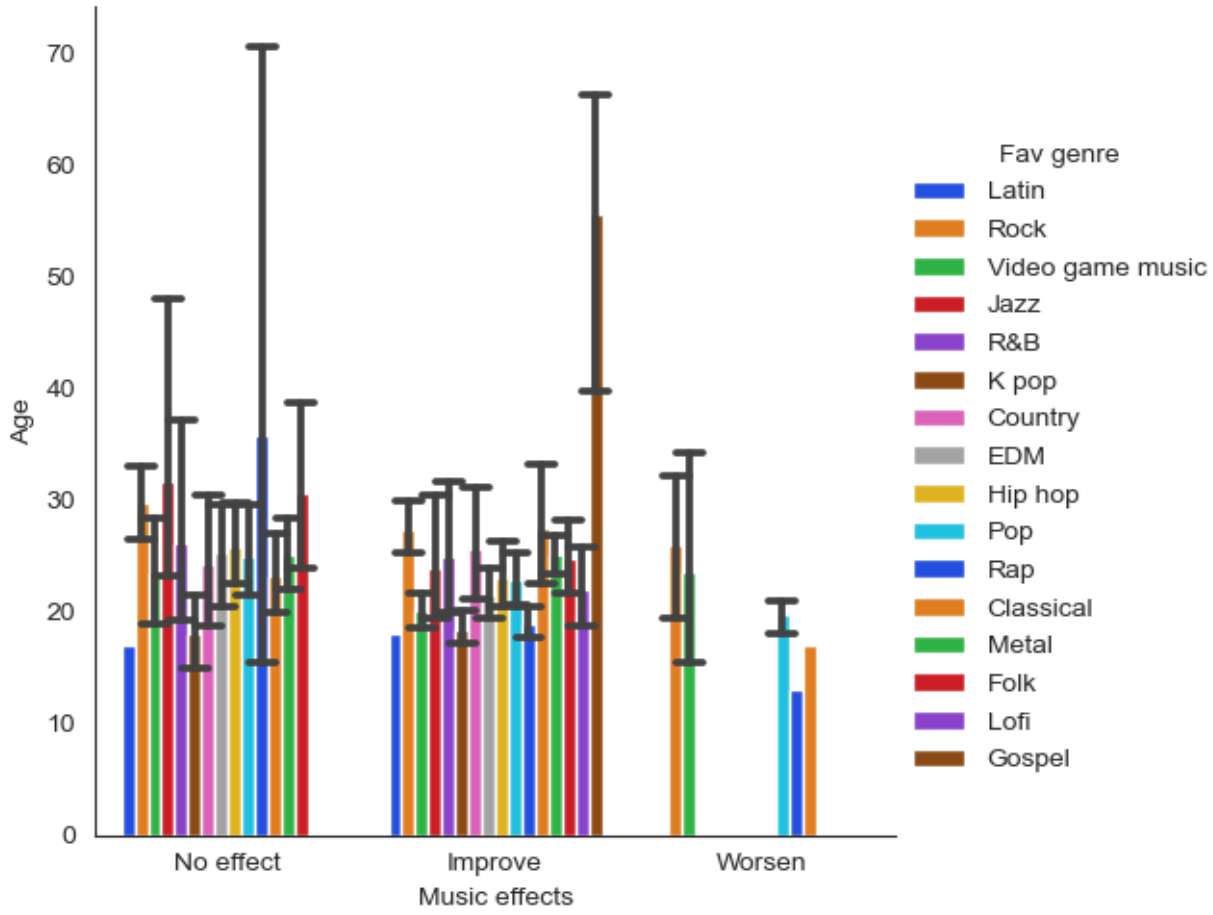


11-Bar Plot.

NOTE: In line plot X variable in cetagorical form & Y is in numeric. (Is data followed normal distribution or not)

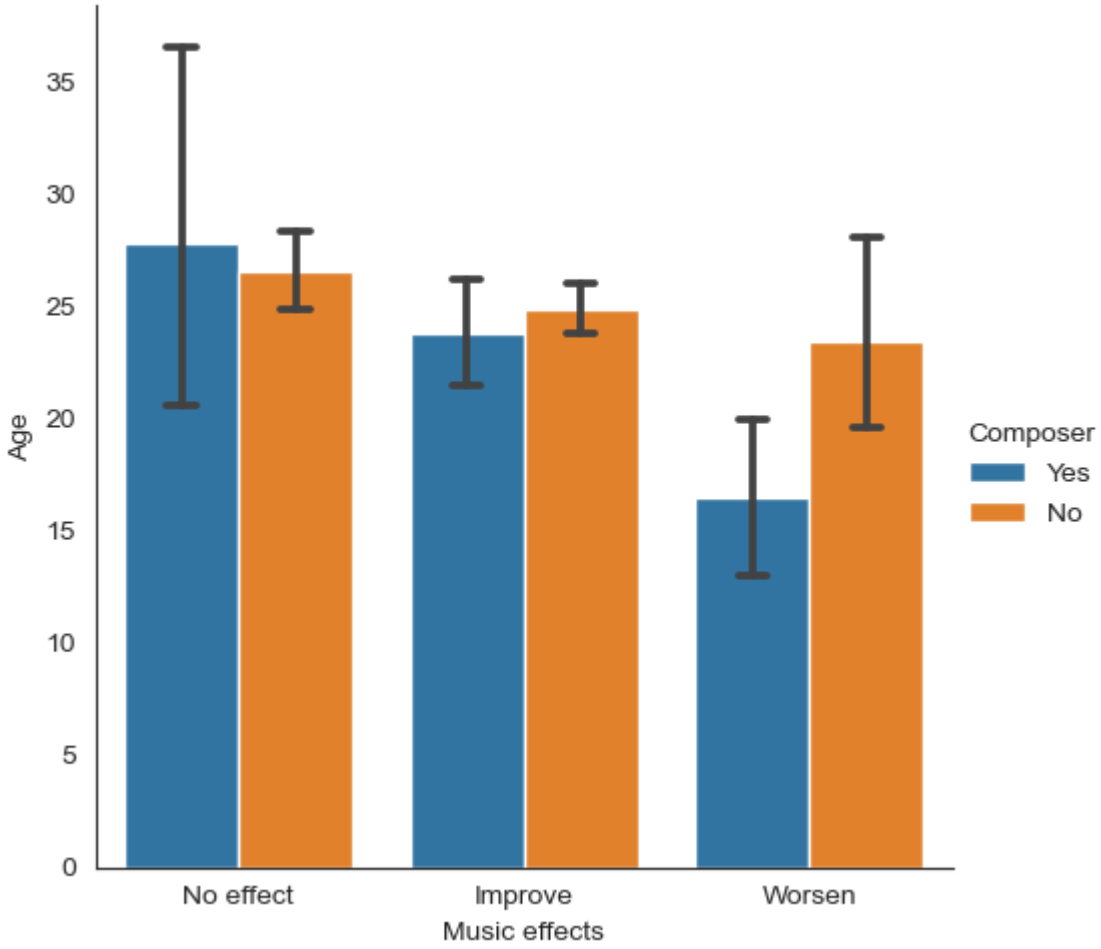
Multiple variable bar plot:

In [73]:
```
sns.catplot(data = music, x='Music effects',y='Age',hue="Fav genre", kind='bar', capsi
plt.show()
```



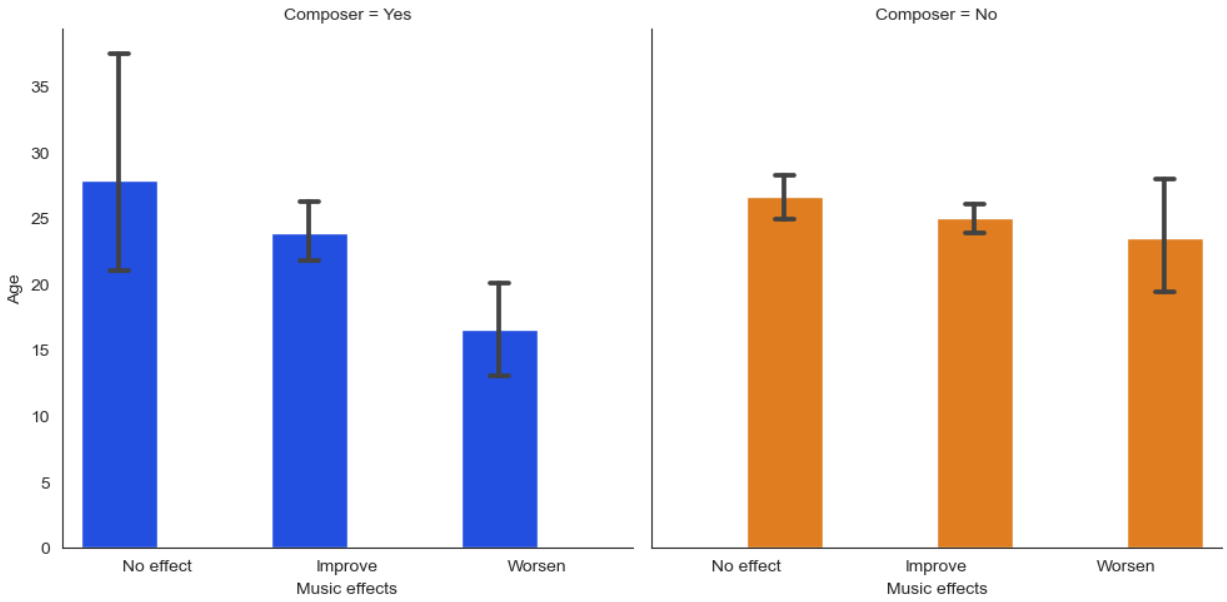
Group bar plot:

In [72]:
```
sns.catplot(data = music, x='Music effects',y='Age',hue="Composer", kind='bar', capsiz
plt.show()
```



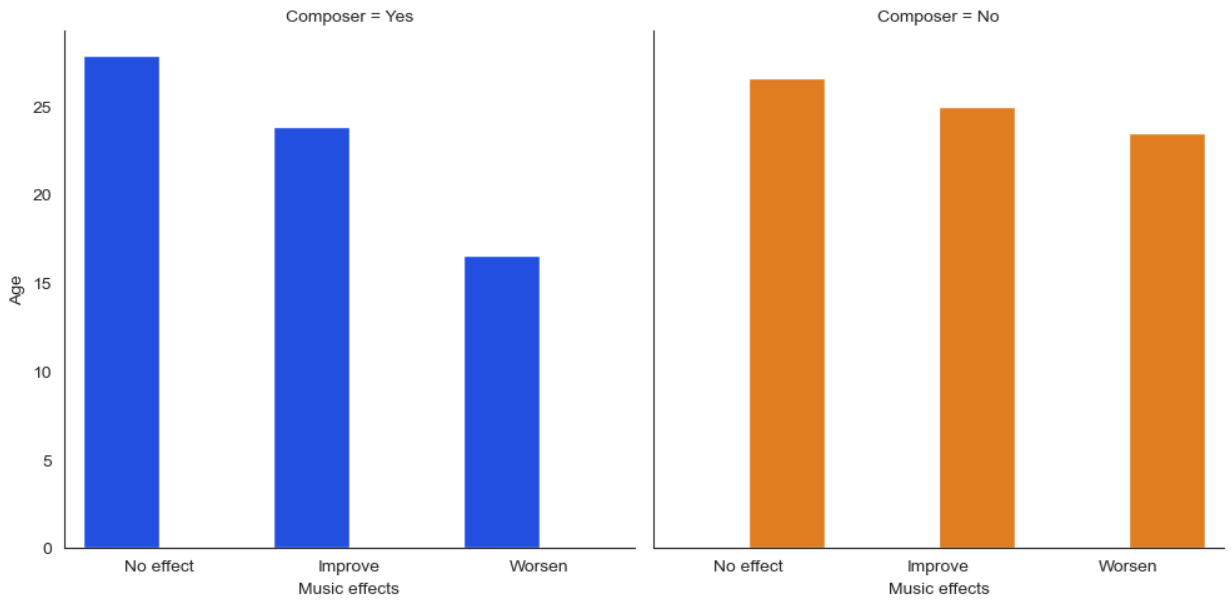
Seperating groups:

In [75]:
```
sns.catplot(data = music, x='Music effects',y='Age',hue="Composer", kind='bar', capsiz
plt.show()
```



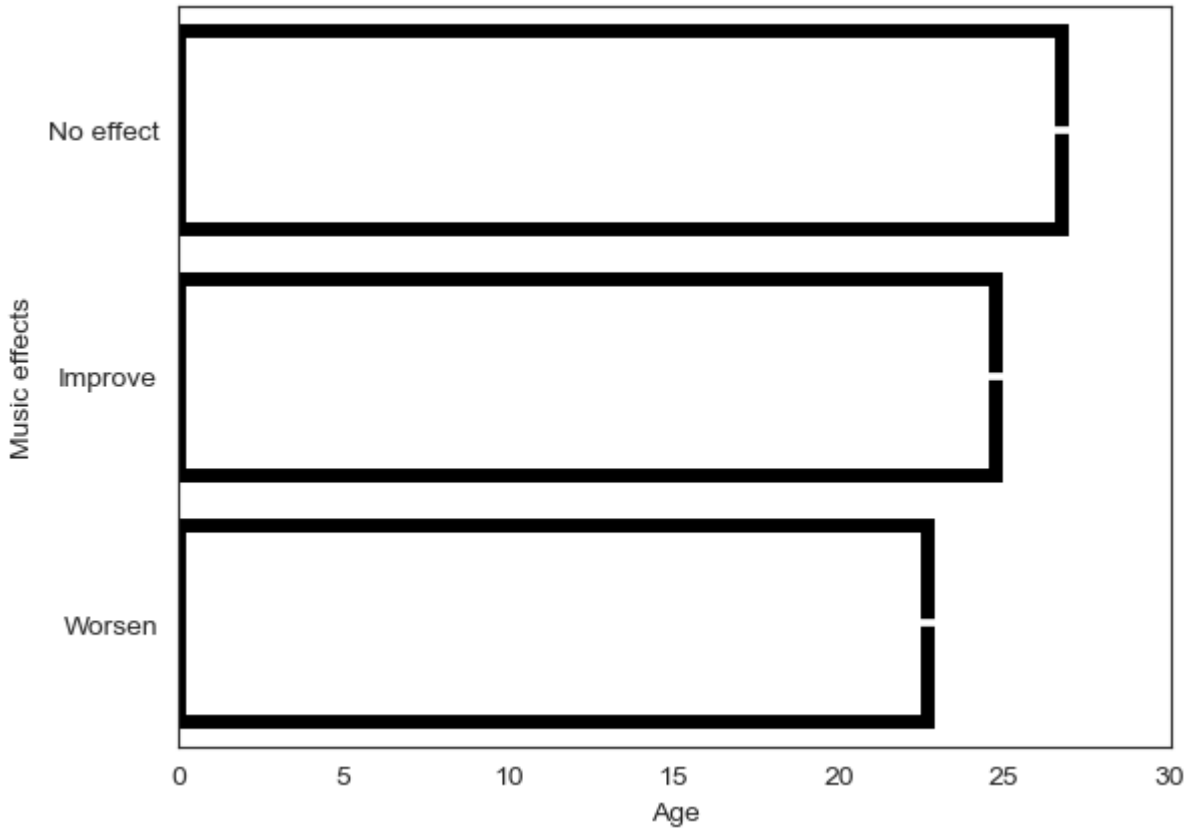
Removing Error bars:

In [76]:
```
sns.catplot(data = music, x='Music effects',y='Age',hue="Composer", kind='bar', capsiz
plt.show()
```



horizental bar plot:

In [77]:
```
P=sns.barplot(x="Age", y="Music effects", data=music, linewidth=5, facecolor=(1, 1, 1,
plt.show()
```

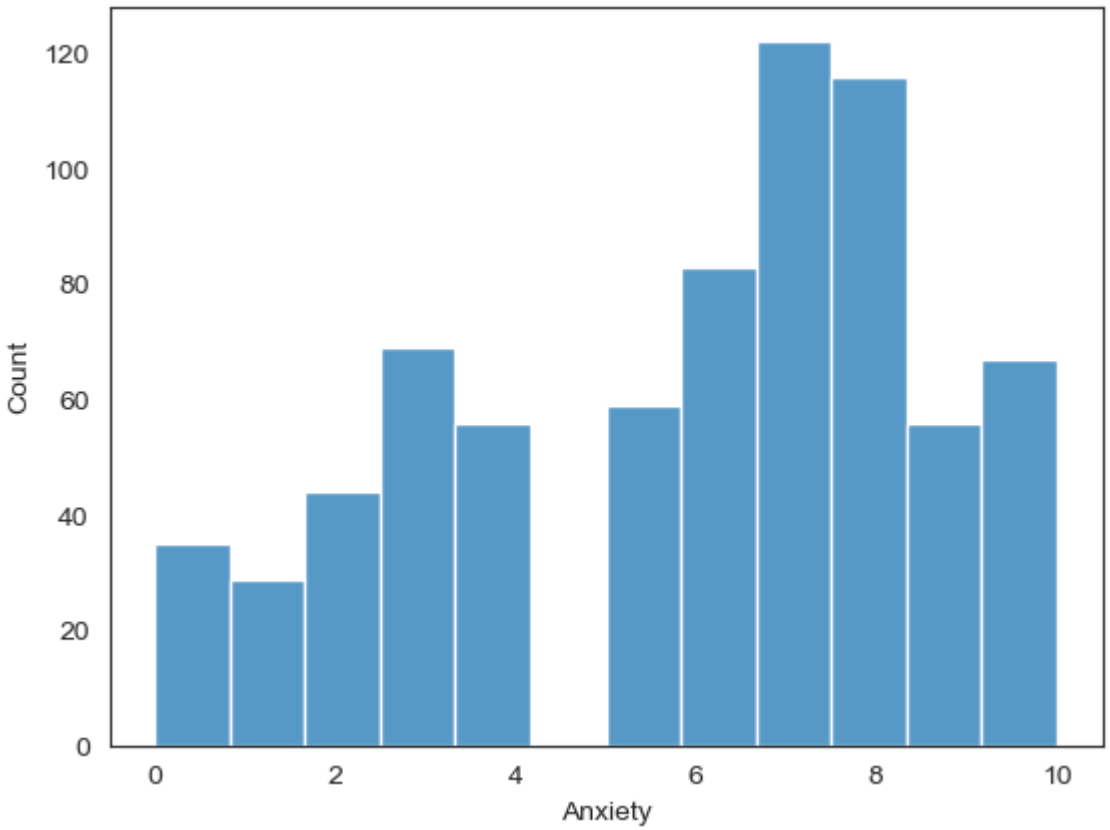


12-Histogram Plot.

NOTE: In Histogram plot only one numerical variable we use. (it usually tell us about normal distribution, and how our data is spreaded)

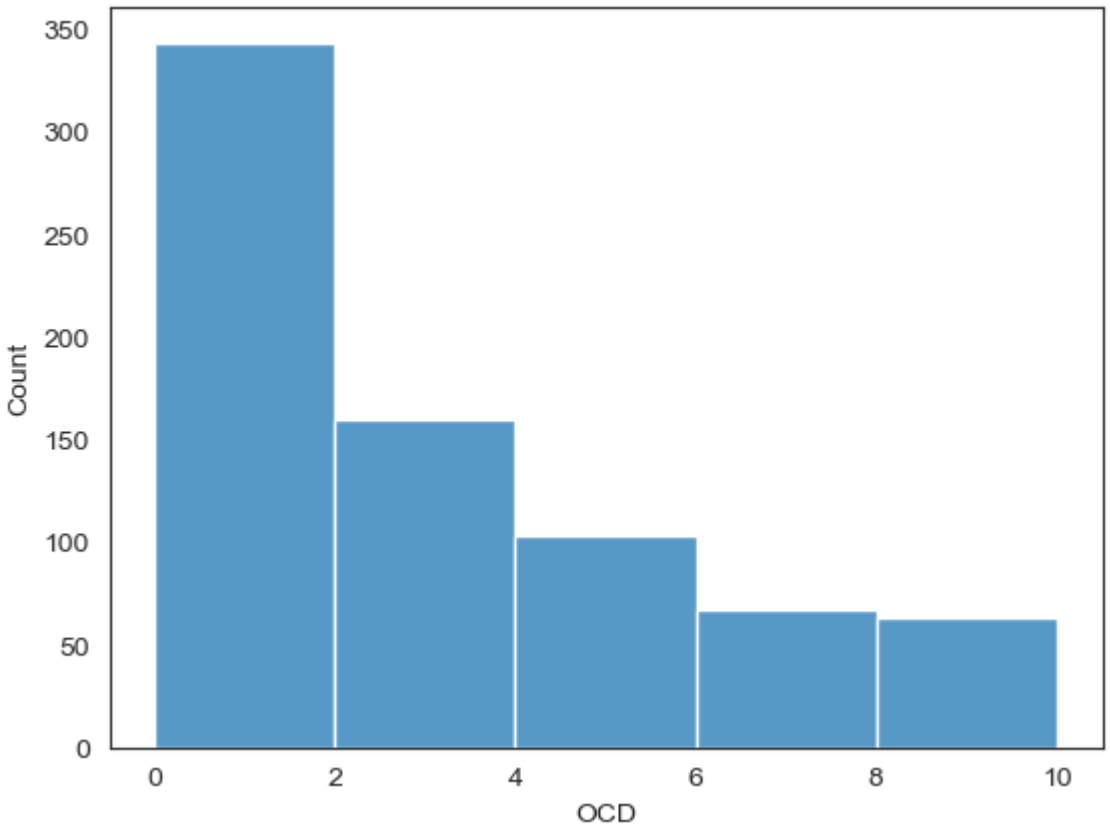
Simple Hist plot:

In [84]:
```
sns.histplot(data=music, x="Anxiety")
plt.show()
```



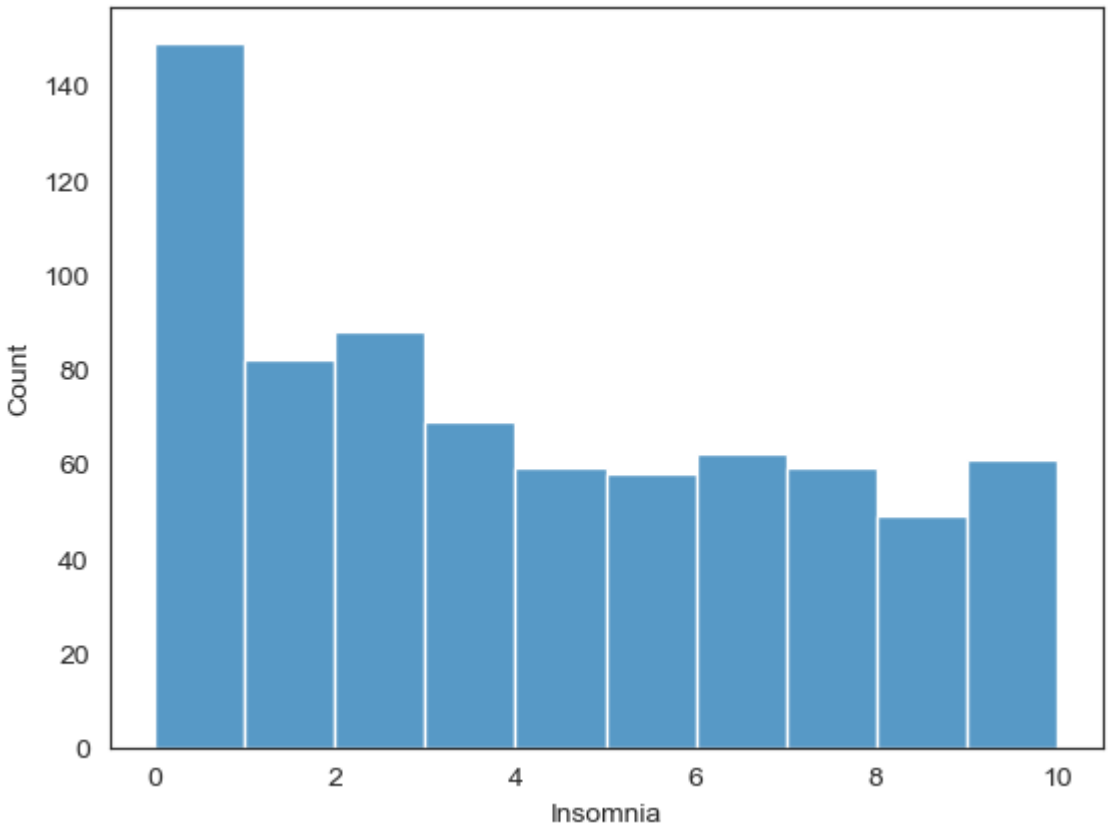
Hist plot by adjusting bin size:

In [86]:
```
sns.histplot(data=music, x='OCD', palette='Antique', binwidth=2)
plt.show()
```



Hist plot by adjusting number os bins:

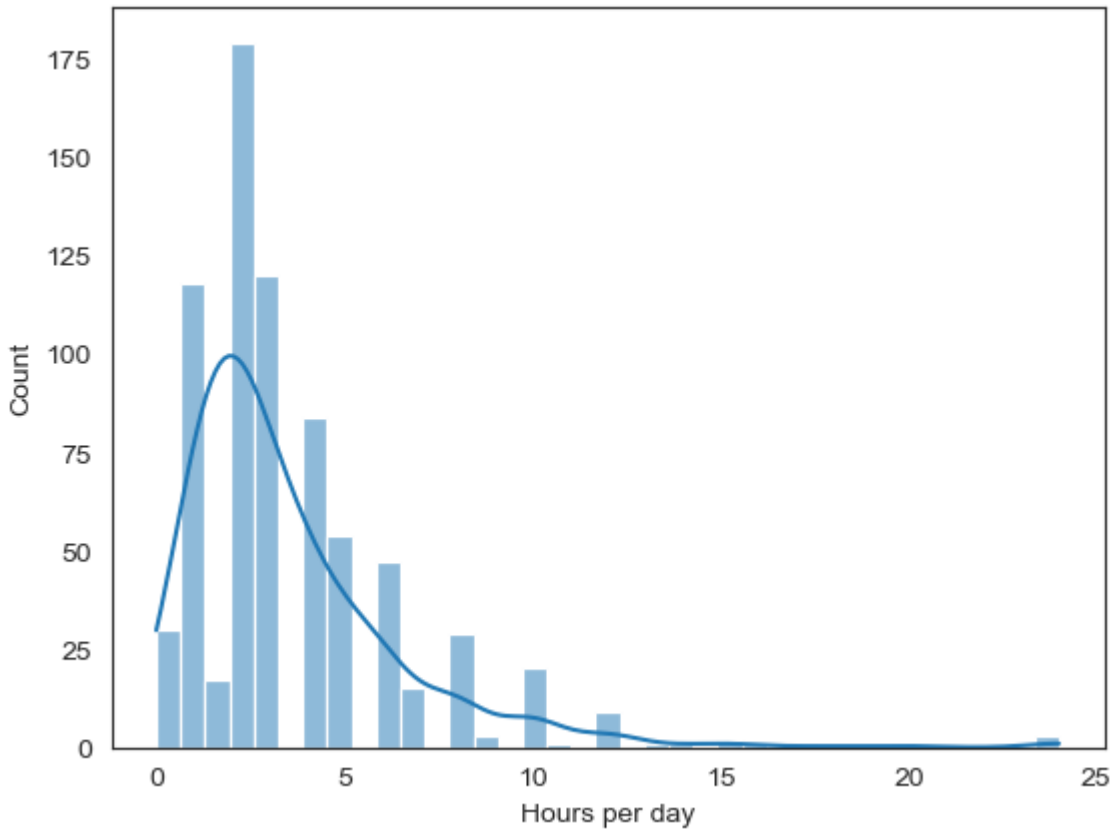
In [89]:
```
sns.histplot(data=music, x='Insomnia', palette='o3', bins=10)
plt.show()
```



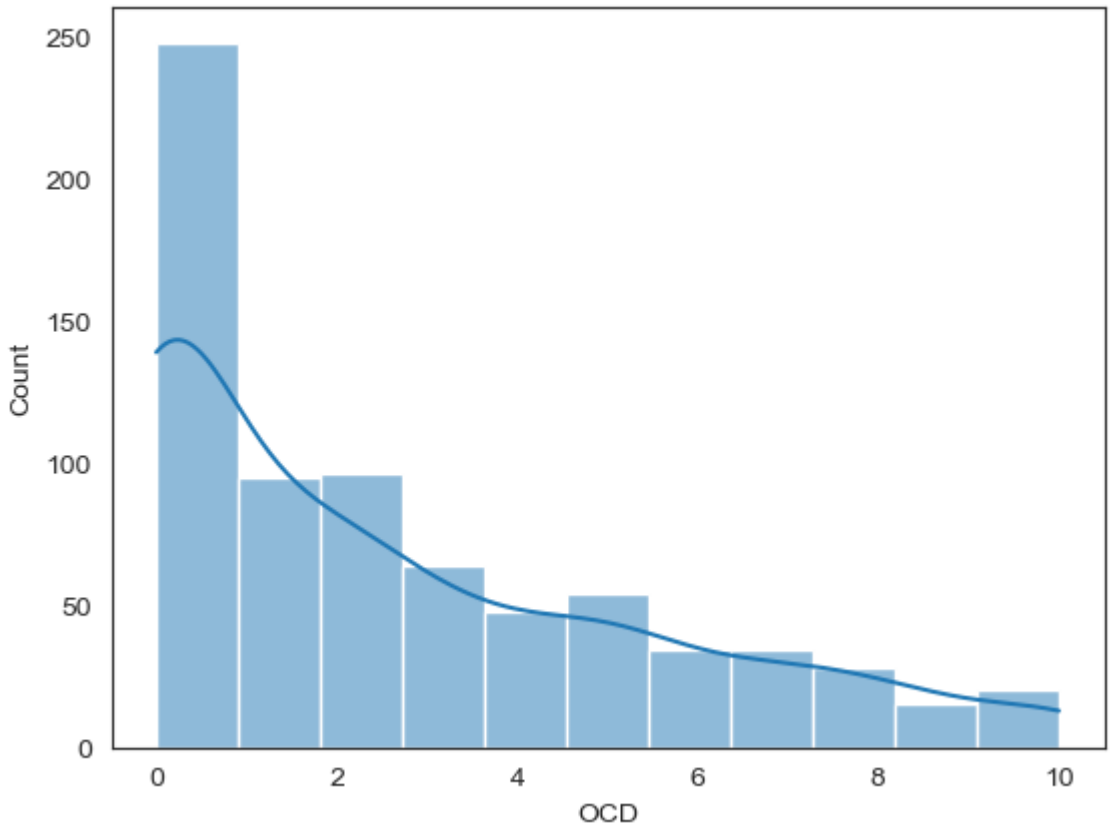
13-Histogram show with trend line

In [78]:
```
sns.histplot(data=music, x="Hours per day", kde=True)
plt.show
```

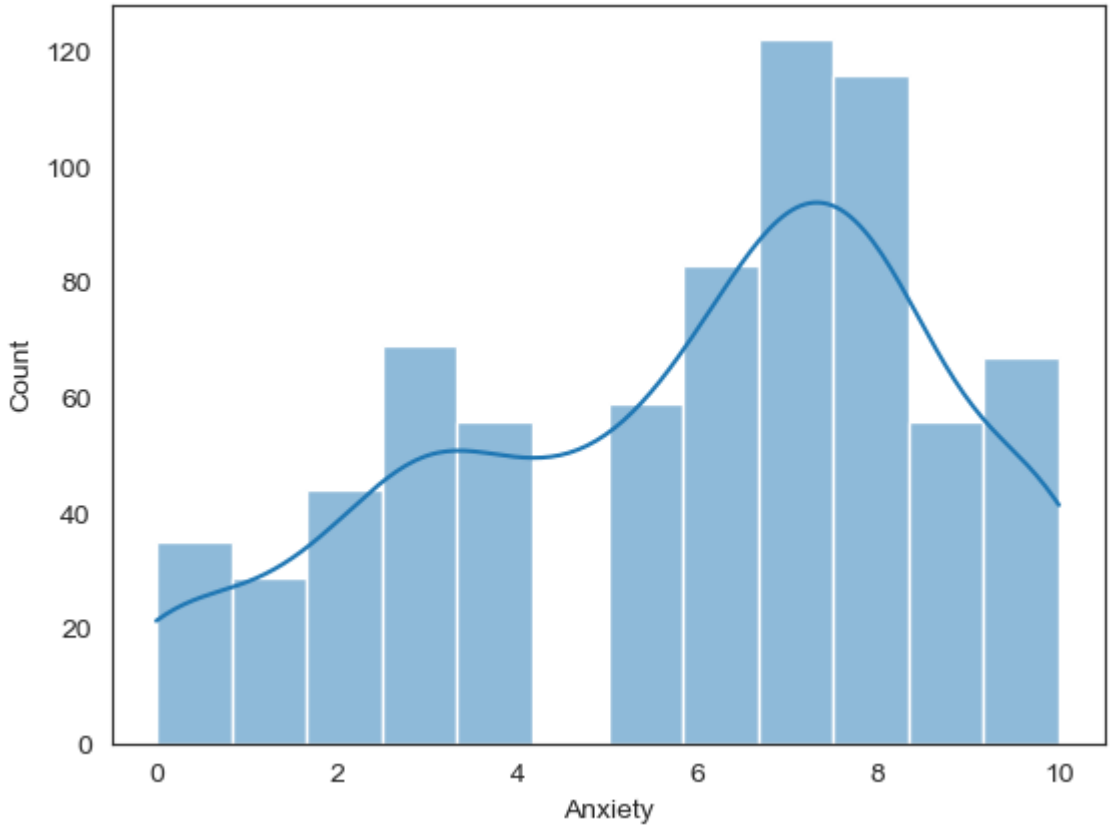
Out[78]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



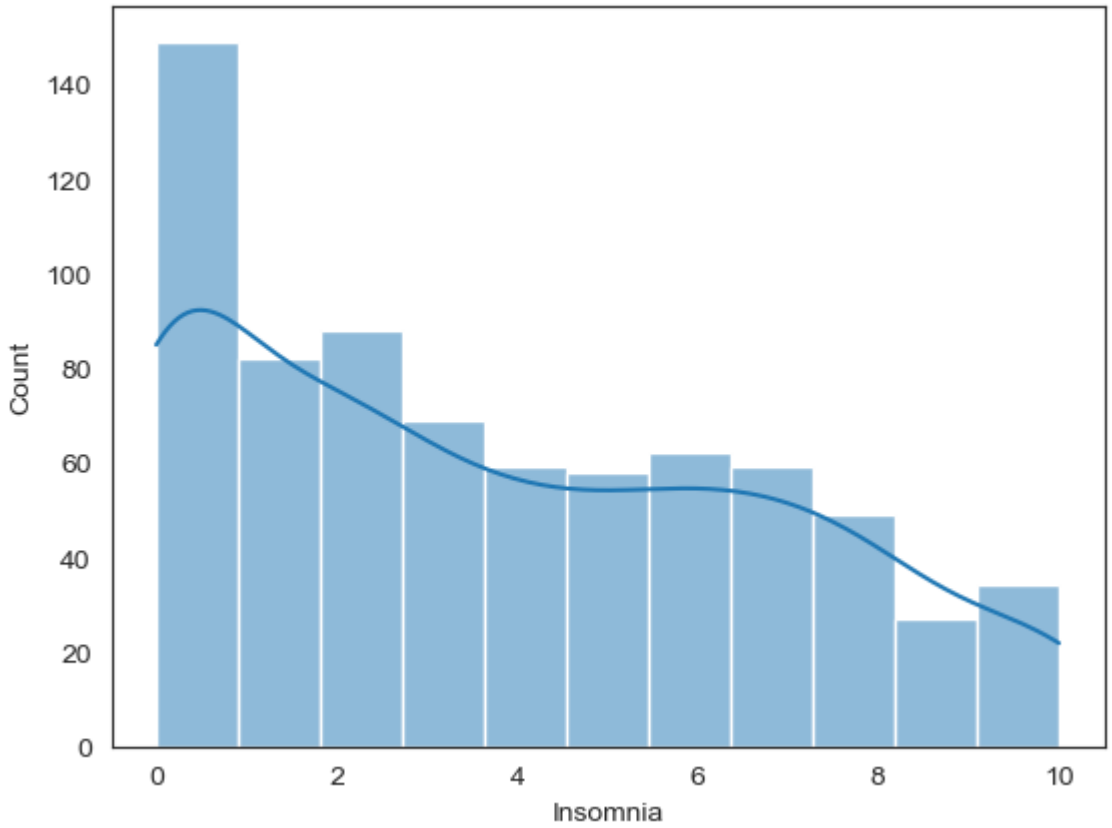
In [79]:
```
sns.histplot(data=music, x="OCD", kde=True)
plt.show()
```



In [82]:
```
sns.histplot(data=music, x="Anxiety", kde=True)
plt.show()
```



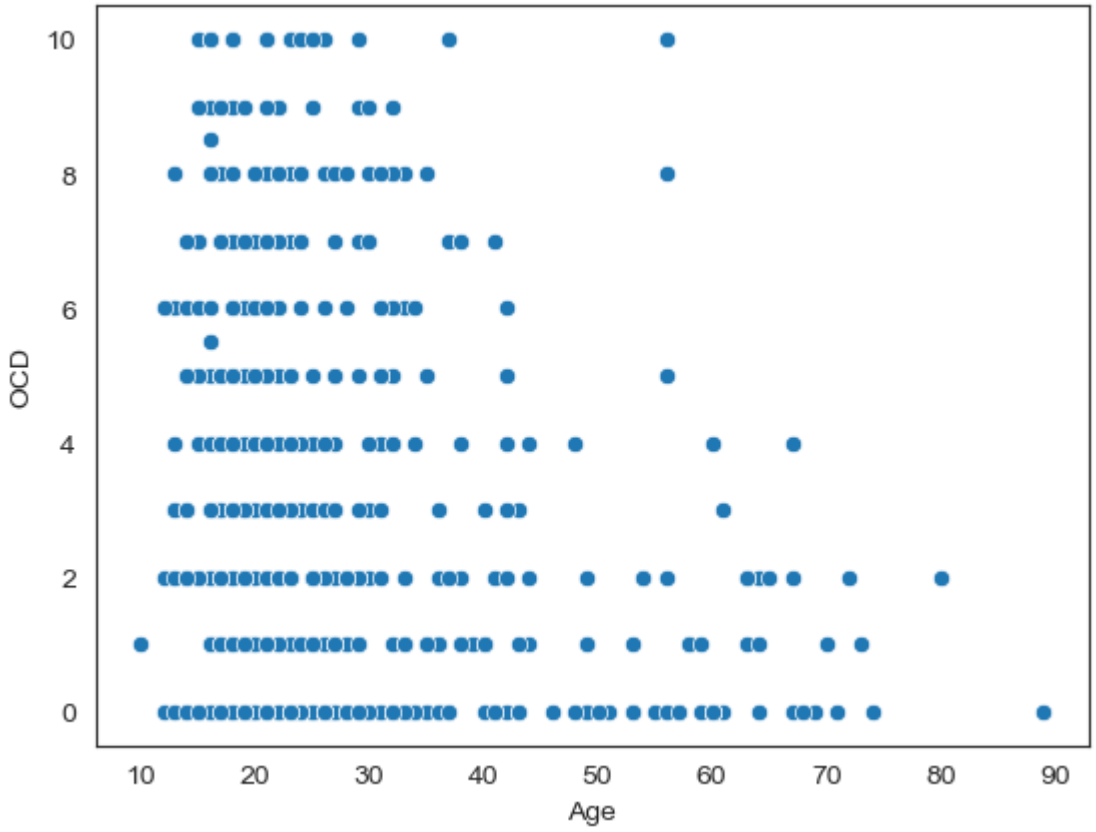
In [83]:
```
sns.histplot(data=music, x="Insomnia", kde=True)
plt.show()
```



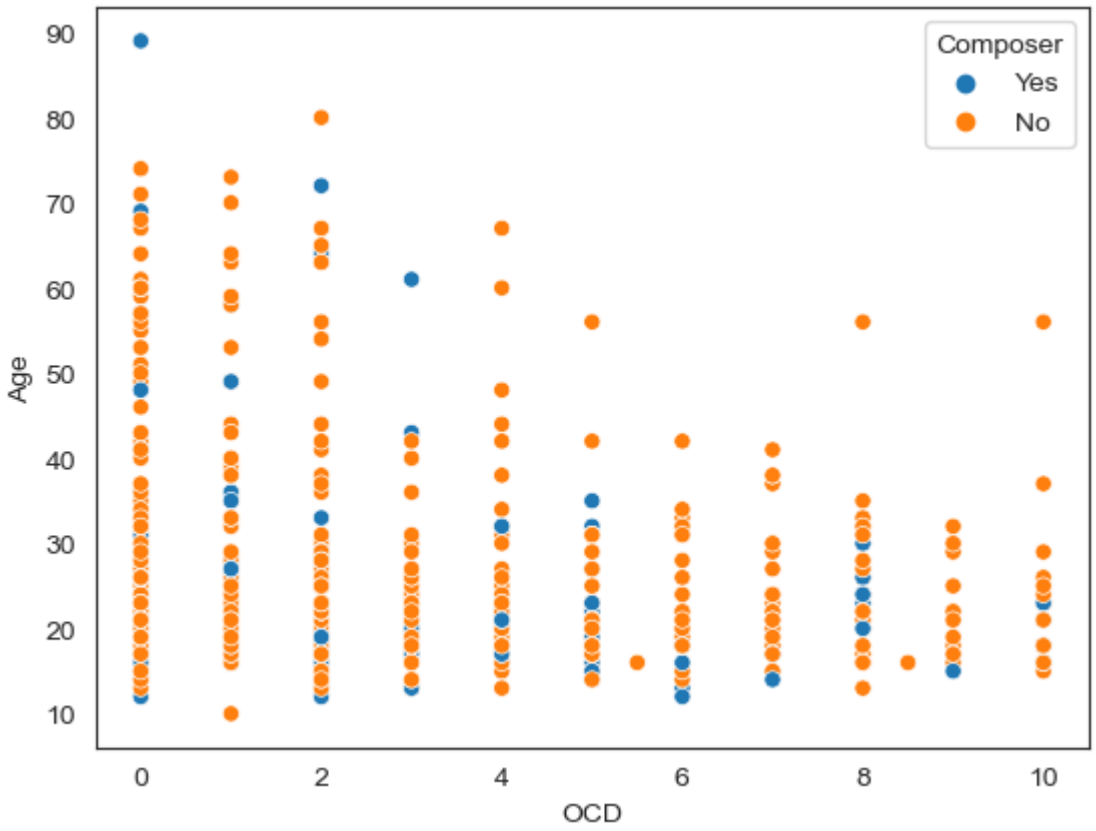
14-Scatter Plot.

NOTE: In Scatter plot we use two X & Y variable both may contain numerical data. (corelation)

In [93]:
```
sns.scatterplot(data = music, x='Age',y="OCD")
plt.show()
```



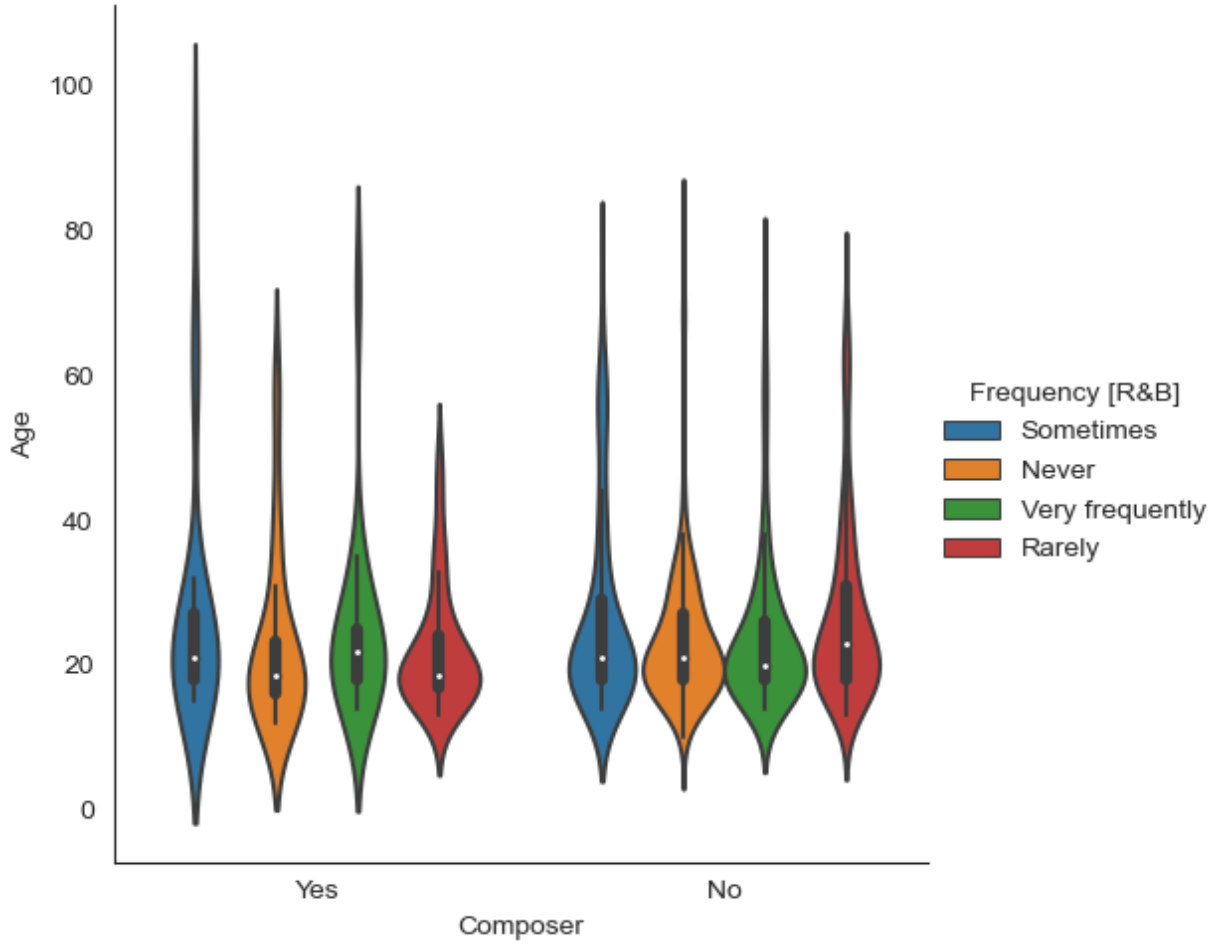
In [94]:
```
sns.scatterplot(data = music, x='OCD',y='Age',hue="Composer")
plt.show()
```



15-Violin Plot

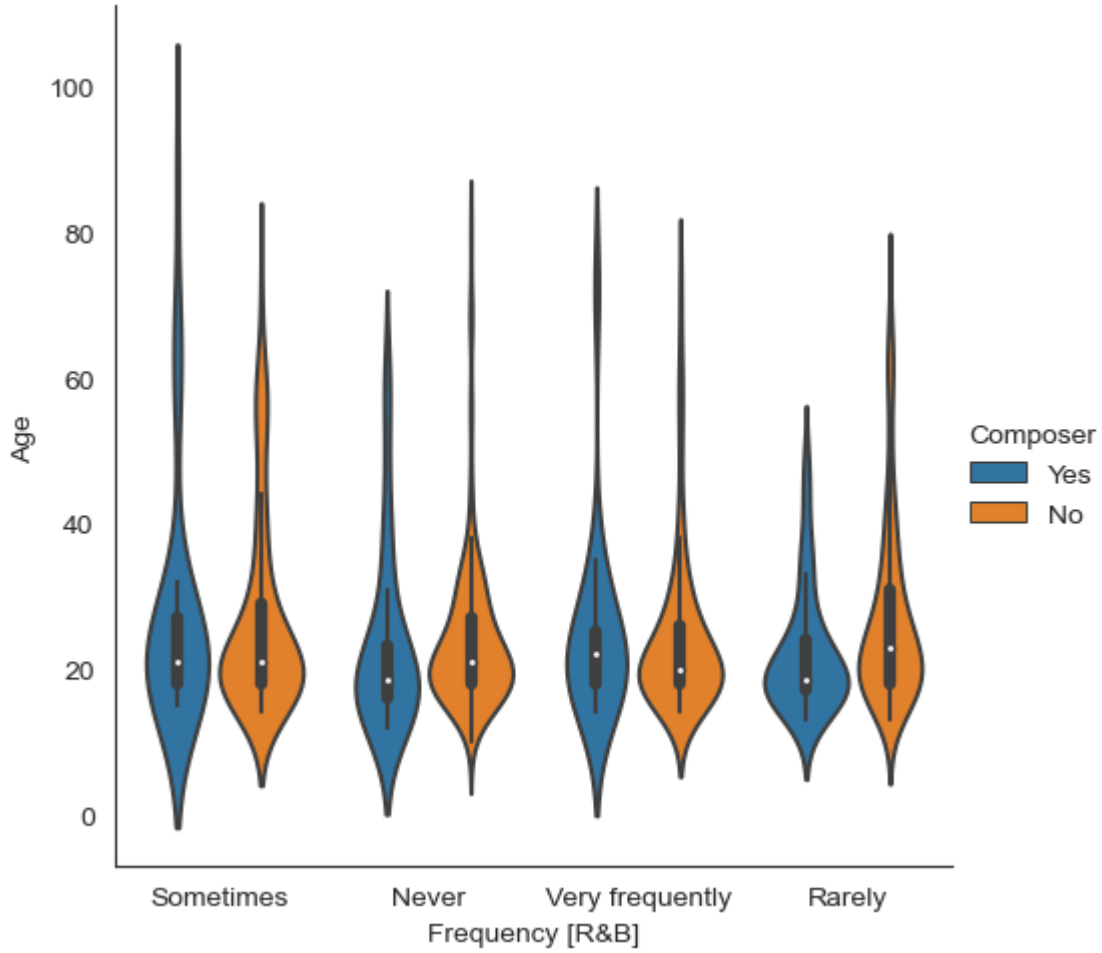
In [101…

```
sns.catplot(data = music, x='Composer',y='Age',hue='Frequency [R&B]',kind='violin')
plt.show()
```



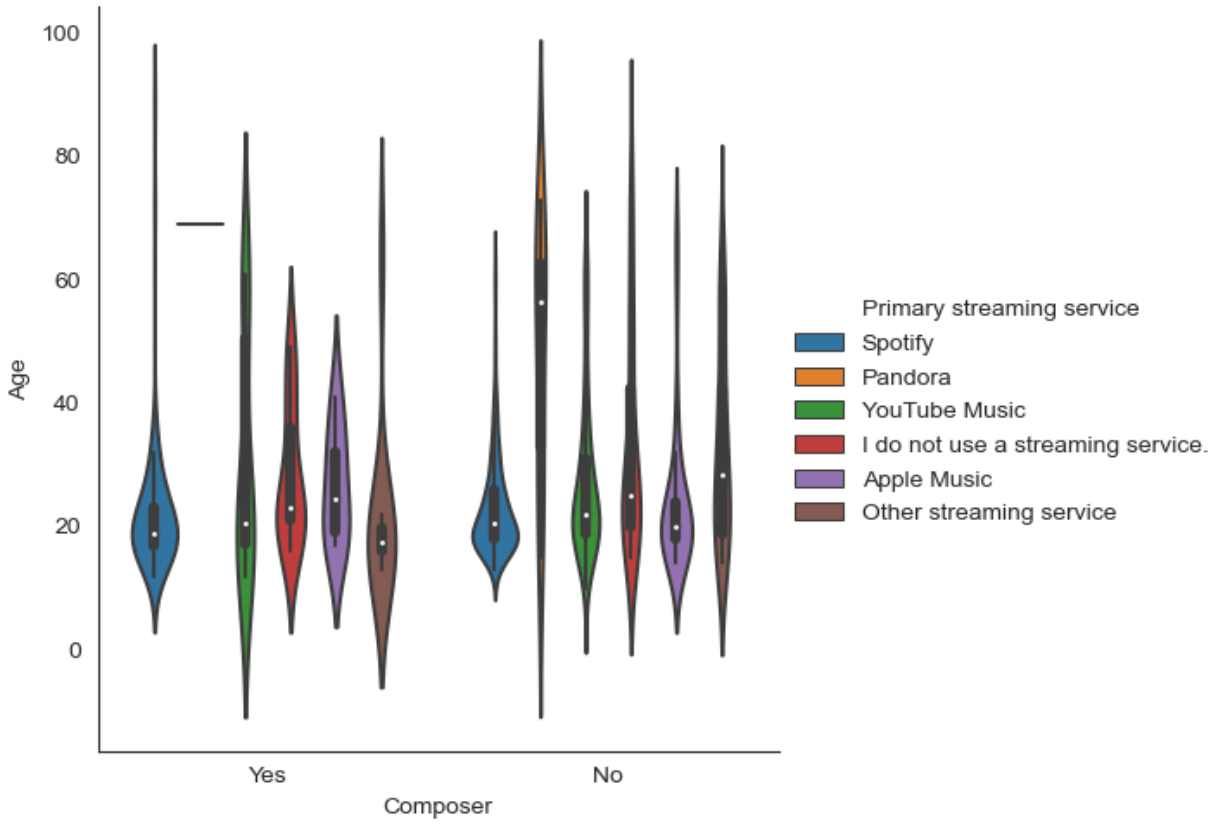
In [102…

```
sns.catplot(data = music, x='Frequency [R&B]',y='Age',hue='Composer',kind='violin')
plt.show()
```

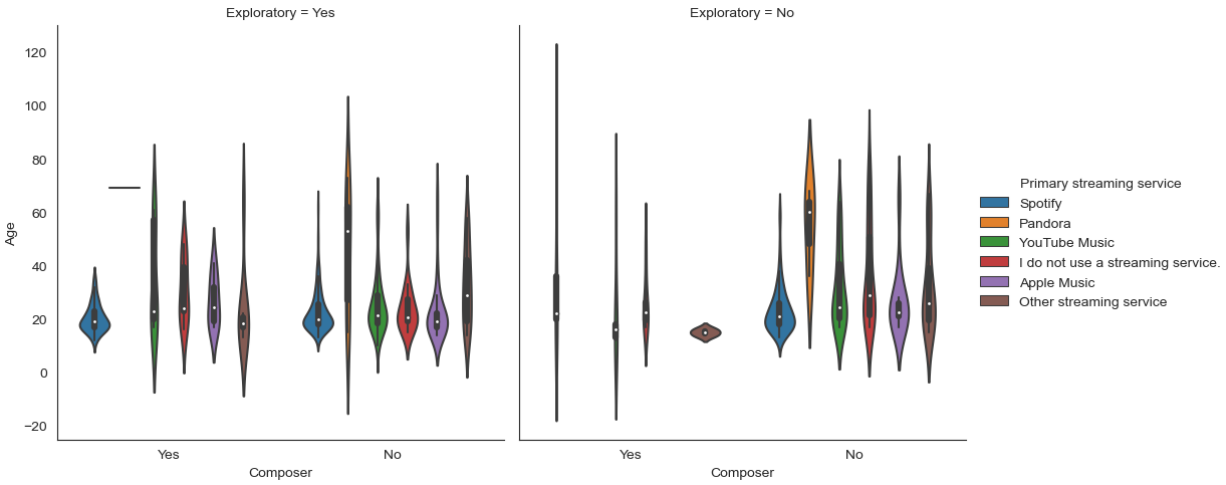


In [104…

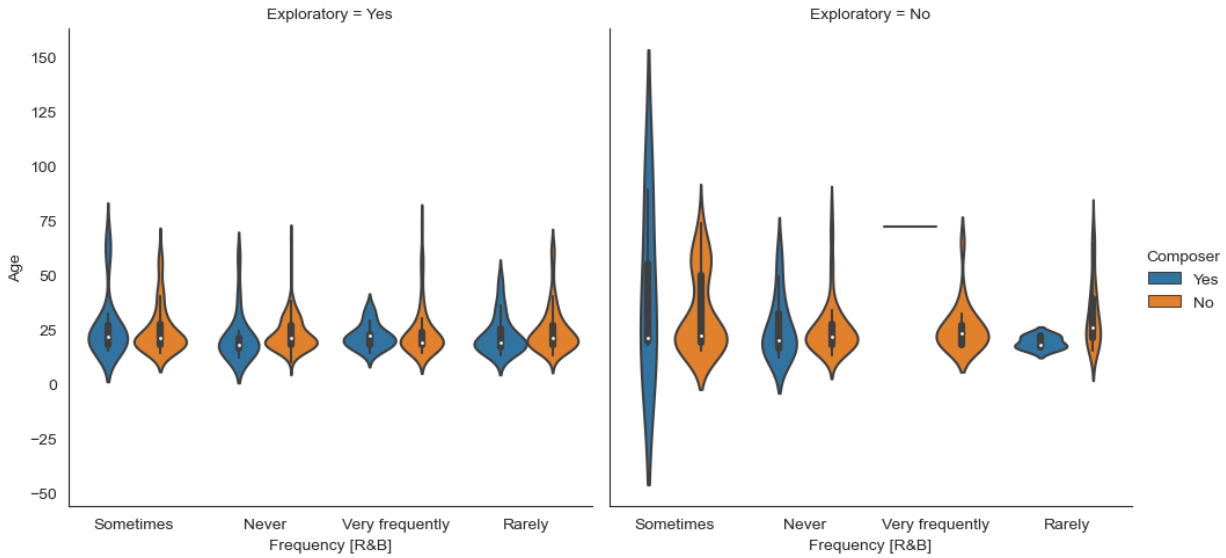
```
sns.catplot(data = music, x='Composer',y='Age',hue='Primary streaming service',kind='v
plt.show()
```



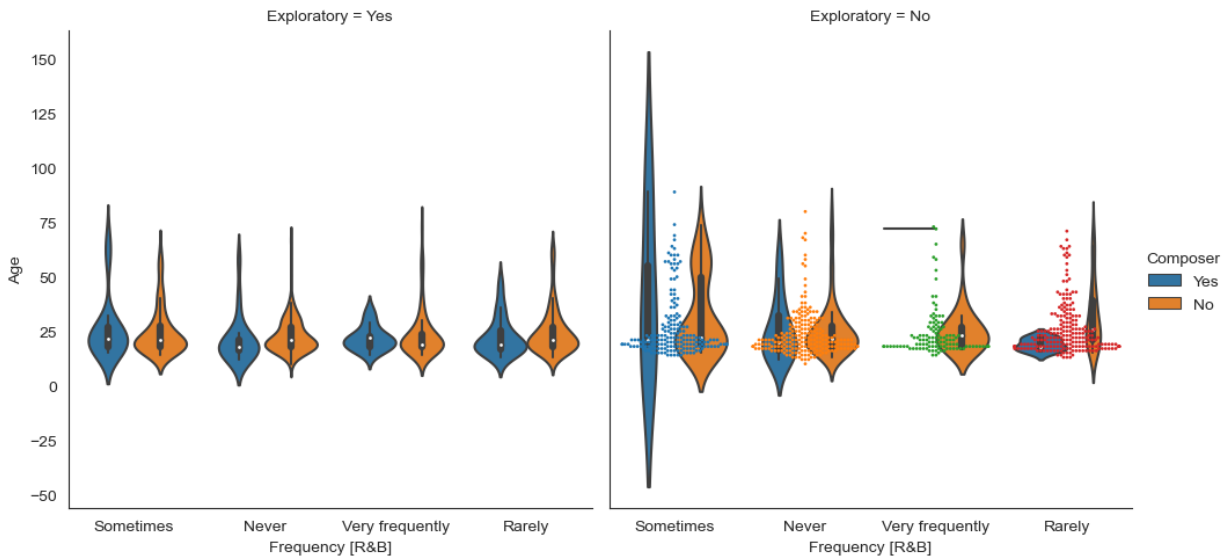
In [106…
```
sns.catplot(data = music, x='Composer',y='Age',hue='Primary streaming service', col='E
plt.show()
```



In [108…
```
sns.catplot(data = music, x='Frequency [R&B]',y='Age',hue='Composer',kind='violin', co
plt.show()
```



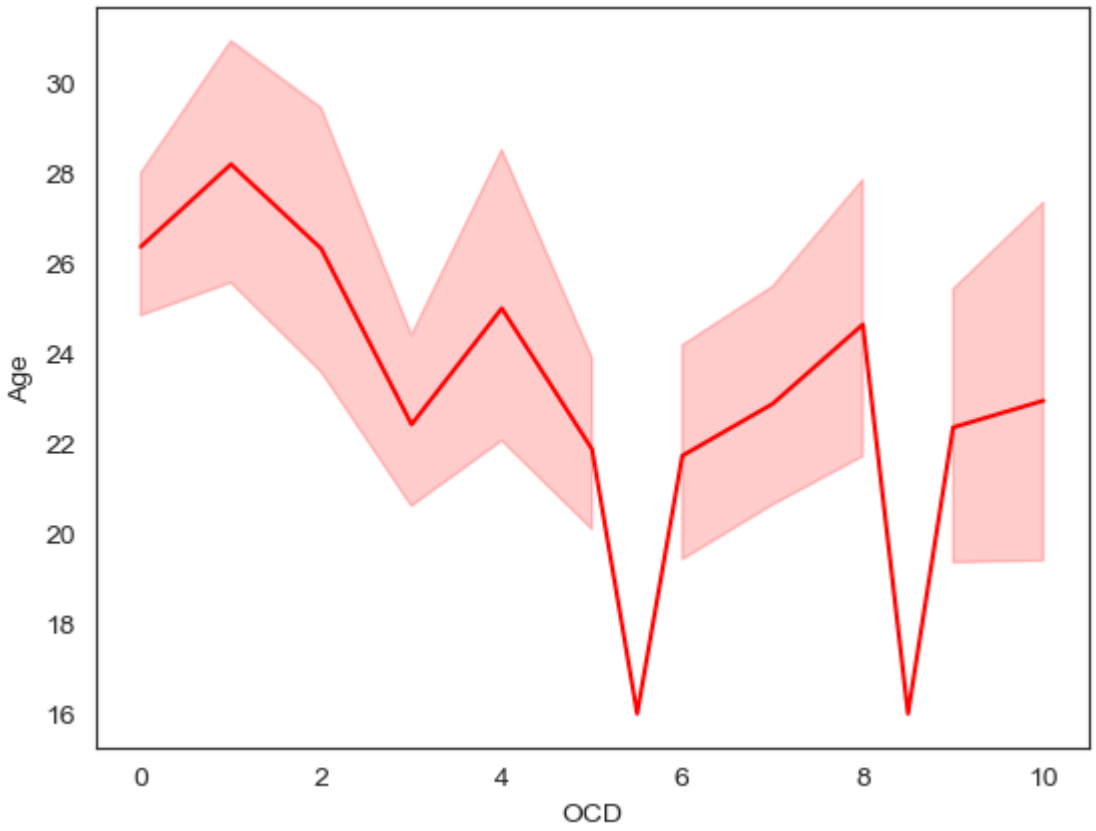
In [111…
```
sns.catplot(data = music, x='Frequency [R&B]',y='Age',hue='Composer',kind='violin', co
sns.swarmplot(data = music, x='Frequency [R&B]',y='Age',size=2)
plt.show()
```



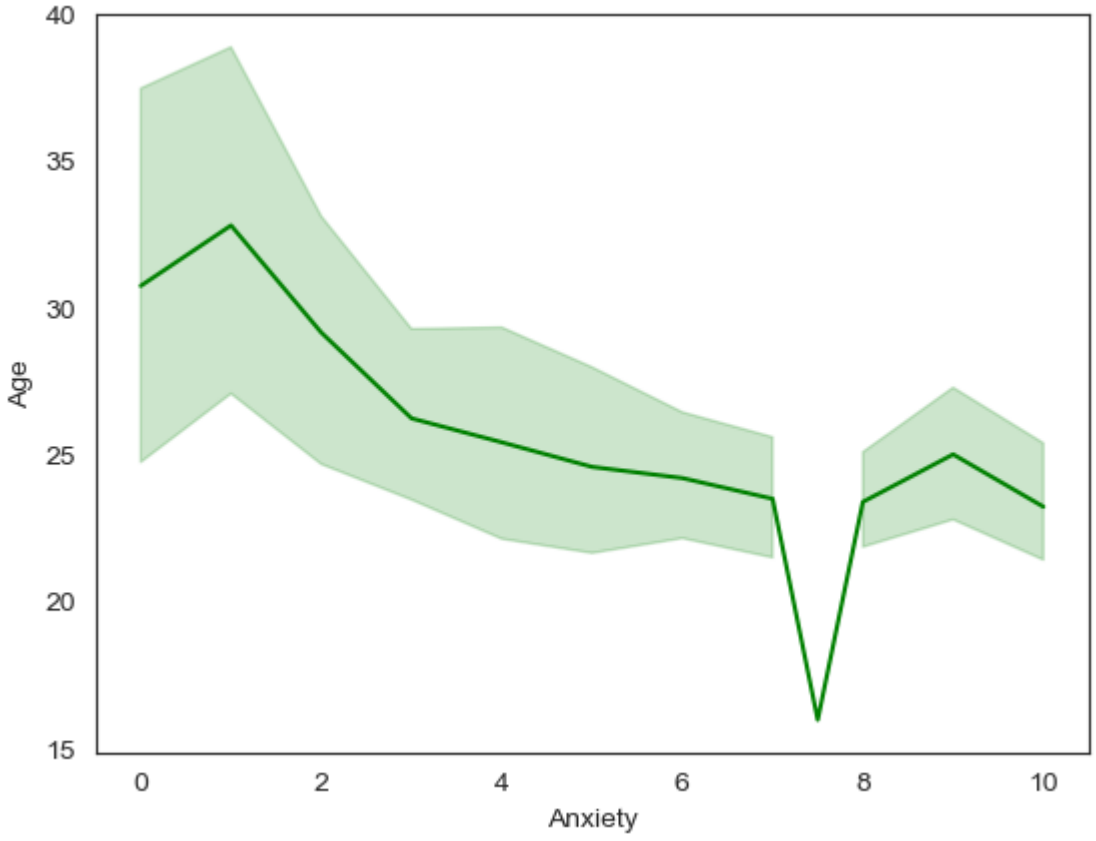
16-Line Plot

NOTE: In line plot we use two X & Y variable both may contain numerical data. (trend line, link, if one variable icreases which type of effects on other variable)

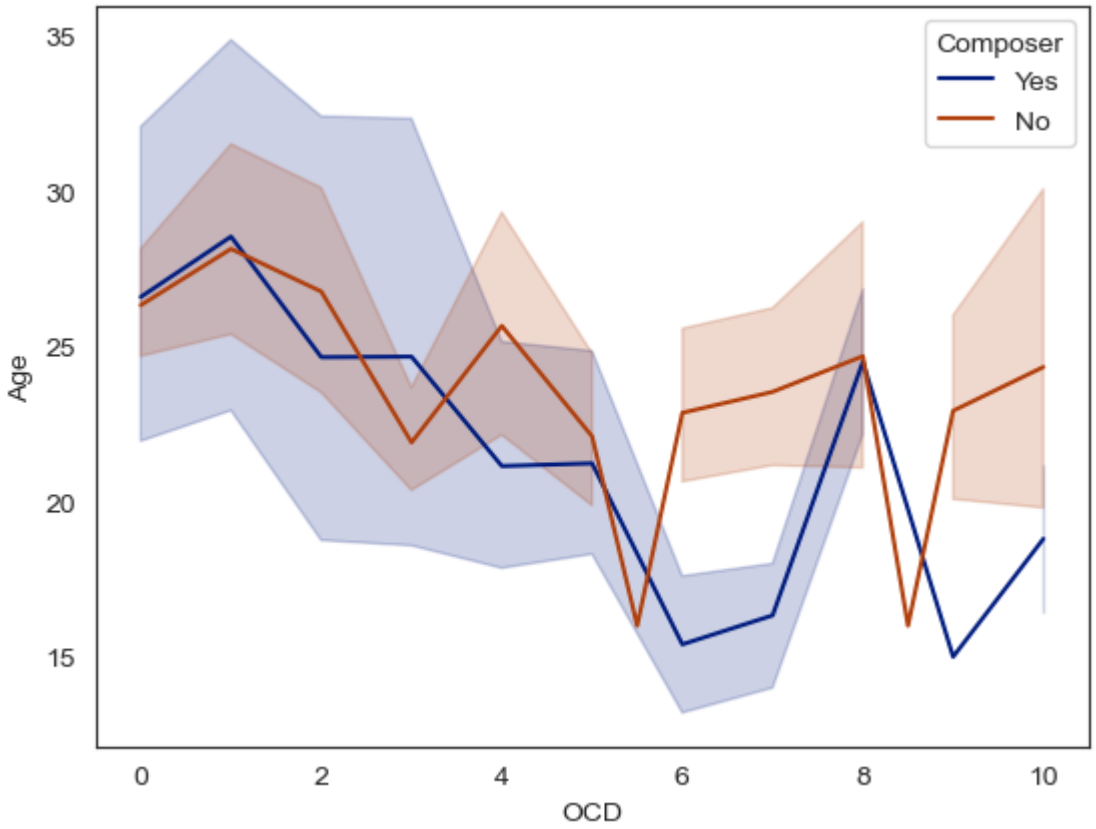
In [113…
```
sns.lineplot(data=music, x='OCD',y='Age', color = 'red')
plt.show()
```



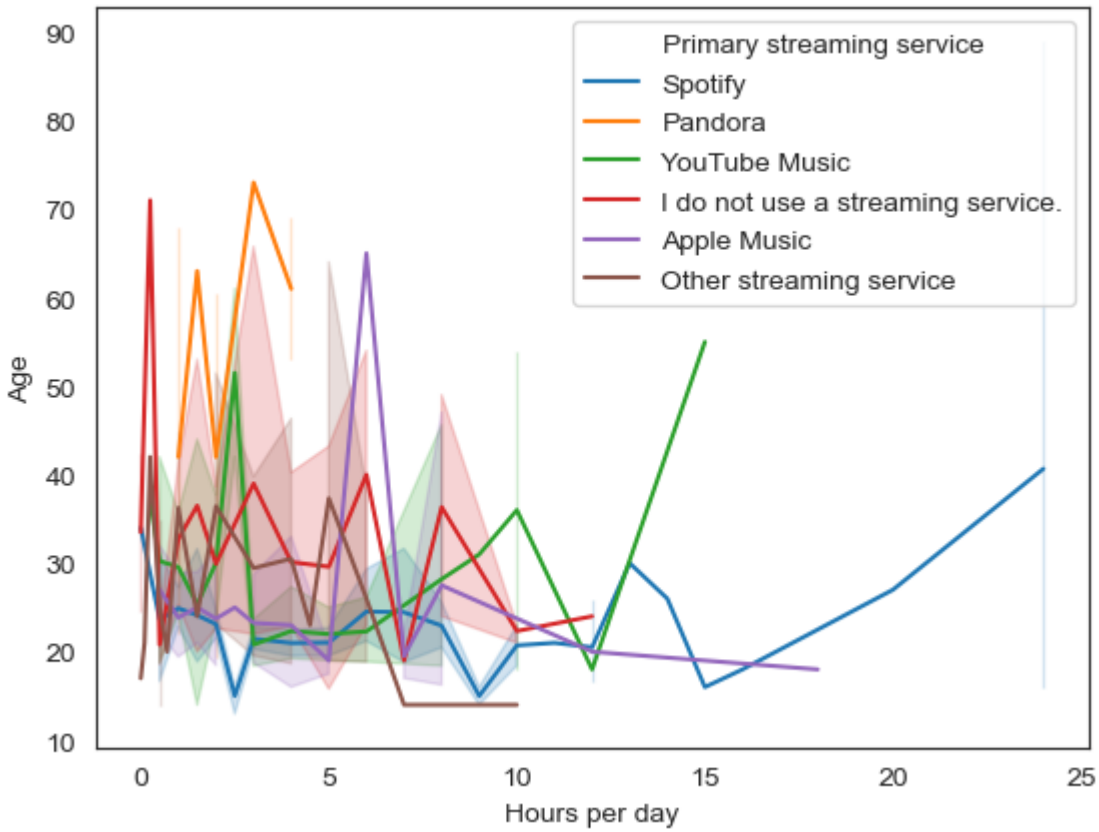
In [115…
```
sns.lineplot(data=music, x='Anxiety',y='Age', color = 'Green')
plt.show()
```



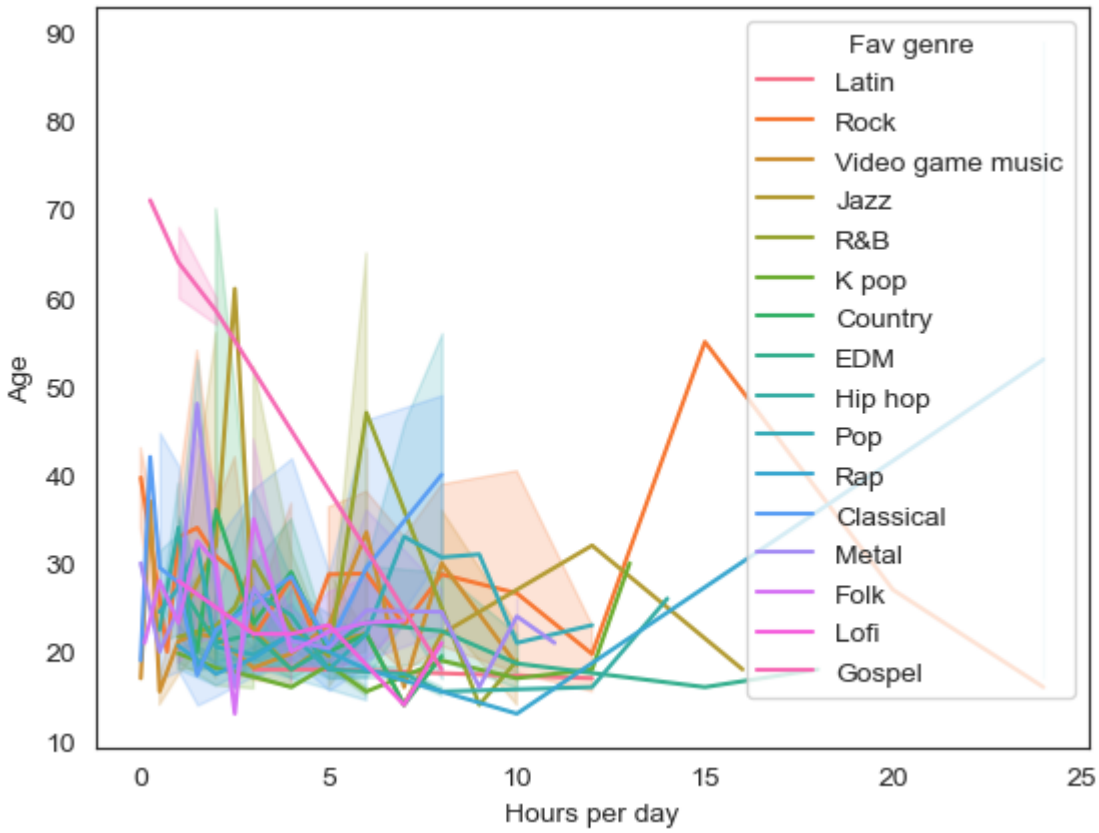
In [120…
```
sns.lineplot(data=music, x='OCD',y='Age', hue='Composer', palette="dark")
plt.show()
```



In [121…
```
sns.lineplot(data = music, x='Hours per day',y='Age',hue='Primary streaming service')
plt.show()
```



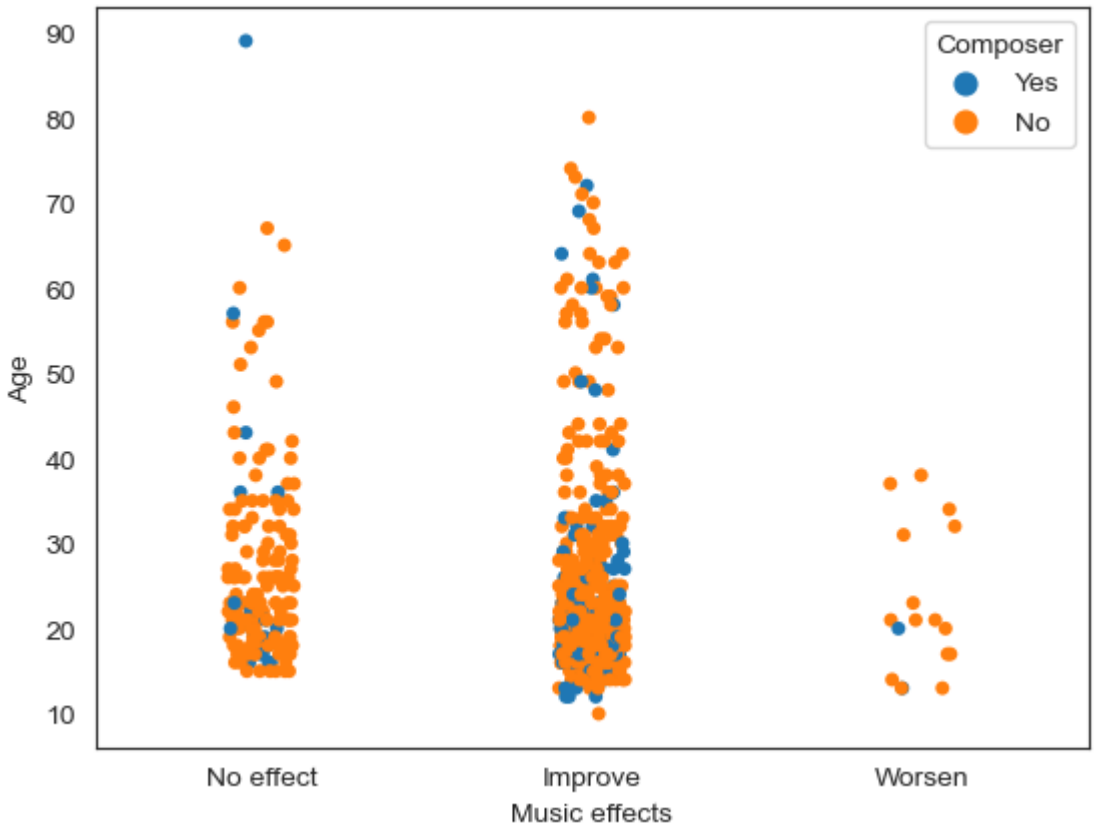
In [122…
```
sns.lineplot(data = music, x='Hours per day',y='Age',hue='Fav genre')
plt.show()
```



In [ ]:
```
sns.stripplot(data= music, x= "lunch", y= "math score", jitter=True, hue="gender")
plt.show()
```

17-Stripplot

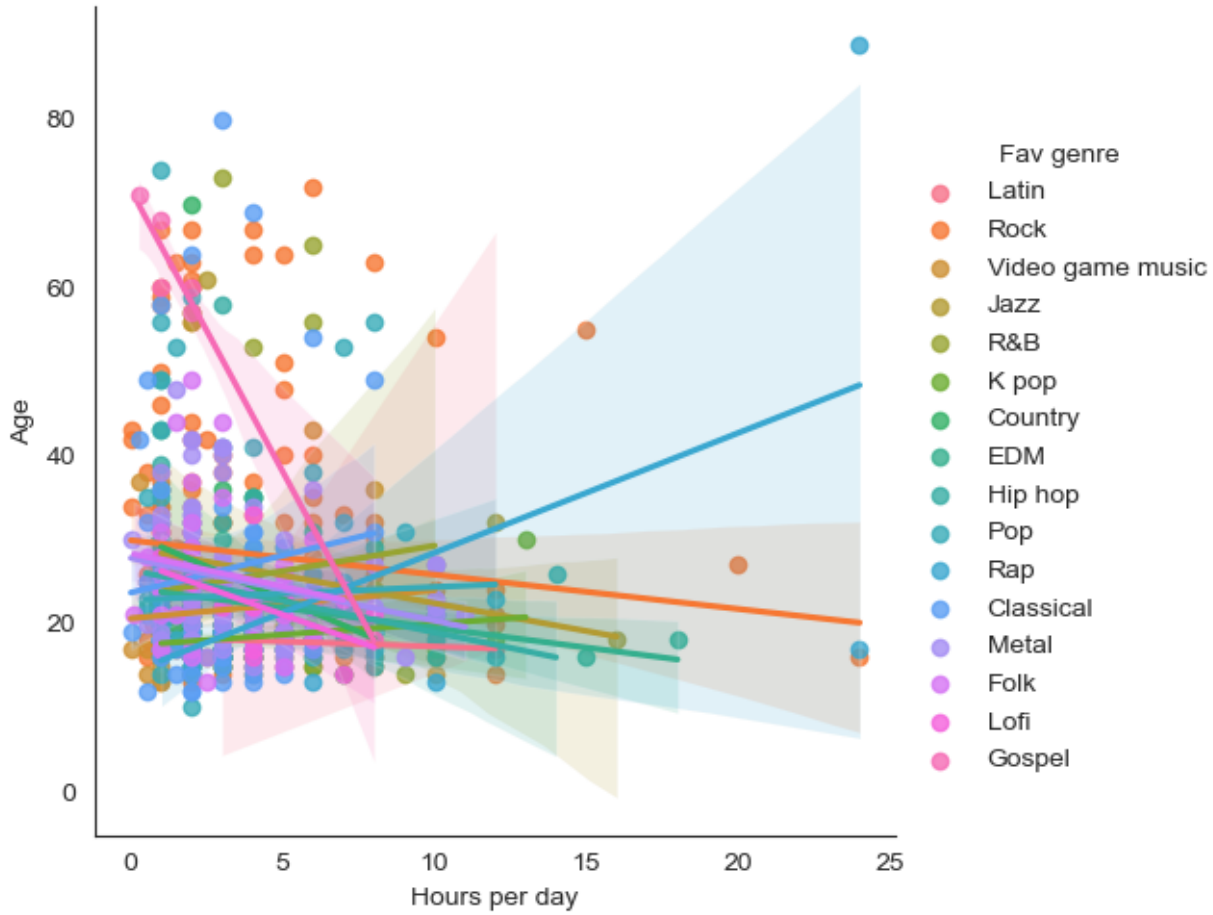
In [125…
```
sns.stripplot(data= music,  x='Music effects',y='Age',hue="Composer", jitter=True)
plt.show()
```



18-Draw a simple liner line graph

In [126…

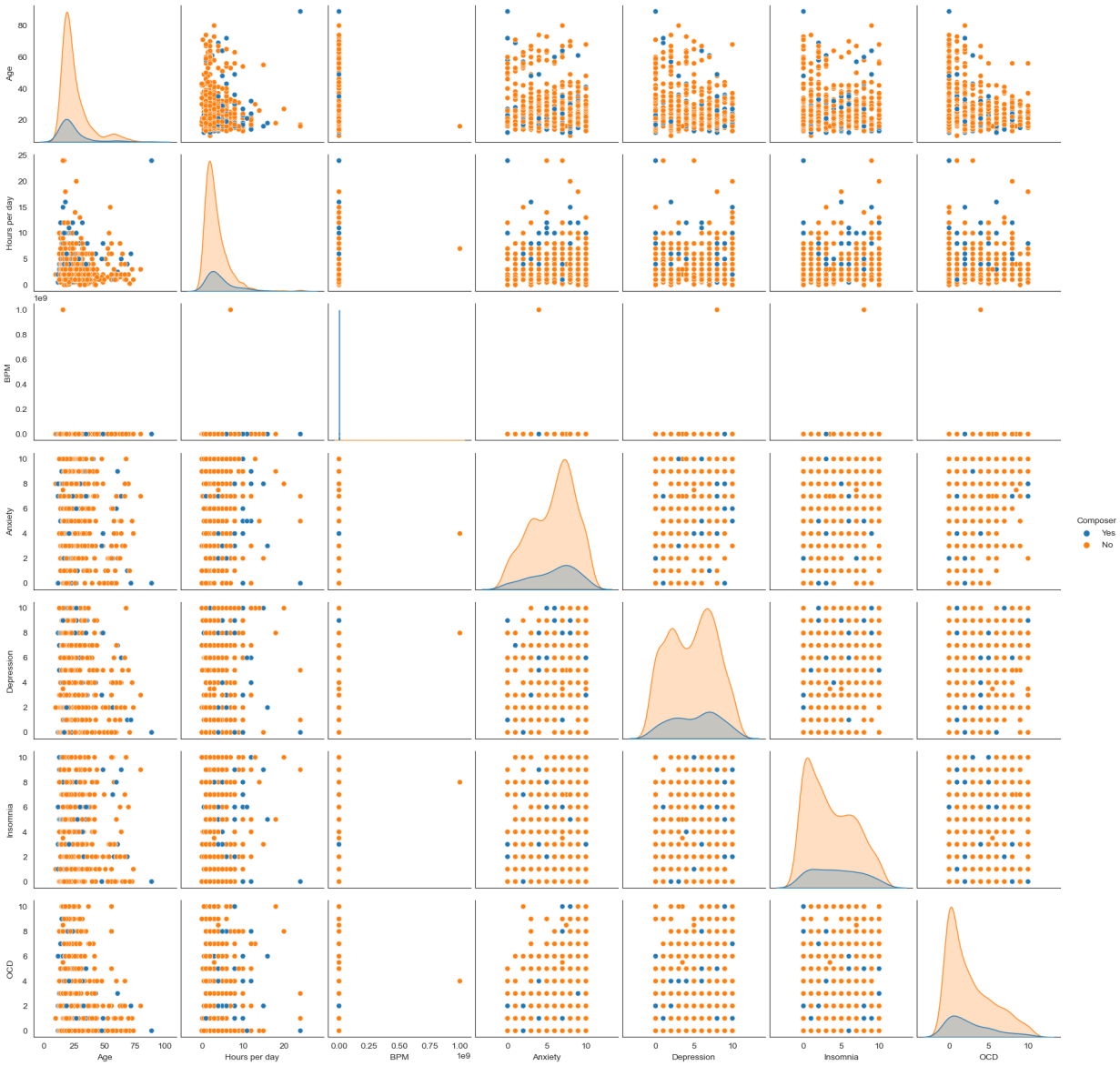
```
sns.lmplot( data = music, x='Hours per day',y='Age' ,hue='Fav genre')

plt.show()
```



19-Scatterplot Matrix

In [143…

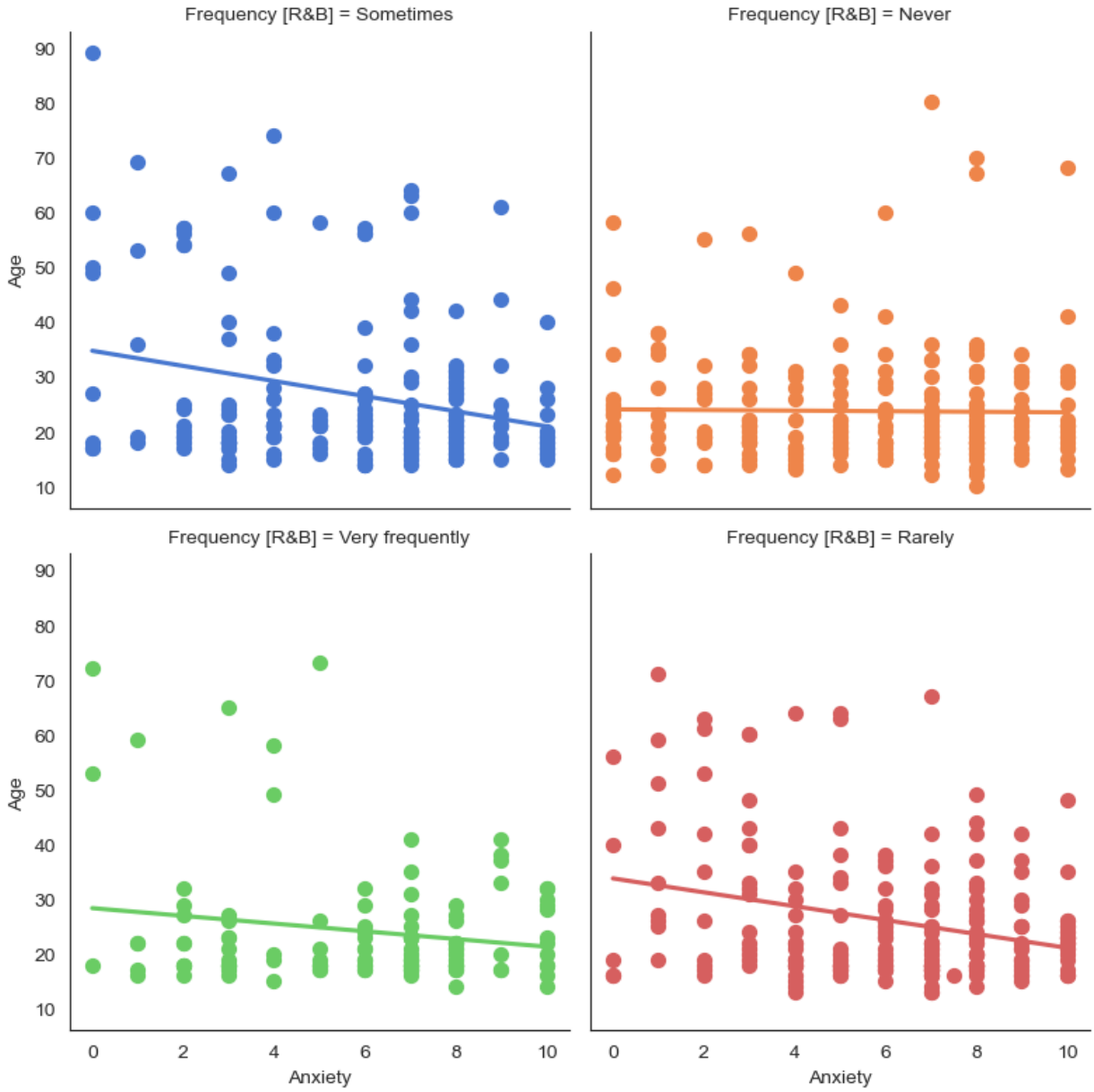
```
sns.pairplot(music, hue="Composer")
plt.show()
```



20-Anscombe's quartet

In [144…

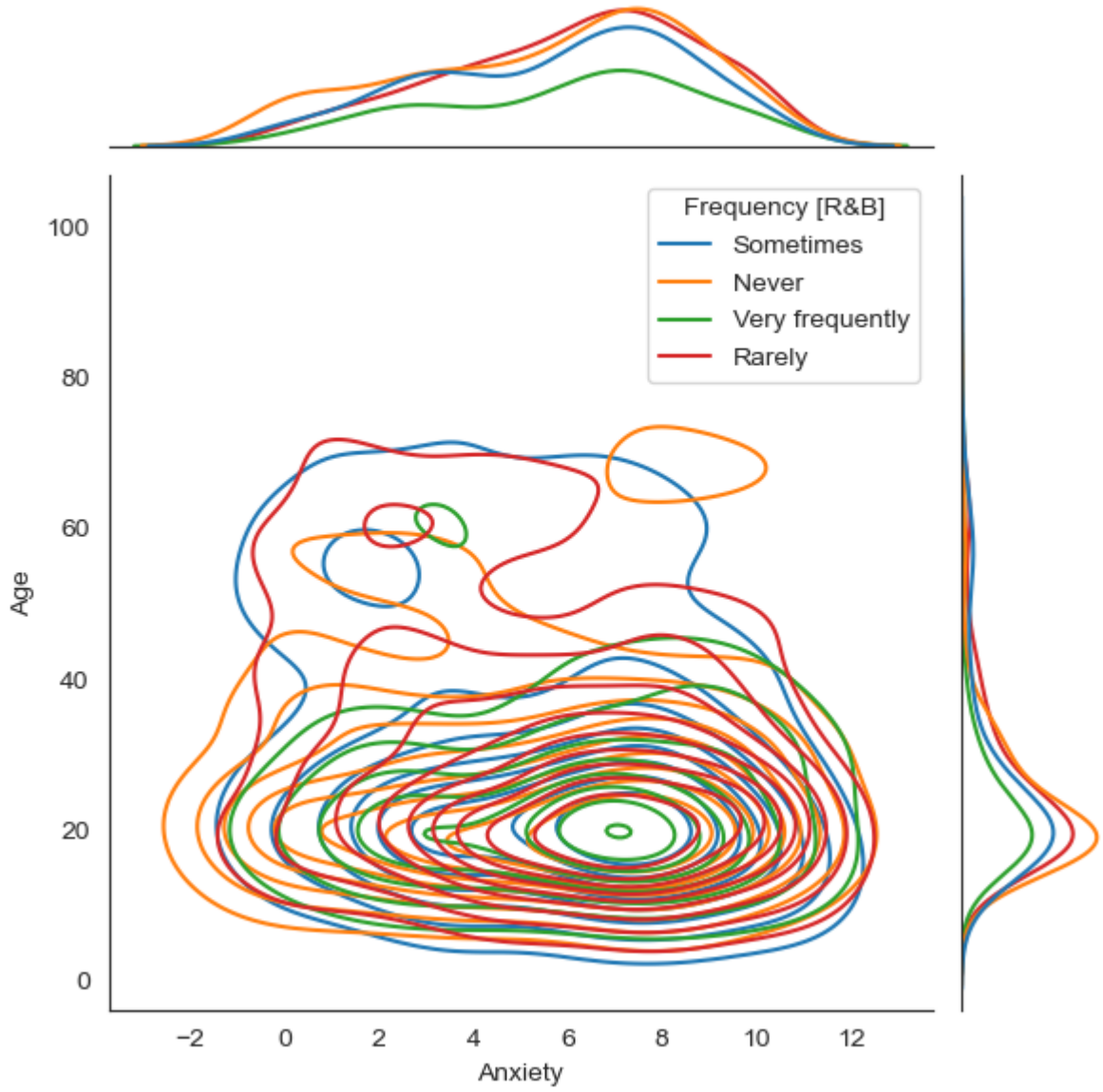
```
sns.lmplot(data=music, x="Anxiety", y="Age", col="Frequency [R&B]", hue="Frequency [R&
plt.show()
```



21-Joint kernel density estimate

In [147…

```
sns.jointplot(data=music,x="Anxiety", y="Age", hue="Frequency [R&B]", kind="kde")
plt.show()
```


Frequency [R&B]
Sometimes
Never
Very frequently
Rarely
Age
Anxiety
100
80
60
40
20
0
−2
0
2
4
6
8
10
12

In [ ]: