```
In [ ]:  # Import all modules needed.
         import pandas as pd
         import matplotlib.pyplot as plt
         import seaborn as sns
         import numpy as np
         import plotly.express as px
```

```
In [ ]:  # load data set from csv file
         df_shi = pd.read_csv('schizo_cleaned.csv')
         #print fist 5 rows of data
         df_shi.head(5)
```

Out[ ]:

	Name	gender	age-group	edu_lvl	Q1	Q2	Q3	Q4	Q5	Q6	...	Q10	Q11	Q12	Q13	Q14	Q1
0	Itba Zahid	female	16-20	College	1	0	0	0	1	2	...	2	2	1	2	1	
1	Ayesha Iftikhar	female	16-20	College	1	0	0	1	0	0	...	0	0	0	2	1	
2	Faiqa Niyamat	female	16-20	College	1	0	0	1	0	0	...	0	1	1	0	1	
3	Ishal Fatima	female	16-20	College	1	0	1	1	0	0	...	0	0	1	2	1	
4	Aqsa Imran	female	16-20	College	0	0	0	0	1	0	...	1	0	0	2	1	

5 rows × 23 columns

```
In [ ]:  # Get some info. on the dataset
         df_shi.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 601 entries, 0 to 600
Data columns (total 23 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   Name       601 non-null    object
 1   gender     601 non-null    object
 2   age-group  601 non-null    object
 3   edu_lvl    601 non-null    object
 4   Q1         601 non-null    int64
 5   Q2         601 non-null    int64
 6   Q3         601 non-null    int64
 7   Q4         601 non-null    int64
 8   Q5         601 non-null    int64
 9   Q6         601 non-null    int64
 10  Q7         601 non-null    int64
 11  Q8         601 non-null    int64
 12  Q9         601 non-null    int64
 13  Q10        601 non-null    int64
 14  Q11        601 non-null    int64
 15  Q12        601 non-null    int64
 16  Q13        601 non-null    int64
 17  Q14        601 non-null    int64
 18  Q15        601 non-null    int64
 19  Q16        601 non-null    int64
 20  Q17        601 non-null    int64
 21  Q18        601 non-null    int64
 22  Q19        601 non-null    int64
dtypes: int64(19), object(4)
memory usage: 108.1+ KB
```

In [ ]:
```python
# Get some descrition of the data.
df_shi.describe()
```

Out[ ]:

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
count	601.000000	601.000000	601.000000	601.000000	601.000000	601.000000	601.000000	601.000000
mean	0.818636	0.648918	0.760399	0.843594	0.850250	0.738769	0.737105	0.783694
std	0.561592	0.735892	0.697014	0.778136	0.759959	0.776731	0.777241	0.791497
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
50%	1.000000	0.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
75%	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
max	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000	2.000000

In [ ]:
```python
# shape of the Dataset
df_shi.shape
```

Out[ ]:
```
(601, 23)
```

In [ ]:
```python
df_shi['gender'].value_counts()
```

```
Out[ ]:   male      303
          female    298
          Name: gender, dtype: int64
```

```
In [ ]:   df_shi['Q1'].value_counts()
```

```
Out[ ]:   1    392
          0    159
          2     50
          Name: Q1, dtype: int64
```

```
In [ ]:   df_shi['Q2'].value_counts()
```

```
Out[ ]:   0    305
          1    202
          2     94
          Name: Q2, dtype: int64
```
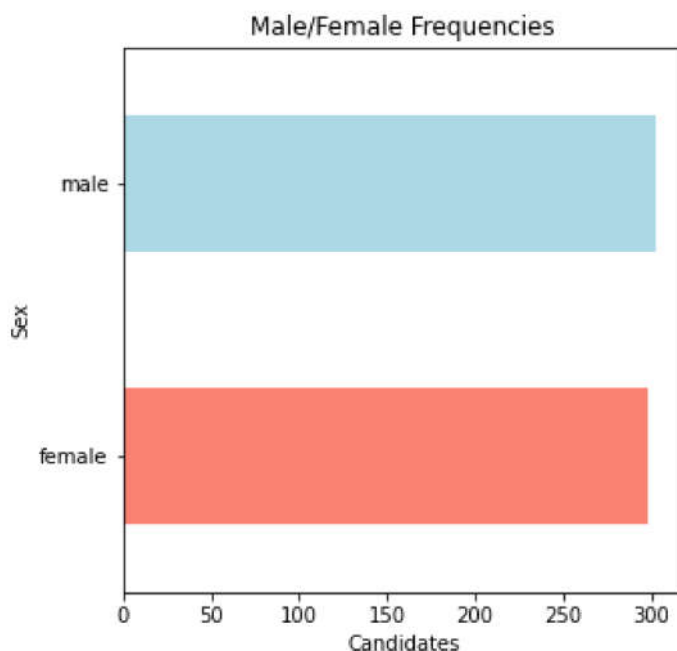
```
In [ ]:   a= df_shi['Q3'].value_counts()
          print(a)
```

```
          1    275
          0    235
          2     91
          Name: Q3, dtype: int64
```

```
In [ ]:   df_shi['Q3'].value_counts()
```

```
Out[ ]:   1    275
          0    235
          2     91
          Name: Q3, dtype: int64
```

```
In [ ]:   plt.figure(figsize=(5,5))
          figure_1 = df_shi["gender"].value_counts(ascending = True).plot.barh(color=["salmon",
          plt.title("Male/Female Frequencies")
          plt.ylabel("Sex")
          plt.xlabel("Candidates")
          plt.show()
```

