

In [3]:

```
from pandasai import PandasAI
from pandasai.llm.openai import OpenAI
import pandas as pd
```

In [2]:

```
df = pd.read_csv("Netflix Userbase.csv")
```

In [4]:

```
llm = OpenAI(api_token="<YOUR_OPENAI_TOKEN>")
```

In [5]:

```
pandas_ai = PandasAI(llm=llm)
```

In [6]:

```
pandas_ai(df, "show the first 5 records")
```

Out[6]:

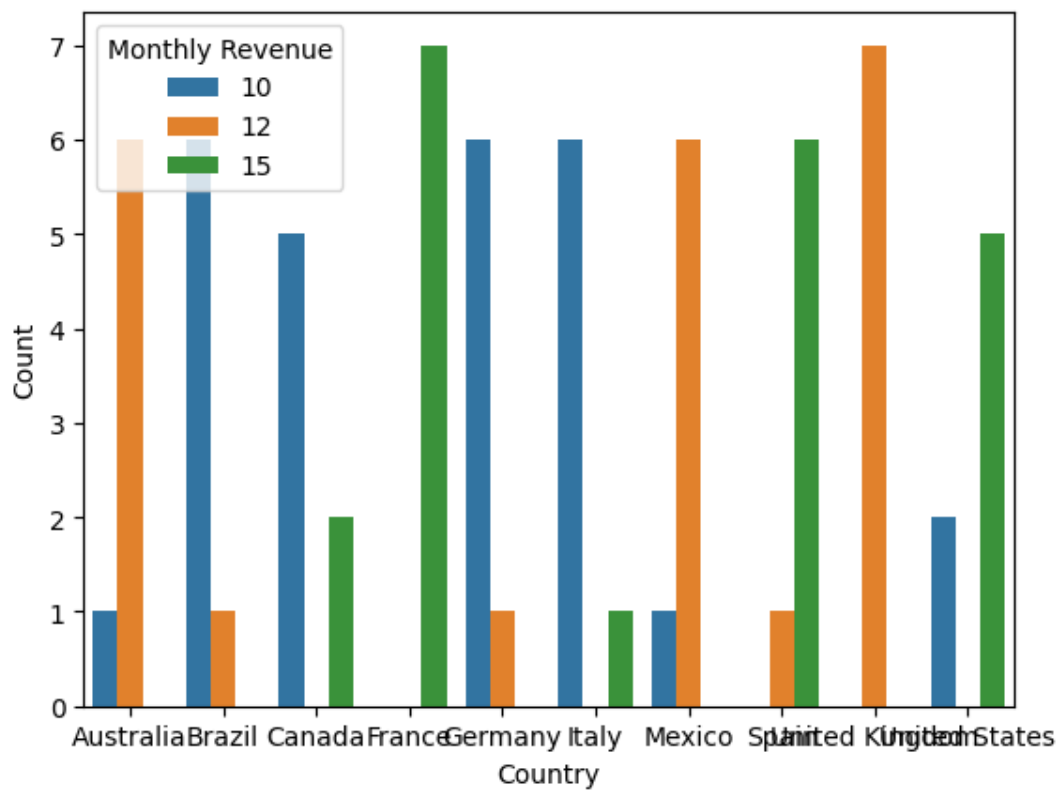
	User ID	Subscription Type	Monthly Revenue	Join Date	Last Payment Date	Country	Age	Gender	Device	Plan Duration
0	1	Basic	10	2022-01-15	2023-06-10	United States	28	Male	Smartphone	1 Month
1	2	Premium	15	2021-09-05	2023-06-22	Canada	35	Female	Tablet	1 Month
2	3	Standard	12	2023-02-28	2023-06-27	United Kingdom	42	Male	Smart TV	1 Month
3	4	Standard	12	2022-07-10	2023-06-26	Australia	51	Female	Laptop	1 Month
4	5	Basic	10	2023-05-01	2023-06-28	Germany	33	Male	Smartphone	1 Month

In [13]:

```
pandas_ai(df, "count the subscriptions group by Country and monthly revenue and plot \
by seaborn library and rotate the lables by 90 degree")
```

Out[13]:

..