In [ ]:
```python
# Death event per each sex
figure_3 = plt.bar(df_shi["Q1"].value_counts().index, df_shi["Q1"].value_counts(), col
# plt.title("Death Number per Gender")
plt.xlabel("Q1")
plt.ylabel("Count")
plt.show()
```
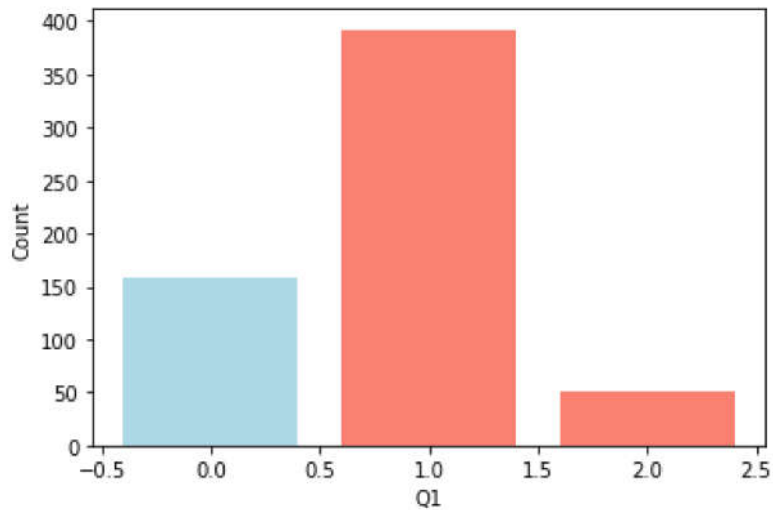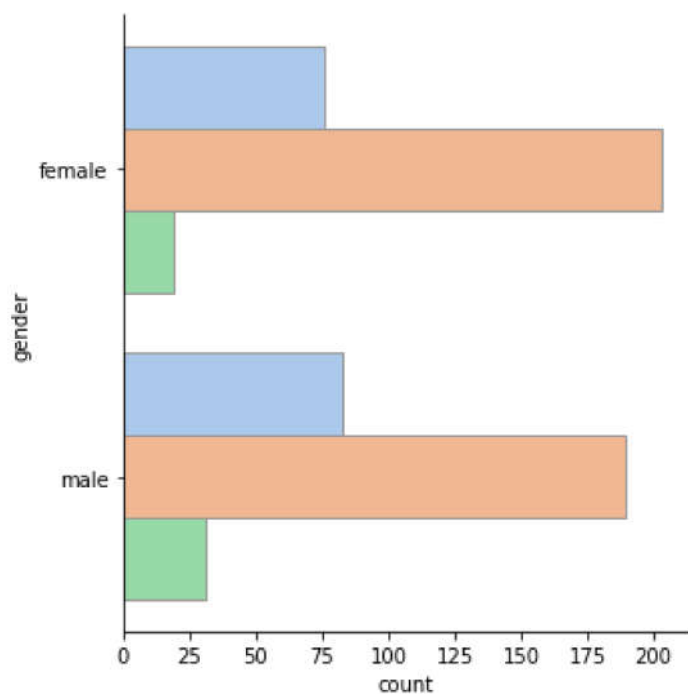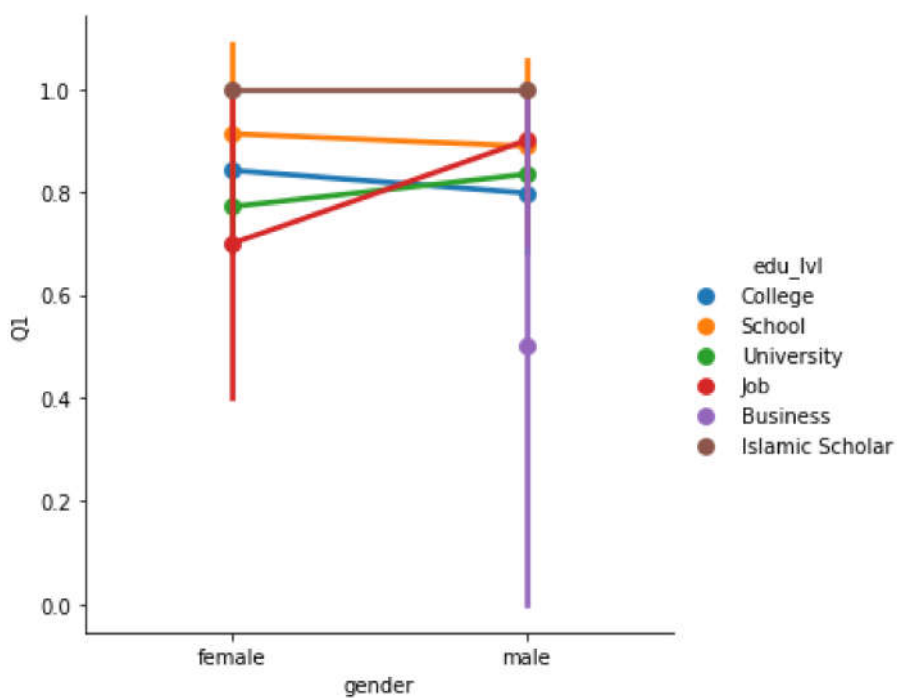

In [ ]:
```python
# Death event per each sex
figure_3 = plt.bar(df_shi["Q2"].value_counts().index, df_shi["Q2"].value_counts(), col
# plt.title("Death Number per Gender")
plt.xlabel("Q2")
plt.ylabel("Count")
plt.show()
```
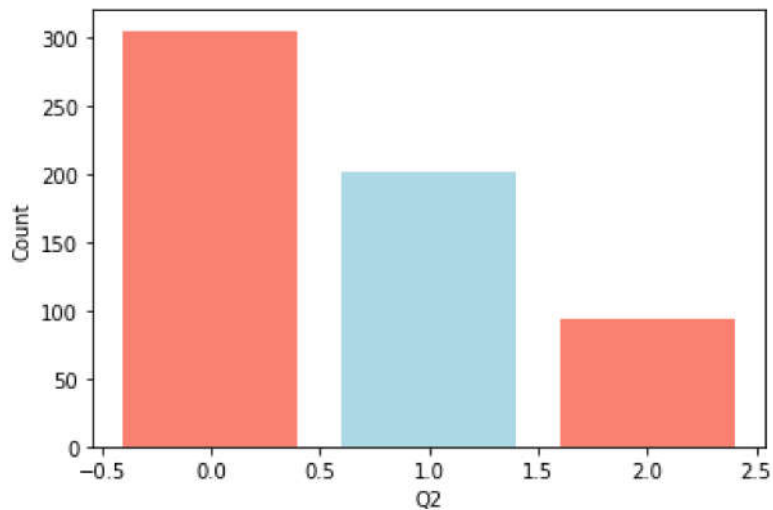

In [ ]:
```python
sns.catplot(y="gender", hue="Q1", kind="count",
            palette="pastel", edgecolor=".6",
            data=df_shi)
```

Out[ ]:
```
<seaborn.axisgrid.FacetGrid at 0x207944b2440>
```

```
In [ ]:  sns.catplot(x="gender", y="Q1", hue="edu_lvl", kind="point", data=df_shi)
```

Out[ ]: <seaborn.axisgrid.FacetGrid at 0x20796e96c80>
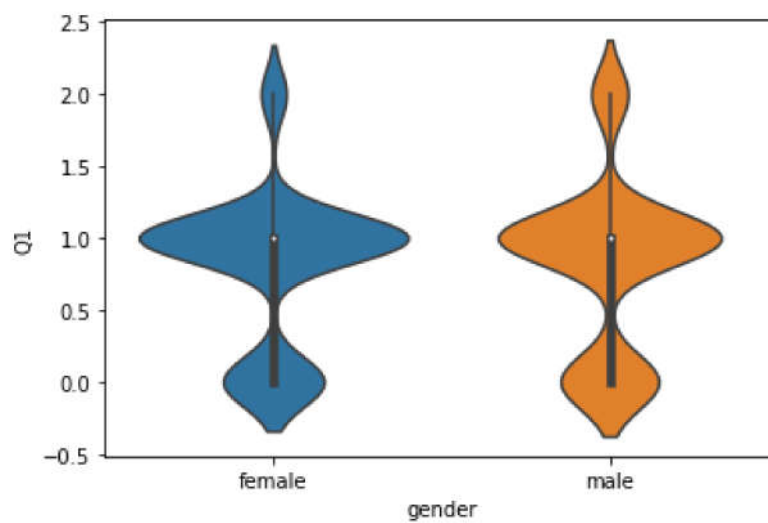


```
In [ ]:  sns.catplot(data=df_shi, orient="h", kind="box")
```

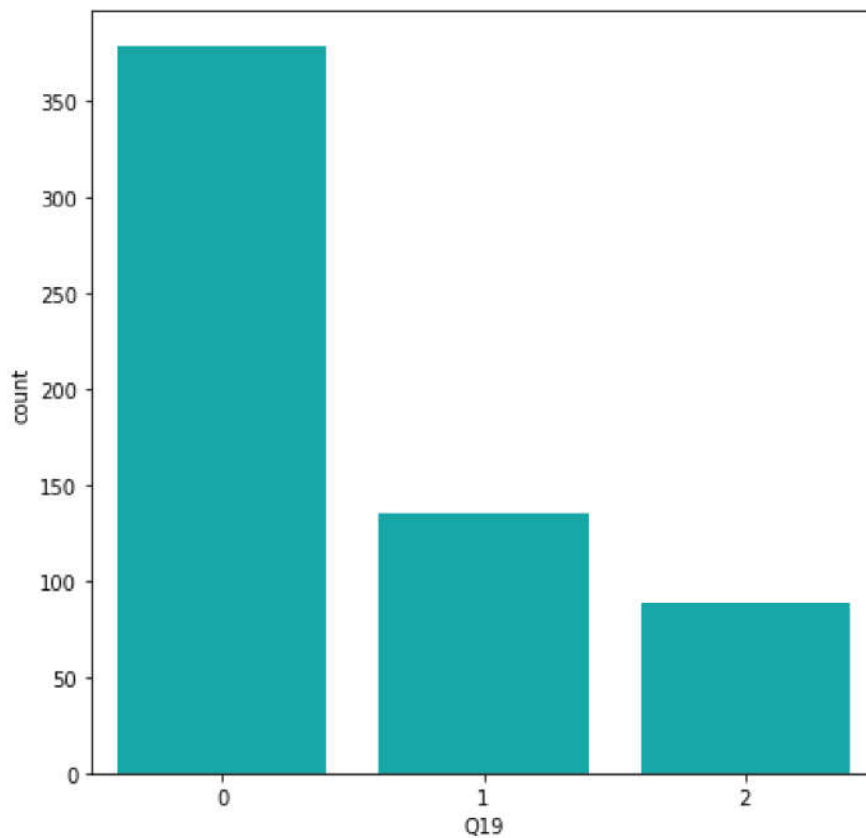Out[ ]: <seaborn.axisgrid.FacetGrid at 0x2079810f850>
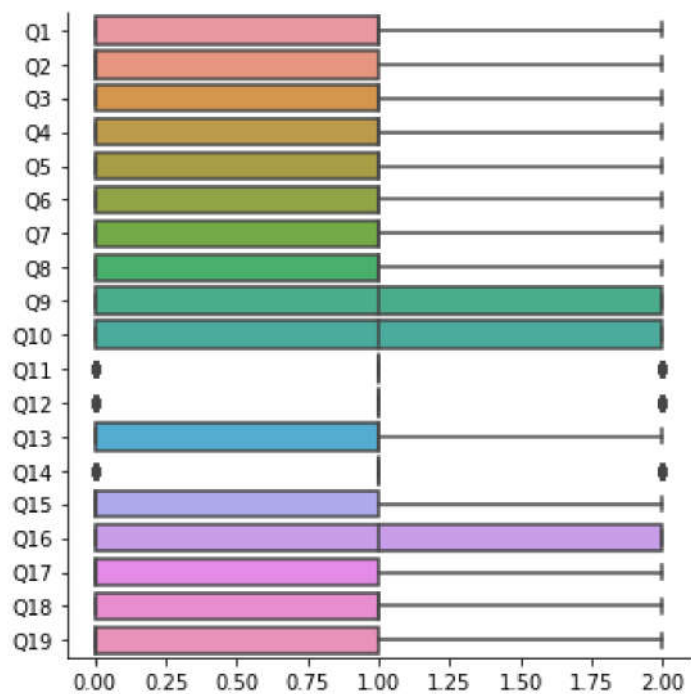
```
In [ ]:   sns.violinplot(x=df_shi.gender, y=df_shi.Q1)
```

```
Out[ ]:   <AxesSubplot:xlabel='gender', ylabel='Q1'>
```



```
In [ ]:   plt.subplots(figsize=(7, 7))
          sns.countplot(x="Q19", data=df_shi, color="c")
```

```
Out[ ]:   <AxesSubplot:xlabel='Q19', ylabel='count'>
```

```
In [ ]:   sns.catplot(x="gender", y="edu_lvl", hue="age-group",

                      kind="swarm", data=df_shi)
```

```
c:\Users\Administrator\AppData\Local\Programs\Python\Python310\lib\site-packages\seab
orn\categorical.py:1296: UserWarning: 72.8% of the points cannot be placed; you may w
ant to decrease the size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
c:\Users\Administrator\AppData\Local\Programs\Python\Python310\lib\site-packages\seab
orn\categorical.py:1296: UserWarning: 72.6% of the points cannot be placed; you may w
ant to decrease the size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
```

```
Out[ ]:   <seaborn.axisgrid.FacetGrid at 0x2079efdf550>
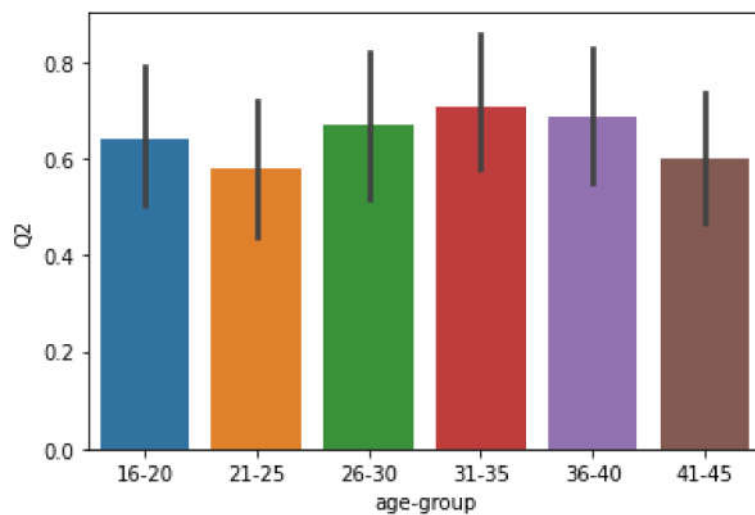```
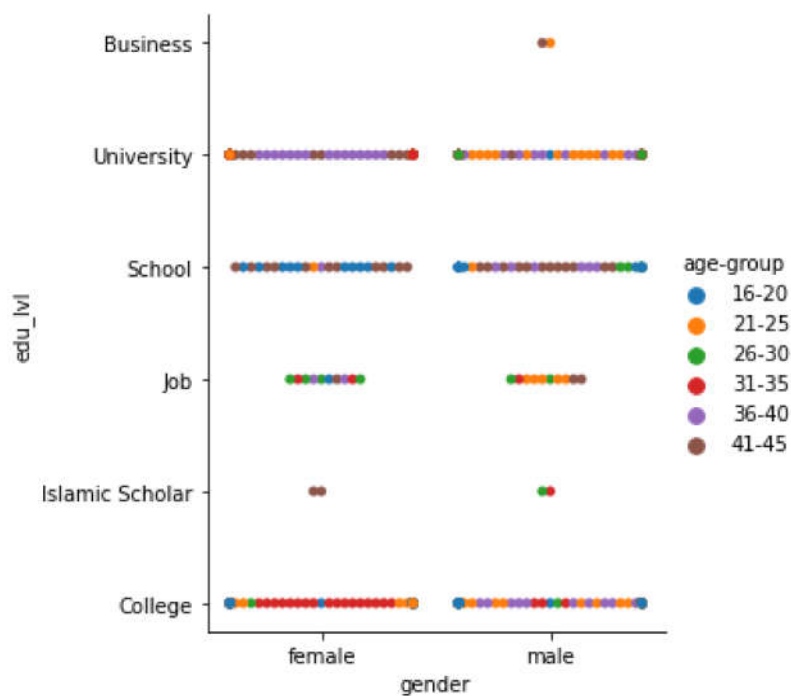
```
In [ ]:   sns.catplot(x="age-group", kind="count", palette="ch:.25", data=df_shi)
```
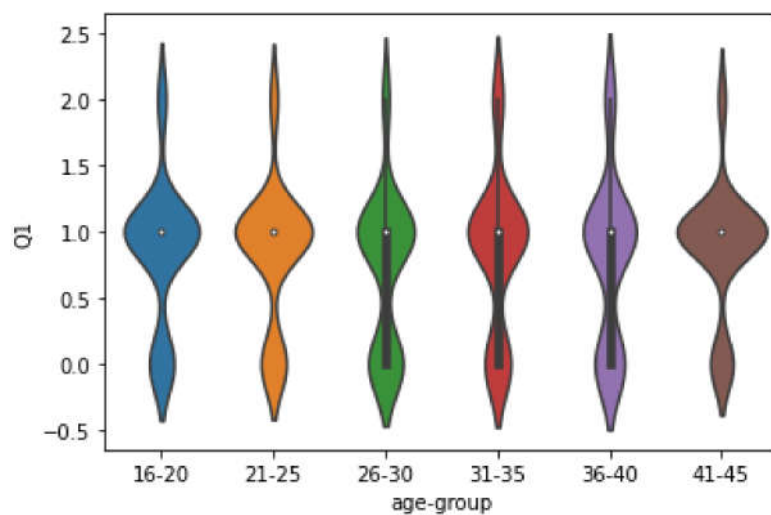
```
Out[ ]:   <seaborn.axisgrid.FacetGrid at 0x2079f01d2d0>
```
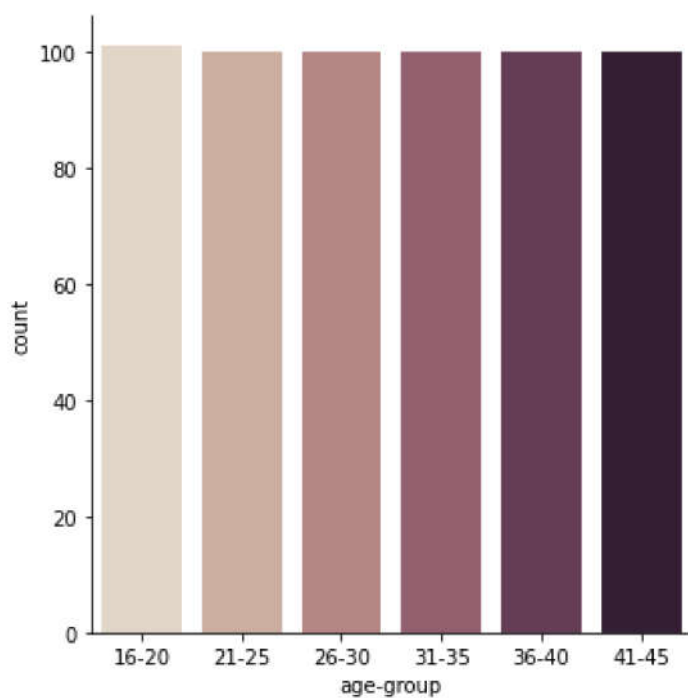


```
In [ ]:   sns.barplot(x='age-group',y='Q2',data=df_shi)
```

```
Out[ ]:   <AxesSubplot:xlabel='age-group', ylabel='Q2'>
```

```
In [ ]:   sns.violinplot(x='age-group', y='Q1',data=df_shi)
```

Out[ ]:   <AxesSubplot:xlabel='age-group', ylabel='Q1'>


```
In [ ]:   sns.stripplot(x='age-group',y='Q1',data=df_shi)
```

Out[ ]:   <AxesSubplot:xlabel='age-group', ylabel='Q1'>