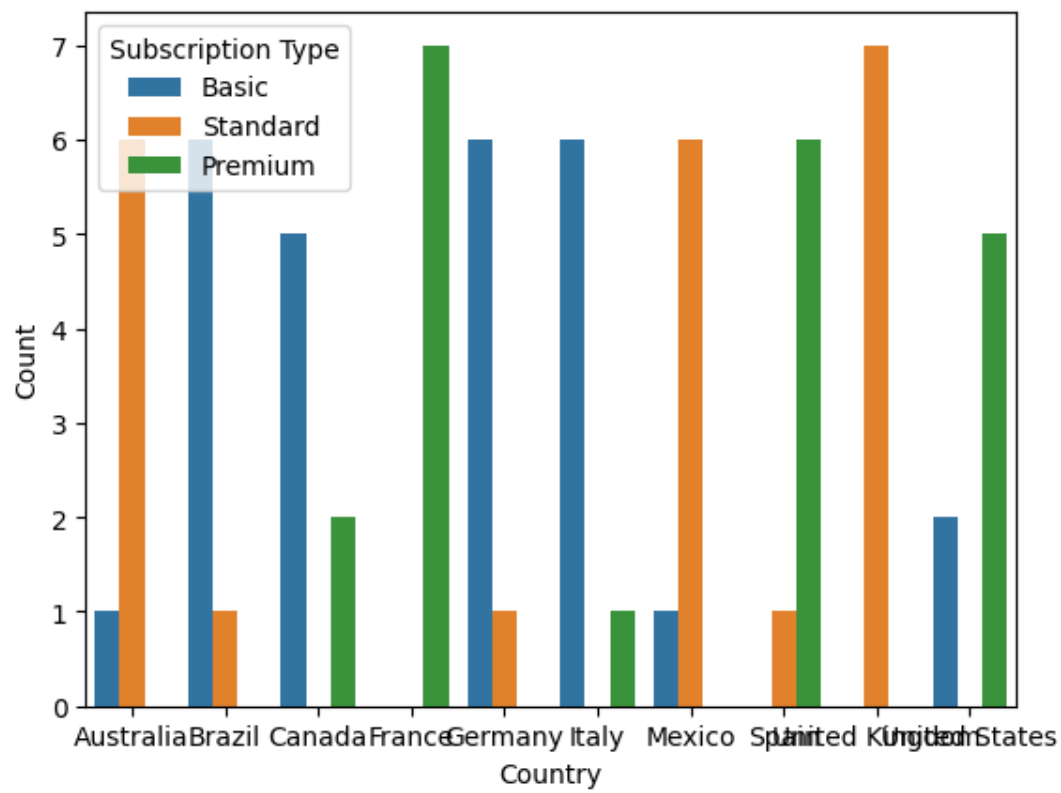


In [12]:

```
pandas_ai(df, "count the subscriptions group by Country & Subscription Type and \
plot the bar graph using seaborn library and rotate the x lables by 45 degree")
```

Out[12]:

..

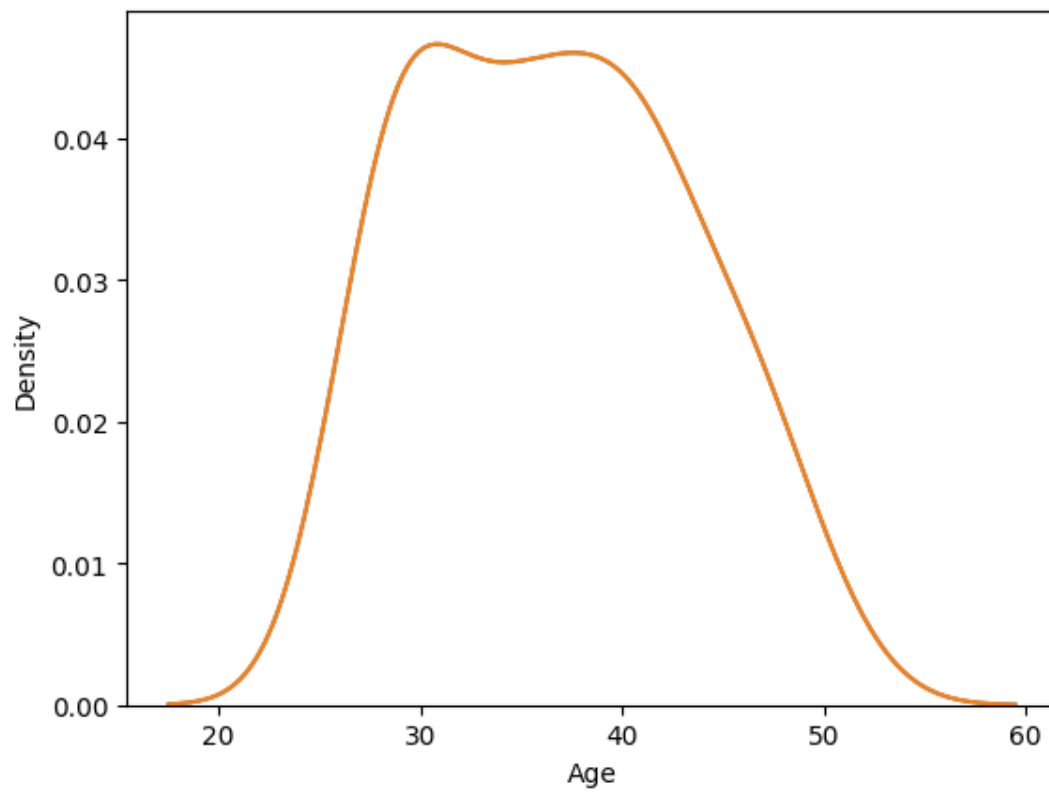


In [16]:

```
pandas_ai(df, "plot the distribution KDE graph of the Age using seaborn")
```

Out[16]:

<Axes: xlabel='Age', ylabel='Density'>

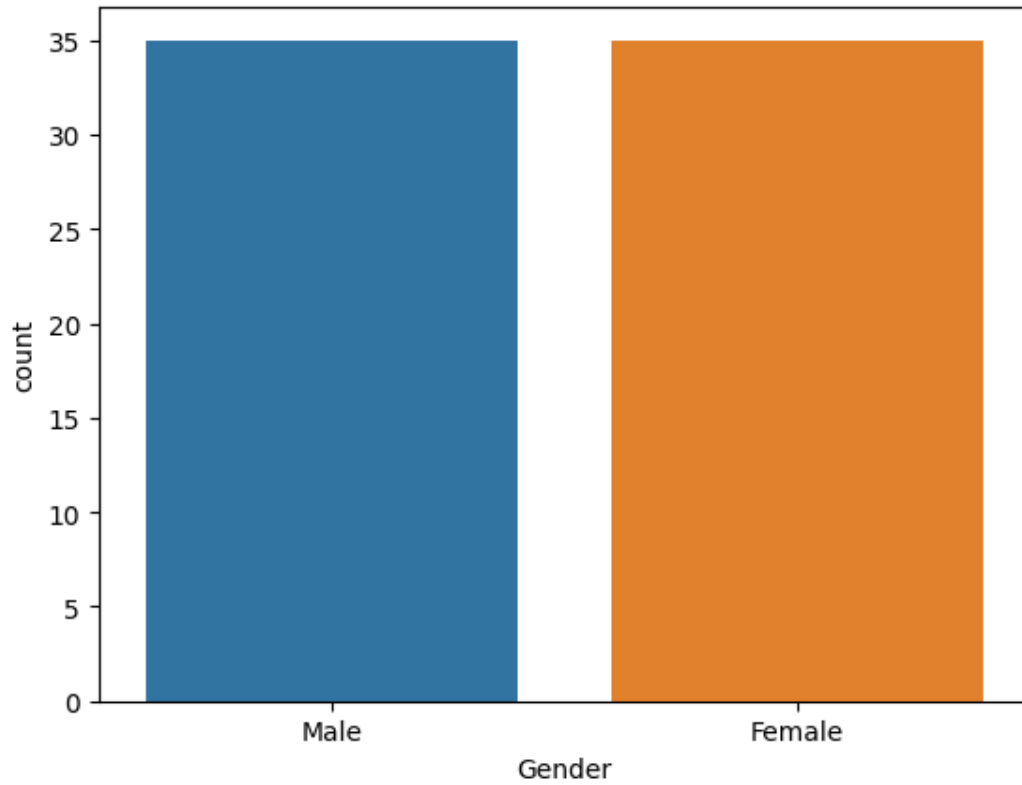


```
In [18]:
```

```
pandas_ai(df, "plot the count of Gender Male vs Female using seaborn also print the counts")
```

```
Out[18]:
```

```
Male      35  
Female    35  
Name: Gender, dtype: int64
```

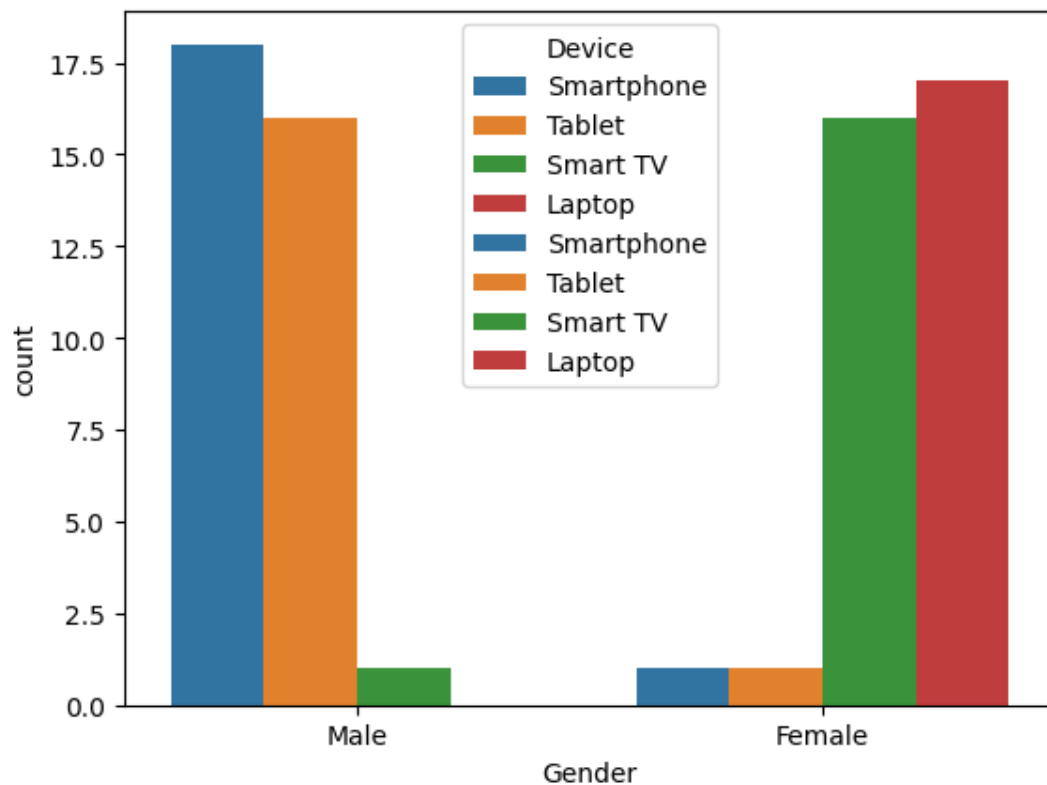


In [19]:

```
pandas_ai(df, "plot the visual group by gender and Device using seaborn")
```

Out[19]:

<Axes: xlabel='Gender', ylabel='count'>

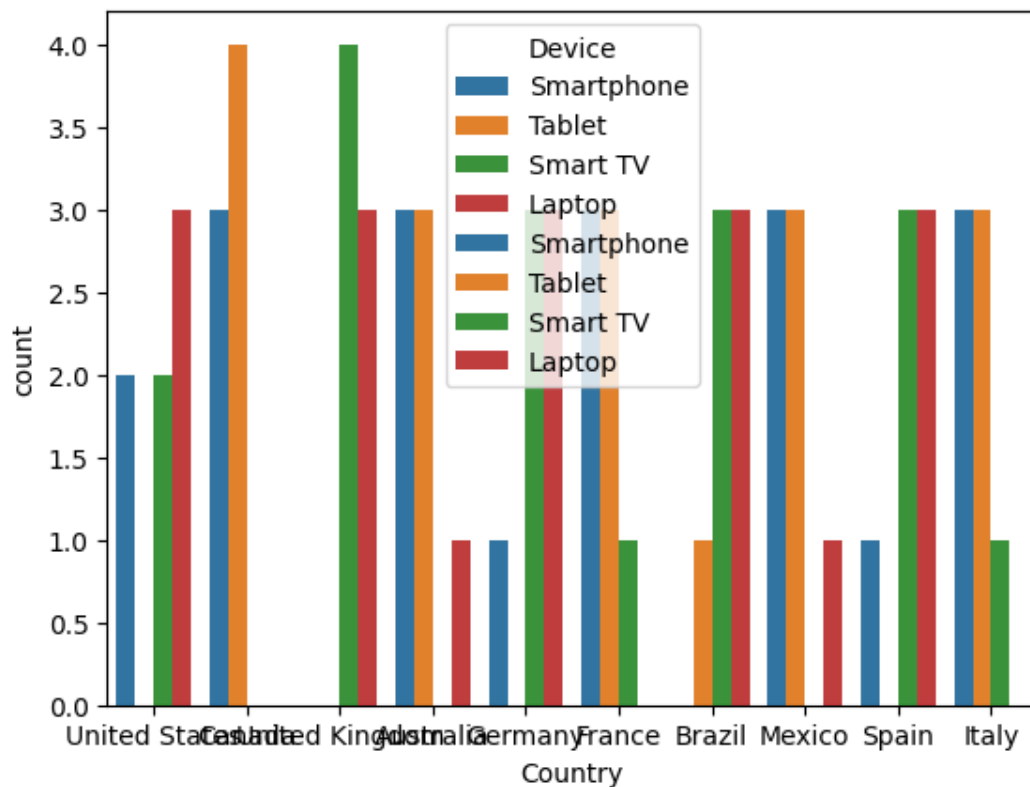


In [20]:

```
pandas_ai(df, "plot the visual group by country and Device using seaborn")
```

Out[20]:

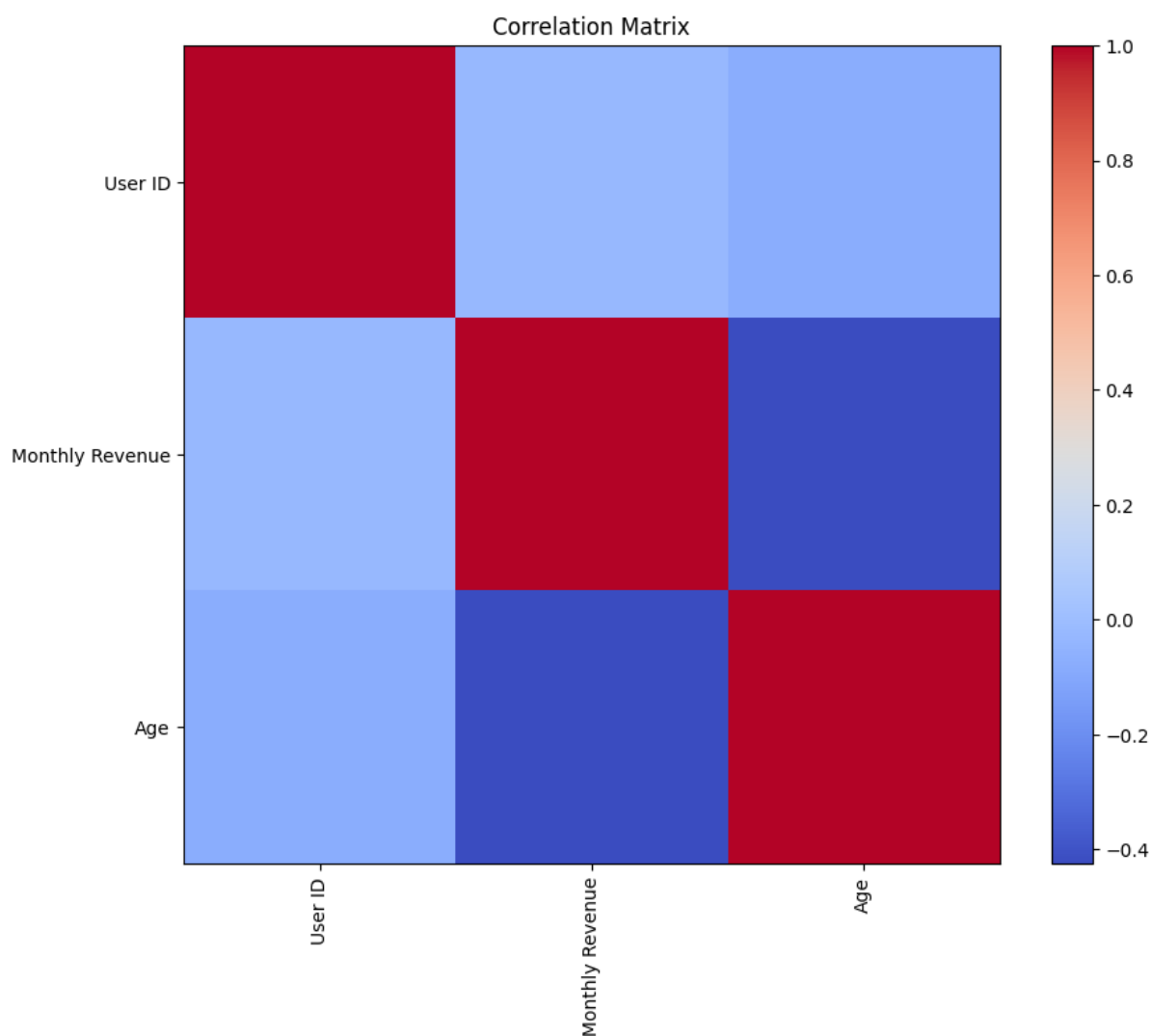
<Axes: xlabel='Country', ylabel='count'>



In [22]:

```
pandas_ai(df, "plot the correlation visual")
```

<string>:3: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid columns or specify the value of numeric_only to silence this warning.



In [23]:

```
pandas_ai(df, "are there any duplicates, NA, NULL, NAN available in data frame ?")
```

Out[23]:

'False False False'

In [24]:

```
pandas_ai(df, "describe the data frame")
```

Out[24]:

	User ID	Monthly Revenue	Age
count	70.000000	70.000000	70.000000
mean	35.500000	12.128571	36.600000
std	20.351085	2.070647	6.601273
min	1.000000	10.000000	26.000000
25%	18.250000	10.000000	31.000000
50%	35.500000	12.000000	36.000000
75%	52.750000	15.000000	41.000000
max	70.000000	15.000000	51.000000

In [25]:

```
pandas_ai(df, "what is the shape of the data frame ?")
```

Out[25]:

```
'70 10'
```

In []: