DIVORCE EXPLORATORY DATA

May 19, 2023

[1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

[2]:
```
divo=pd.read_csv('divorce_data.csv')
```

[3]:
```
divo.head()
```

[3]:
```
   Q1  Q2  Q3  Q4  Q5  Q6  Q7  Q8  Q9  Q10  …  Q46  Q47  Q48  Q49  Q50  Q51  \
0   2   2   4   1   0   0   0   0   0    0  …    2    1    3    3    3    2
1   4   4   4   4   4   0   0   4   4    4  …    2    2    3    4    4    4
2   2   2   2   2   1   3   2   1   1    2  …    3    2    3    1    1    1
3   3   2   3   2   3   3   3   3   3    3  …    2    2    3    3    3    3
4   2   2   1   1   1   1   0   0   0    0  …    2    1    2    3    2    2

   Q52  Q53  Q54  Divorce
0    3    2    1        1
1    4    2    2        1
2    2    2    2        1
3    2    2    2        1
4    2    1    0        1

[5 rows x 55 columns]
```

[4]:
```
divo.tail()
```

[4]:
```
     Q1  Q2  Q3  Q4  Q5  Q6  Q7  Q8  Q9  Q10  …  Q46  Q47  Q48  Q49  Q50  \
165   0   0   0   0   0   0   0   0   0    0  …    1    0    4    1    1
166   0   0   0   0   0   0   0   0   0    0  …    4    1    2    2    2
167   1   1   0   0   0   0   0   0   0    1  …    3    0    2    0    1
168   0   0   0   0   0   0   0   0   0    0  …    3    3    2    2    3
169   0   0   0   0   0   0   0   1   0    0  …    3    4    4    0    1

     Q51  Q52  Q53  Q54  Divorce
165    4    2    2    2        0
```

```
166    2    3    2    2         0
167    1    3    0    0         0
168    2    4    3    1         0
169    3    3    3    1         0

[5 rows x 55 columns]
```

[5]:
```
divo.shape
```

[5]:
```
(170, 55)
```

[6]:
```
divo.columns
```

[6]:
```
Index(['Q1', 'Q2', 'Q3', 'Q4', 'Q5', 'Q6', 'Q7', 'Q8', 'Q9', 'Q10', 'Q11',
       'Q12', 'Q13', 'Q14', 'Q15', 'Q16', 'Q17', 'Q18', 'Q19', 'Q20', 'Q21',
       'Q22', 'Q23', 'Q24', 'Q25', 'Q26', 'Q27', 'Q28', 'Q29', 'Q30', 'Q31',
       'Q32', 'Q33', 'Q34', 'Q35', 'Q36', 'Q37', 'Q38', 'Q39', 'Q40', 'Q41',
       'Q42', 'Q43', 'Q44', 'Q45', 'Q46', 'Q47', 'Q48', 'Q49', 'Q50', 'Q51',
       'Q52', 'Q53', 'Q54', 'Divorce'],
      dtype='object')
```

[7]:
```
divo.duplicated().sum()
```

[7]:
```
20
```

[8]:
```
divo.isnull().sum()
```

[8]:
```
Q1          0
Q2          0
Q3          0
Q4          0
Q5          0
Q6          0
Q7          0
Q8          0
Q9          0
Q10         0
Q11         0
Q12         0
Q13         0
Q14         0
Q15         0
Q16         0
Q17         0
Q18         0
Q19         0
Q20         0
Q21         0
```

```
Q22        0
Q23        0
Q24        0
Q25        0
Q26        0
Q27        0
Q28        0
Q29        0
Q30        0
Q31        0
Q32        0
Q33        0
Q34        0
Q35        0
Q36        0
Q37        0
Q38        0
Q39        0
Q40        0
Q41        0
Q42        0
Q43        0
Q44        0
Q45        0
Q46        0
Q47        0
Q48        0
Q49        0
Q50        0
Q51        0
Q52        0
Q53        0
Q54        0
Divorce    0
dtype: int64
```

[9]:

```
with open('divorce.txt') as f:
    contents = f.read()
    print(contents)
```

```
atribute_id|description
1|If one of us apologizes when our discussion deteriorates, the discussion ends.
2|I know we can ignore our differences, even if things get hard sometimes.
3|When we need it, we can take our discussions with my spouse from the beginning
and correct it.
4|When I discuss with my spouse, to contact him will eventually work.
5|The time I spent with my wife is special for us.
6|We don't have time at home as partners.
```

7|We are like two strangers who share the same environment at home rather than family.

8|I enjoy our holidays with my wife.

9|I enjoy traveling with my wife.

10|Most of our goals are common to my spouse.

11|I think that one day in the future, when I look back, I see that my spouse and I have been in harmony with each other.

12|My spouse and I have similar values in terms of personal freedom.

13|My spouse and I have similar sense of entertainment.

14|Most of our goals for people (children, friends, etc.) are the same.

15|Our dreams with my spouse are similar and harmonious.

16|We're compatible with my spouse about what love should be.

17|We share the same views about being happy in our life with my spouse

18|My spouse and I have similar ideas about how marriage should be

19|My spouse and I have similar ideas about how roles should be in marriage

20|My spouse and I have similar values in trust.

21|I know exactly what my wife likes.

22|I know how my spouse wants to be taken care of when she/he sick.

23|I know my spouse's favorite food.

24|I can tell you what kind of stress my spouse is facing in her/his life.

25|I have knowledge of my spouse's inner world.

26|I know my spouse's basic anxieties.

27|I know what my spouse's current sources of stress are.

28|I know my spouse's hopes and wishes.

29|I know my spouse very well.

30|I know my spouse's friends and their social relationships.

31|I feel aggressive when I argue with my spouse.

32|When discussing with my spouse, I usually use expressions such as ‘you always’ or ‘you never’ .

33|I can use negative statements about my spouse's personality during our discussions.

34|I can use offensive expressions during our discussions.

35|I can insult my spouse during our discussions.

36|I can be humiliating when we discussions.

37|My discussion with my spouse is not calm.

38|I hate my spouse's way of open a subject.

39|Our discussions often occur suddenly.

40|We're just starting a discussion before I know what's going on.

41|When I talk to my spouse about something, my calm suddenly breaks.

42|When I argue with my spouse, ? only go out and I don't say a word.

43|I mostly stay silent to calm the environment a little bit.

44|Sometimes I think it's good for me to leave home for a while.

45|I'd rather stay silent than discuss with my spouse.

46|Even if I'm right in the discussion, I stay silent to hurt my spouse.

47|When I discuss with my spouse, I stay silent because I am afraid of not being able to control my anger.

48|I feel right in our discussions.

49|I have nothing to do with what I've been accused of.

```
50|I'm not actually the one who's guilty about what I'm accused of.
51|I'm not the one who's wrong about problems at home.
52|I wouldn't hesitate to tell my spouse about her/his inadequacy.
53|When I discuss, I remind my spouse of her/his inadequacy.
54|I'm not afraid to tell my spouse about her/his incompetence.
```

[10]:
```
divo.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 170 entries, 0 to 169
Data columns (total 55 columns):
 #   Column   Non-Null Count  Dtype
---  ------   --------------  -----
 0   Q1       170 non-null    int64
 1   Q2       170 non-null    int64
 2   Q3       170 non-null    int64
 3   Q4       170 non-null    int64
 4   Q5       170 non-null    int64
 5   Q6       170 non-null    int64
 6   Q7       170 non-null    int64
 7   Q8       170 non-null    int64
 8   Q9       170 non-null    int64
 9   Q10      170 non-null    int64
 10  Q11      170 non-null    int64
 11  Q12      170 non-null    int64
 12  Q13      170 non-null    int64
 13  Q14      170 non-null    int64
 14  Q15      170 non-null    int64
 15  Q16      170 non-null    int64
 16  Q17      170 non-null    int64
 17  Q18      170 non-null    int64
 18  Q19      170 non-null    int64
 19  Q20      170 non-null    int64
 20  Q21      170 non-null    int64
 21  Q22      170 non-null    int64
 22  Q23      170 non-null    int64
 23  Q24      170 non-null    int64
 24  Q25      170 non-null    int64
 25  Q26      170 non-null    int64
 26  Q27      170 non-null    int64
 27  Q28      170 non-null    int64
 28  Q29      170 non-null    int64
 29  Q30      170 non-null    int64
 30  Q31      170 non-null    int64
 31  Q32      170 non-null    int64
 32  Q33      170 non-null    int64
 33  Q34      170 non-null    int64
 34  Q35      170 non-null    int64
 35  Q36      170 non-null    int64
 36  Q37      170 non-null    int64
 37  Q38      170 non-null    int64
 38  Q39      170 non-null    int64
 39  Q40      170 non-null    int64
 40  Q41      170 non-null    int64
 41  Q42      170 non-null    int64
 42  Q43      170 non-null    int64
 43  Q44      170 non-null    int64
 44  Q45      170 non-null    int64
 45  Q46      170 non-null    int64
 46  Q47      170 non-null    int64
 47  Q48      170 non-null    int64
 48  Q49      170 non-null    int64
 49  Q50      170 non-null    int64
 50  Q51      170 non-null    int64
 51  Q52      170 non-null    int64
 52  Q53      170 non-null    int64
 53  Q54      170 non-null    int64
 54  Divorce  170 non-null    int64
dtypes: int64(55)
memory usage: 73.2 KB
```

```
[11]: divo.describe()
```

```
[11]:               Q1          Q2          Q3          Q4          Q5          Q6  \
       count  170.000000  170.000000  170.000000  170.000000  170.000000  170.000000
       mean     1.776471    1.652941    1.764706    1.482353    1.541176    0.747059
       std      1.627257    1.468654    1.415444    1.504327    1.632169    0.904046
       min      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
       50%      2.000000    2.000000    2.000000    1.000000    1.000000    0.000000
       75%      3.000000    3.000000    3.000000    3.000000    3.000000    1.000000
       max      4.000000    4.000000    4.000000    4.000000    4.000000    4.000000

                      Q7          Q8          Q9         Q10  …         Q46  \
       count  170.000000  170.000000  170.000000  170.000000  …  170.000000
       mean     0.494118    1.452941    1.458824    1.576471  …    2.552941
       std      0.898698    1.546371    1.557976    1.421529  …    1.371786
       min      0.000000    0.000000    0.000000    0.000000  …    0.000000
       25%      0.000000    0.000000    0.000000    0.000000  …    2.000000
       50%      0.000000    1.000000    1.000000    2.000000  …    3.000000
       75%      1.000000    3.000000    3.000000    3.000000  …    4.000000
       max      4.000000    4.000000    4.000000    4.000000  …    4.000000

                     Q47         Q48         Q49         Q50         Q51         Q52  \
       count  170.000000  170.000000  170.000000  170.000000  170.000000  170.000000
       mean     2.270588    2.741176    2.382353    2.429412    2.476471    2.517647
       std      1.586841    1.137348    1.511587    1.405090    1.260238    1.476537
       min      0.000000    0.000000    0.000000    0.000000    0.000000    0.000000
       25%      1.000000    2.000000    1.000000    1.000000    2.000000    1.000000
       50%      2.000000    3.000000    3.000000    2.000000    3.000000    3.000000
       75%      4.000000    4.000000    4.000000    4.000000    4.000000    4.000000
       max      4.000000    4.000000    4.000000    4.000000    4.000000    4.000000

                     Q53         Q54     Divorce
       count  170.000000  170.000000  170.000000
       mean     2.241176    2.011765    0.494118
       std      1.505634    1.667611    0.501442
       min      0.000000    0.000000    0.000000
       25%      1.000000    0.000000    0.000000
       50%      2.000000    2.000000    0.000000
       75%      4.000000    4.000000    1.000000
       max      4.000000    4.000000    1.000000

       [8 rows x 55 columns]
```

[12]:

```
divo.nunique()
```

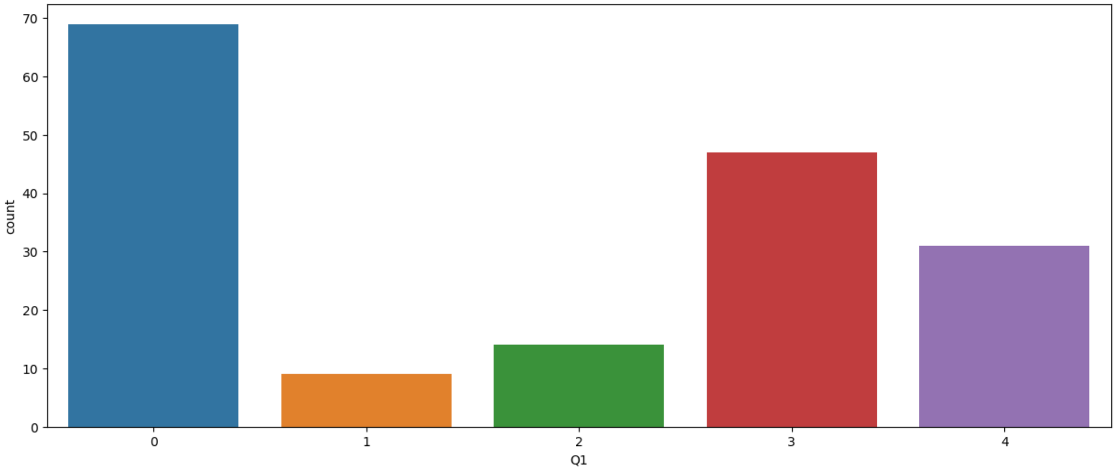
[12]:

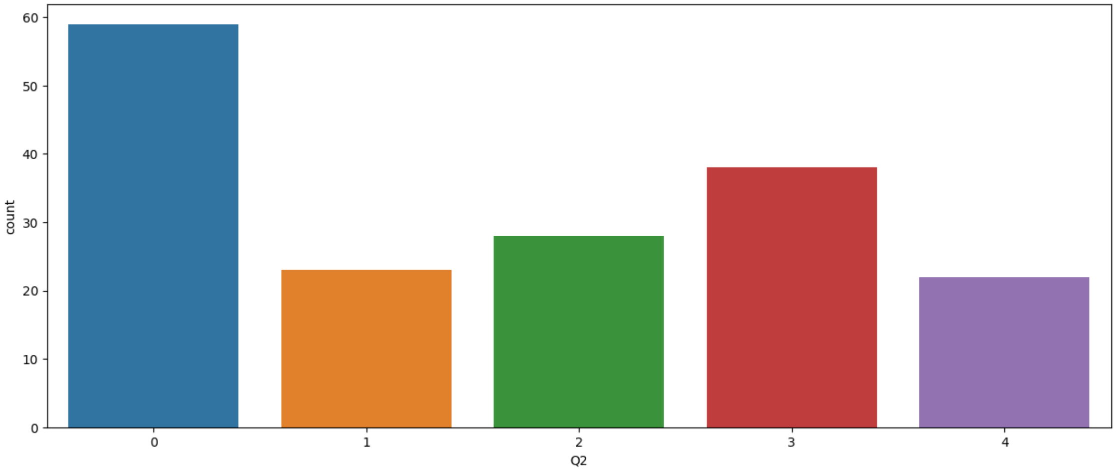
```
Q1          5
Q2          5
Q3          5
Q4          5
Q5          5
Q6          5
Q7          5
Q8          5
Q9          5
Q10         5
Q11         5
Q12         5
Q13         5
Q14         5
Q15         5
Q16         5
Q17         5
Q18         5
Q19         5
Q20         5
Q21         5
Q22         5
Q23         5
Q24         5
```

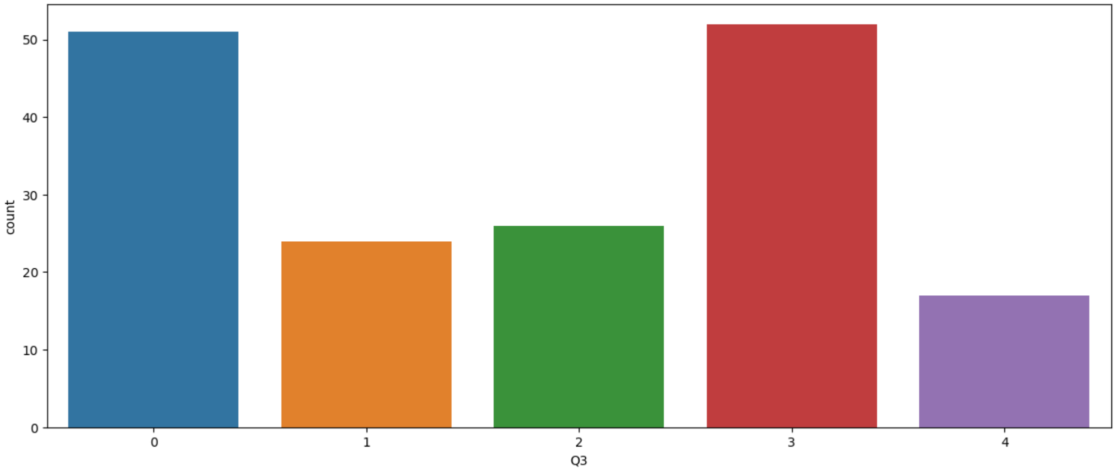
```
Q25        5
Q26        5
Q27        5
Q28        5
Q29        5
Q30        5
Q31        5
Q32        5
Q33        5
Q34        5
Q35        5
Q36        5
Q37        5
Q38        5
Q39        5
Q40        5
Q41        5
Q42        5
Q43        5
Q44        5
Q45        5
Q46        5
Q47        5
Q48        5
Q49        5
Q50        5
Q51        5
Q52        5
Q53        5
Q54        5
Divorce    2
dtype: int64
```

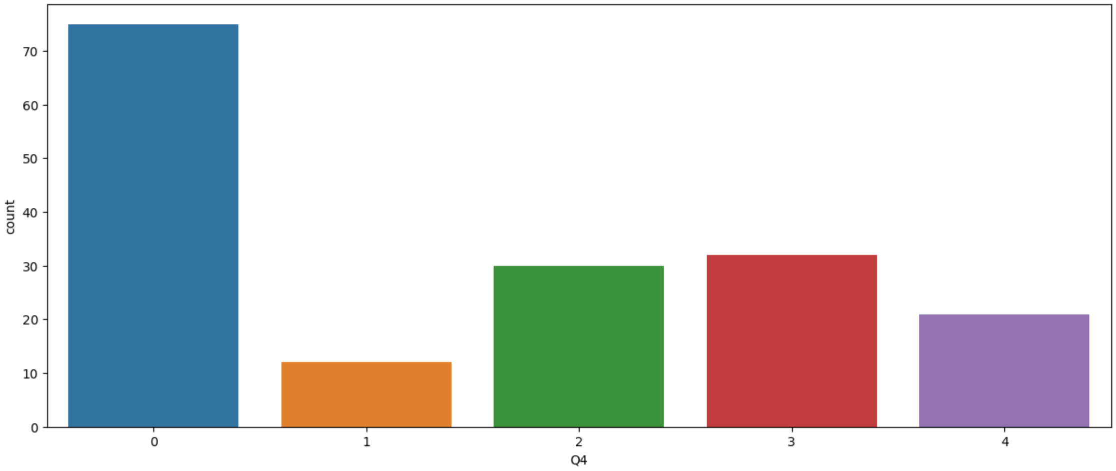
[13]:

```
for i in divo.columns:
    plt.figure(figsize=(15,6))
    sns.countplot(divo[i], data=divo)
    plt.show()
```

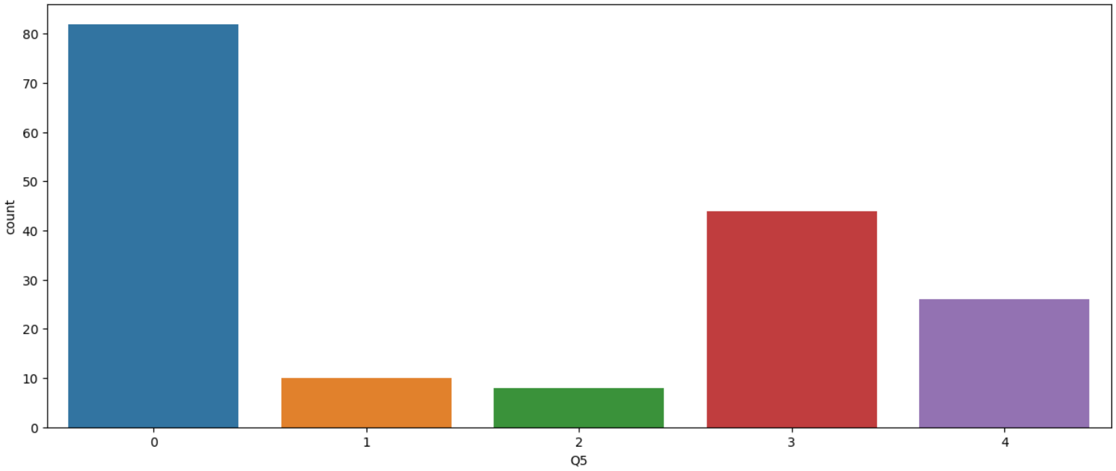

count
0
10
20
30
40
50
60
70
0
1
2
3
4
Q1


count
0
10
20
30
40
50
60
0
1
2
3
4
Q2

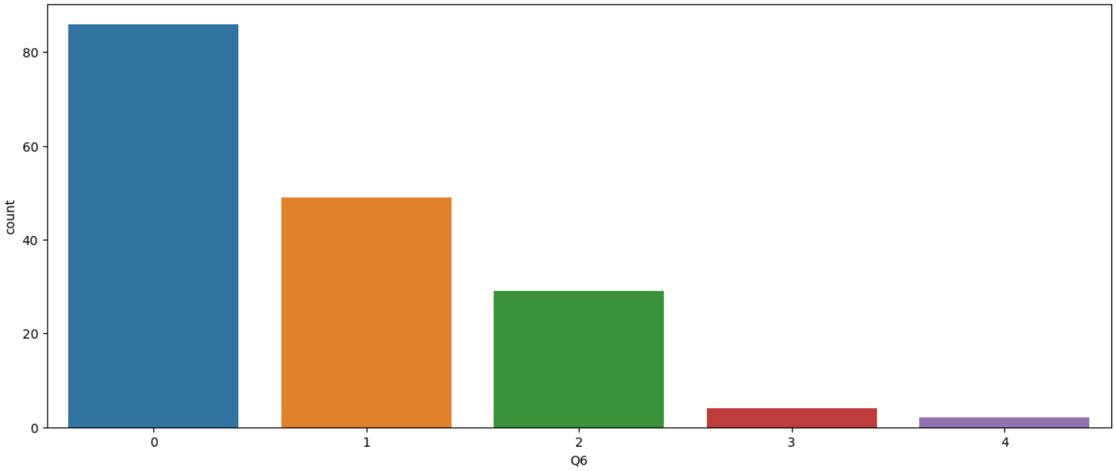

count
0
10
20
30
40
50
0
1
2
3
4
Q3



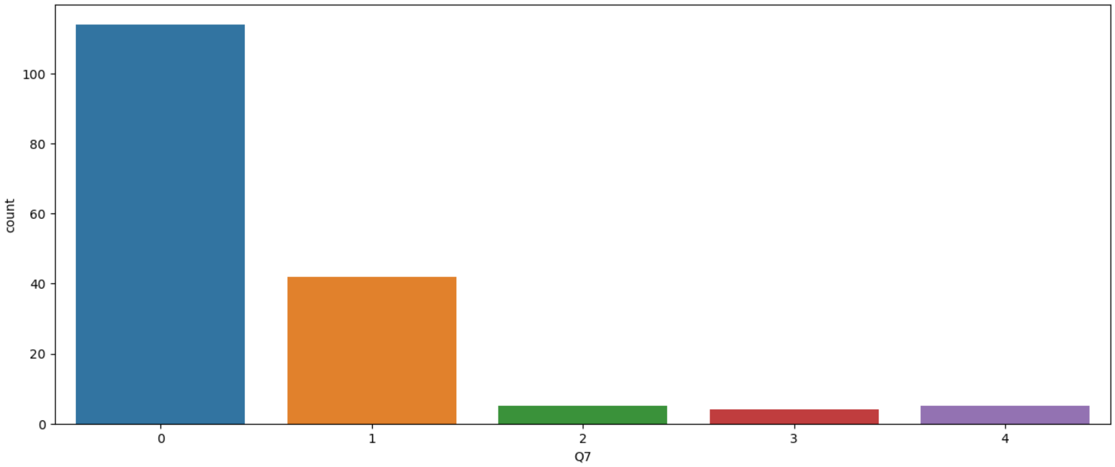
70
60
50
40
30
20
10
0
count
0
1
2
3
4
Q4

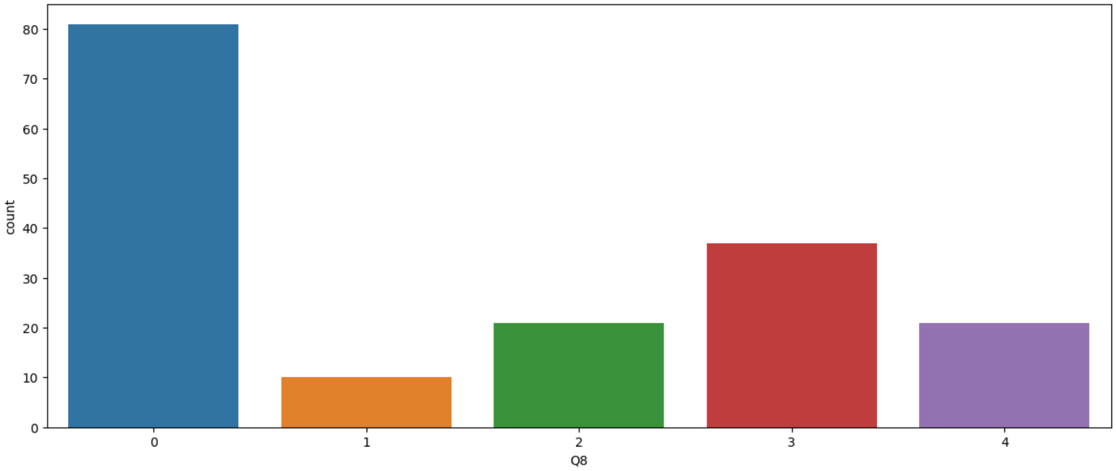


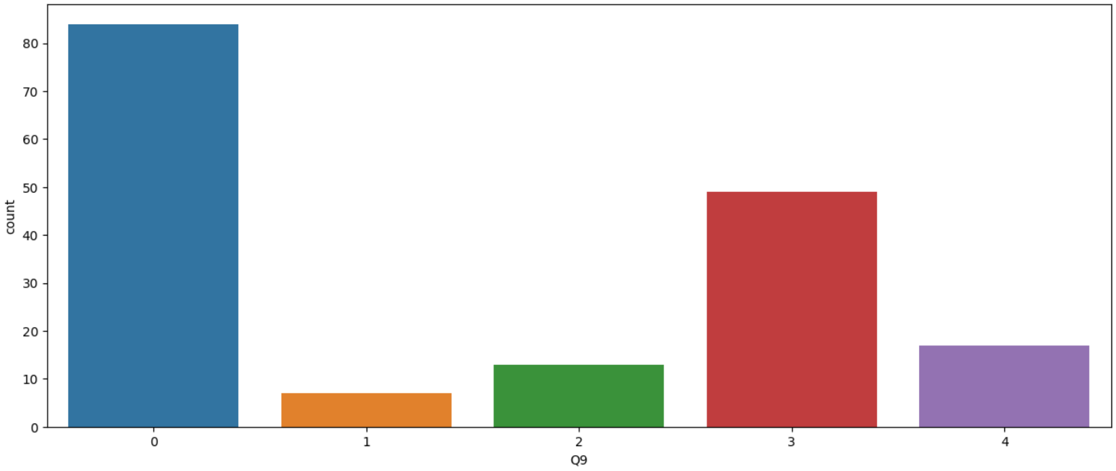
80
70
60
50
40
30
20
10
0
count
0
1
2
3
4
Q5

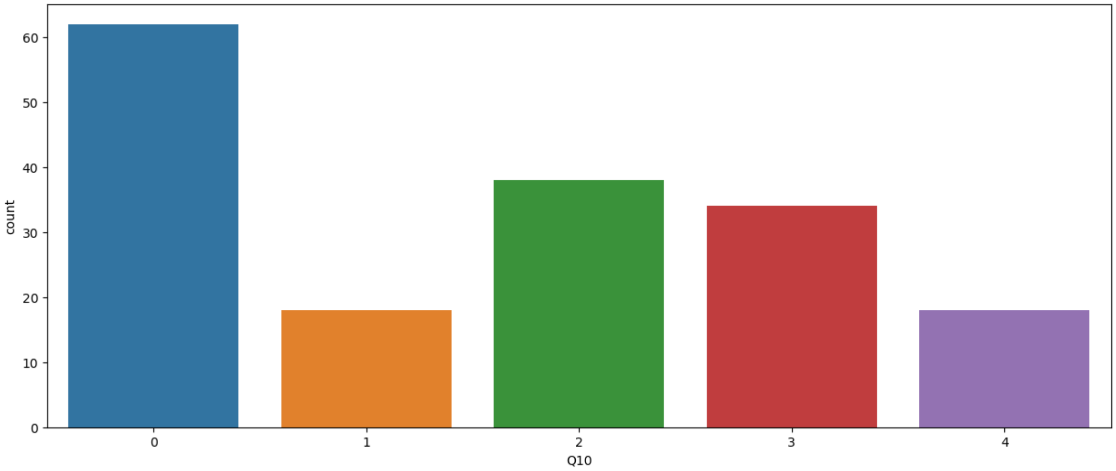


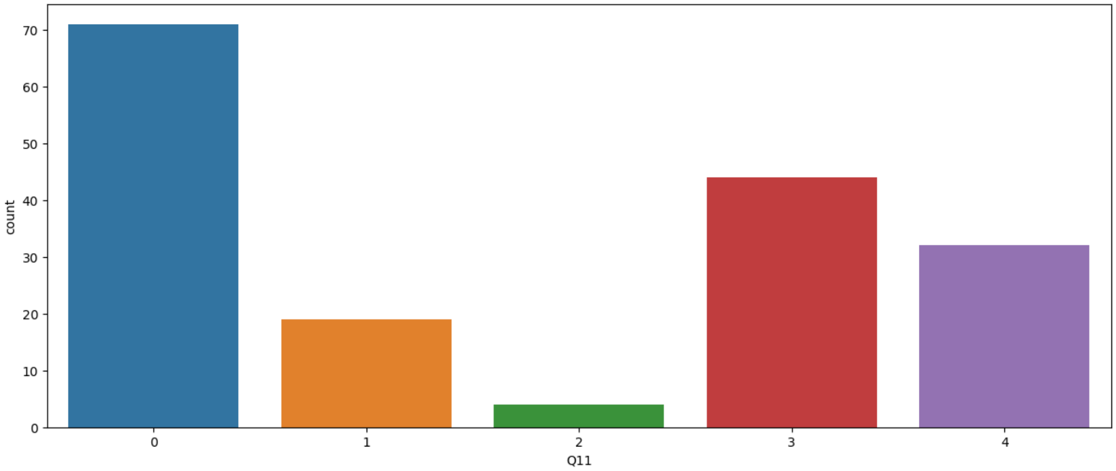
80
60
40
20
0
count
0
1
2
3
4
Q6

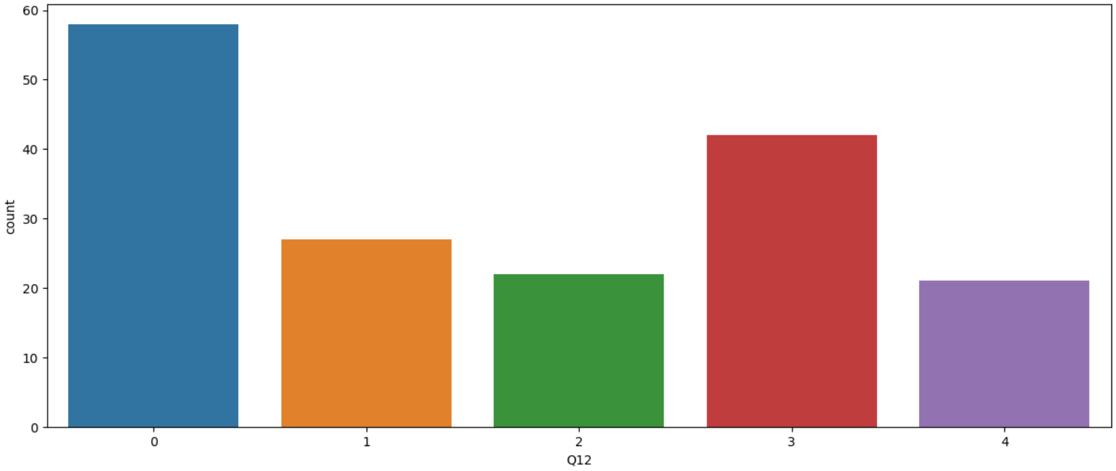

count
0
20
40
60
80
100
0
1
2
3
4
Q7

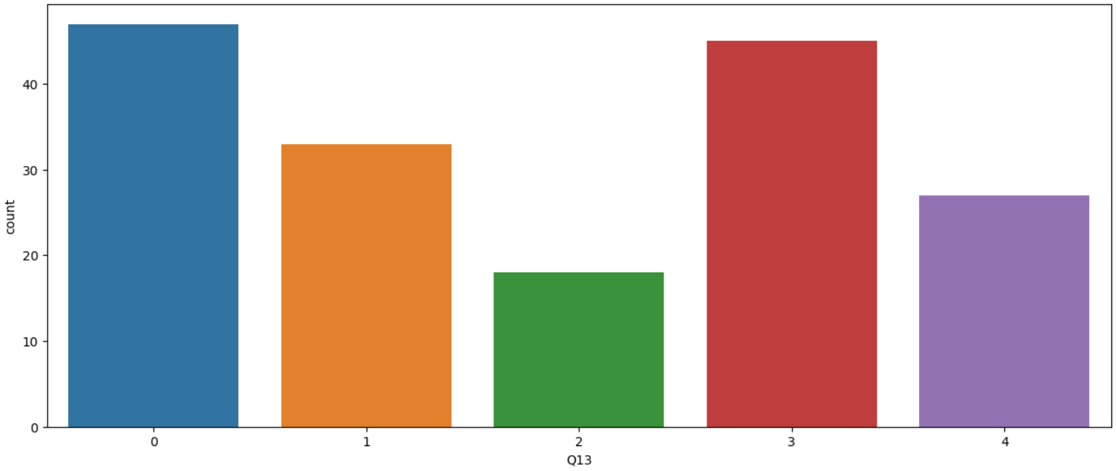

count
0
10
20
30
40
50
60
70
80
0
1
2
3
4
Q8

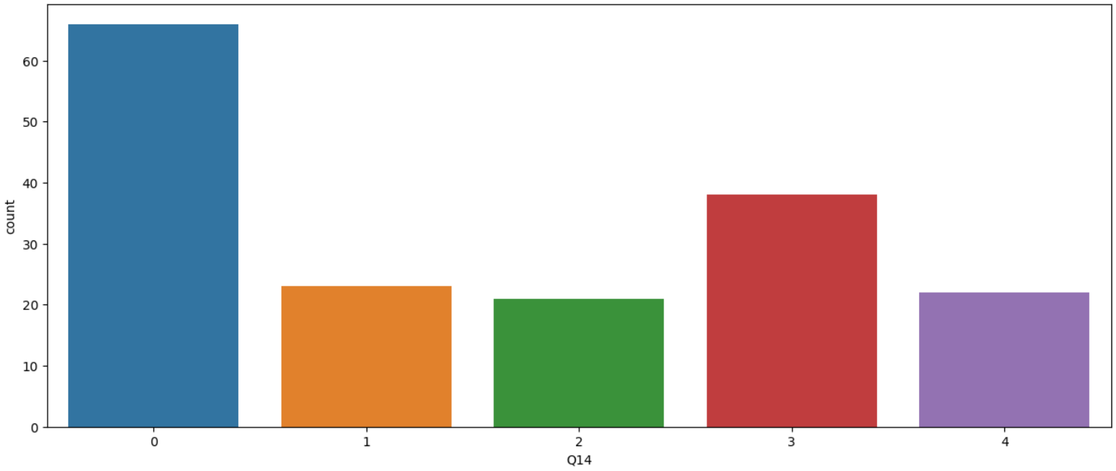

count
0
10
20
30
40
50
60
70
80
0
1
2
3
4
Q9

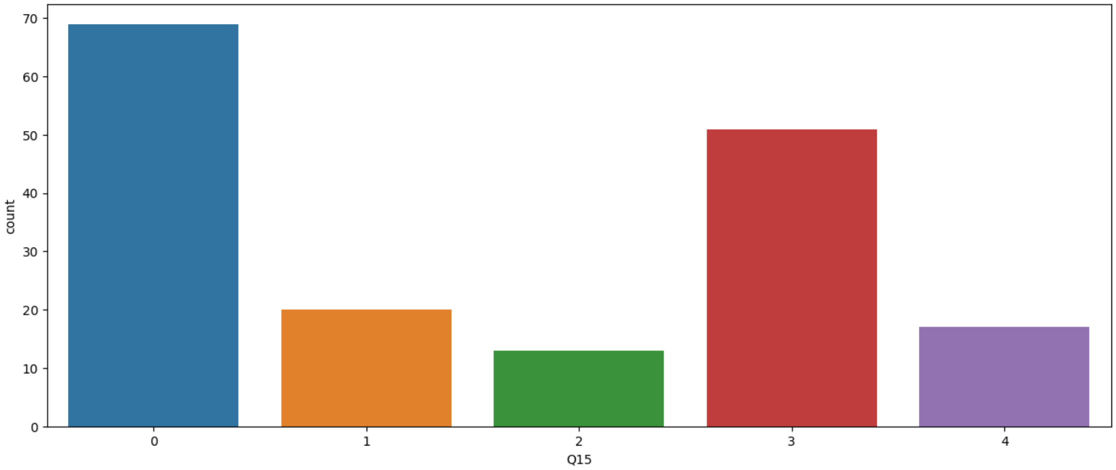

count
0
10
20
30
40
50
60
0
1
2
3
4
Q10

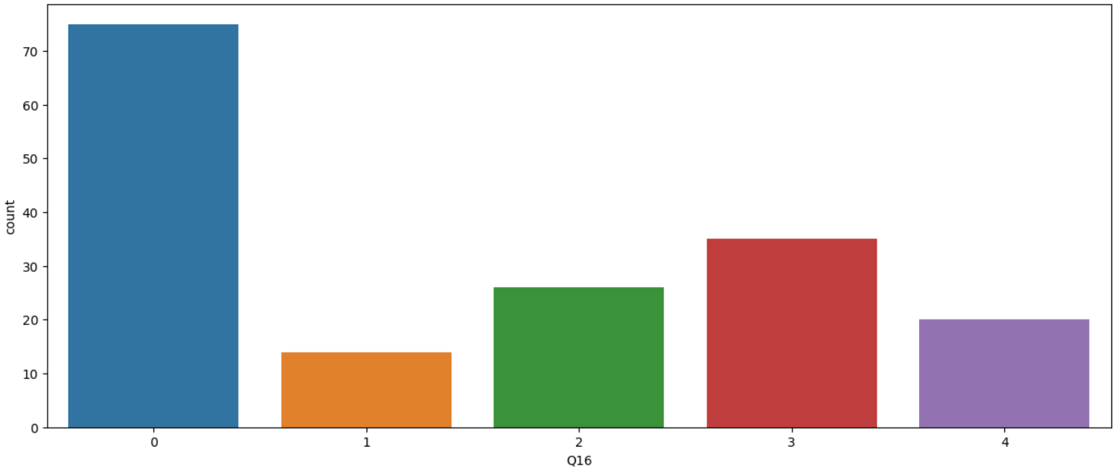

count
0
10
20
30
40
50
60
70
0
1
2
3
4
Q11

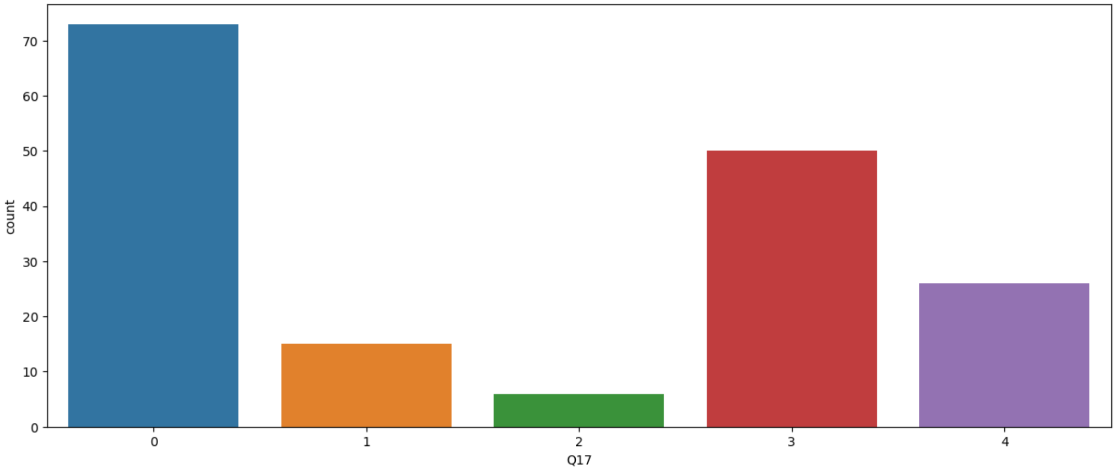

60
50
40
30
20
10
0
count
0
1
2
3
4
Q12

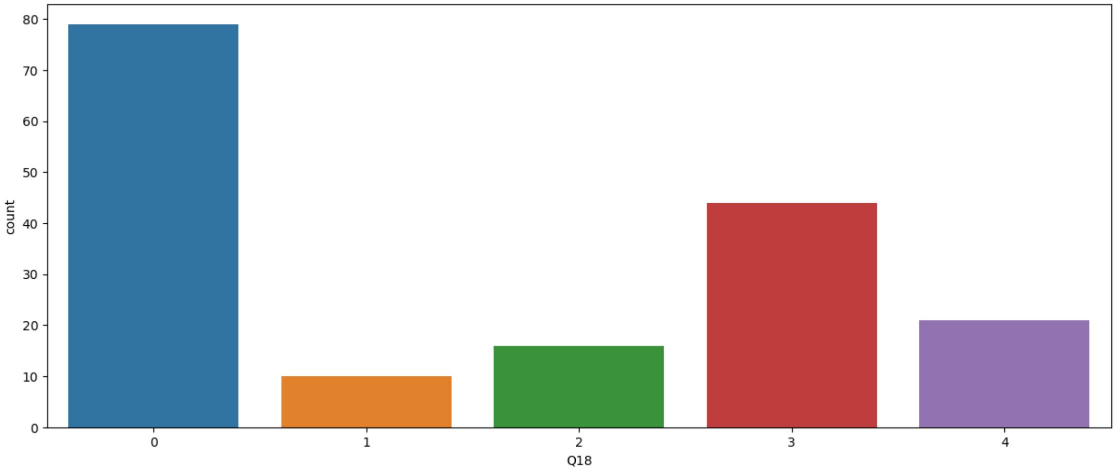

40
30
20
10
0
count
0
1
2
3
4
Q13

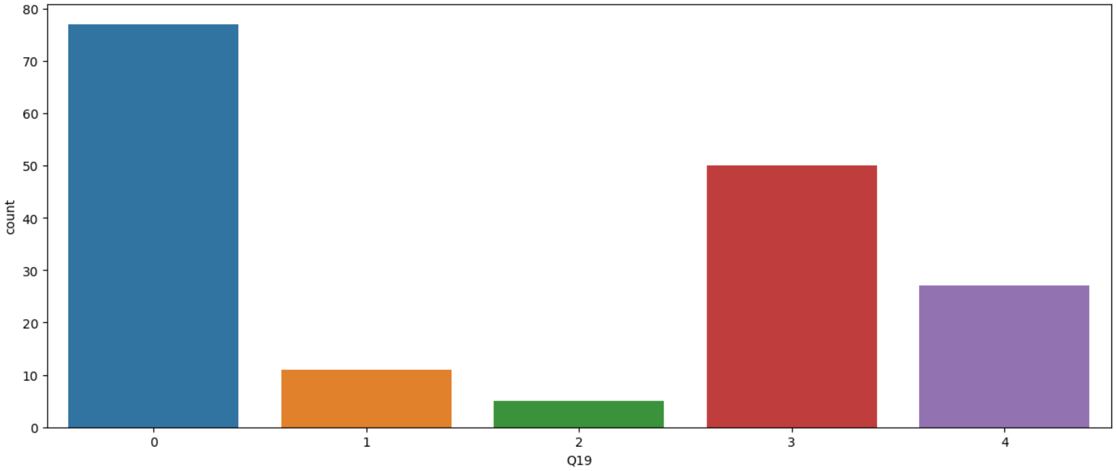

60
50
40
30
20
10
0
count
0
1
2
3
4
Q14

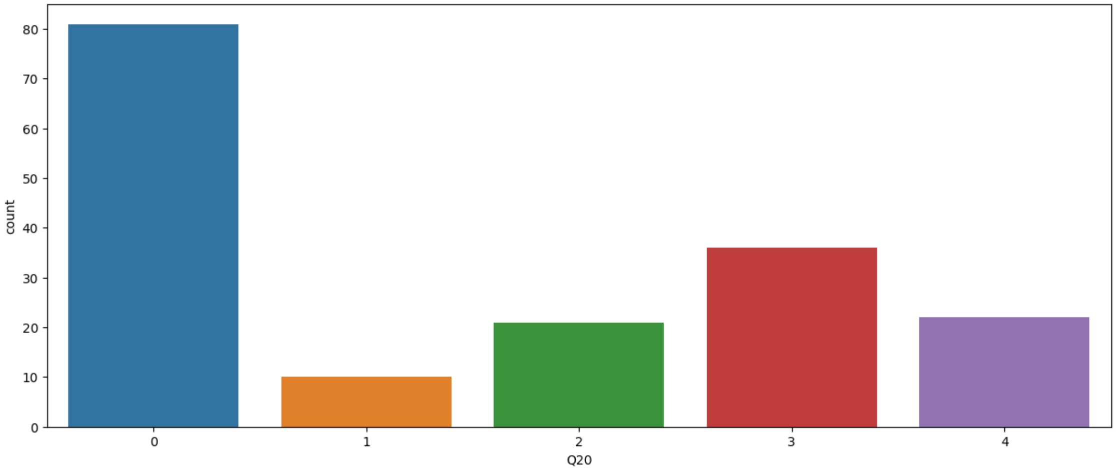

count
0
10
20
30
40
50
60
70
0
1
2
3
4
Q15

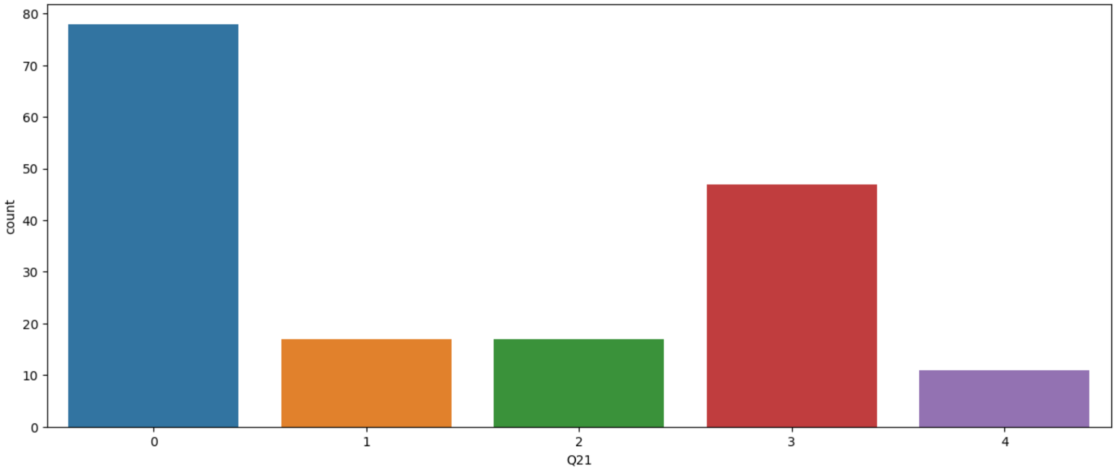

count
0
10
20
30
40
50
60
70
0
1
2
3
4
Q16

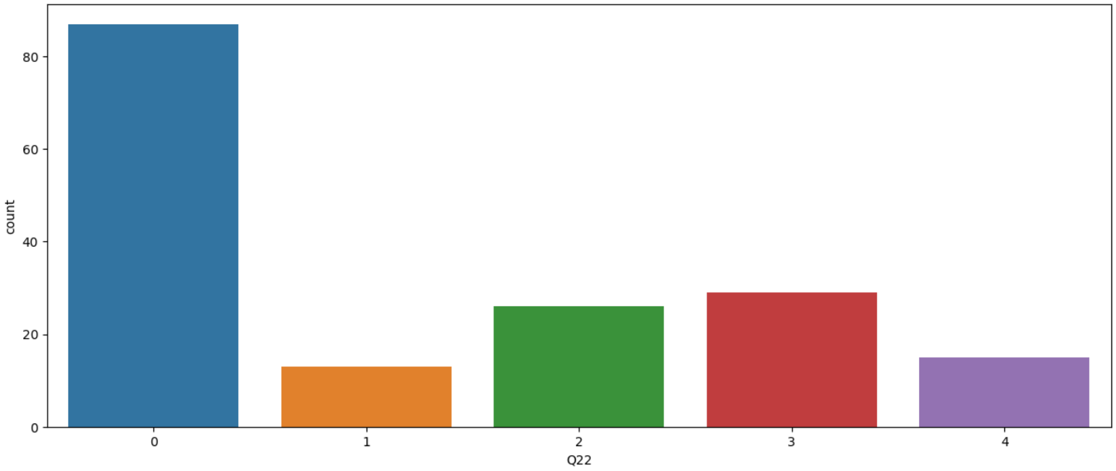

count
0
10
20
30
40
50
60
70
0
1
2
3
4
Q17

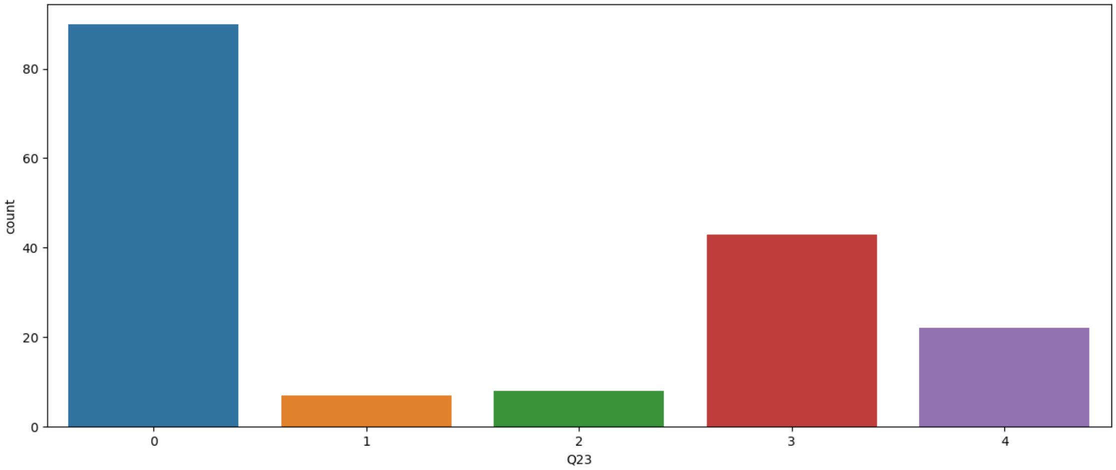

count
0
10
20
30
40
50
60
70
80
0
1
2
3
4
Q18

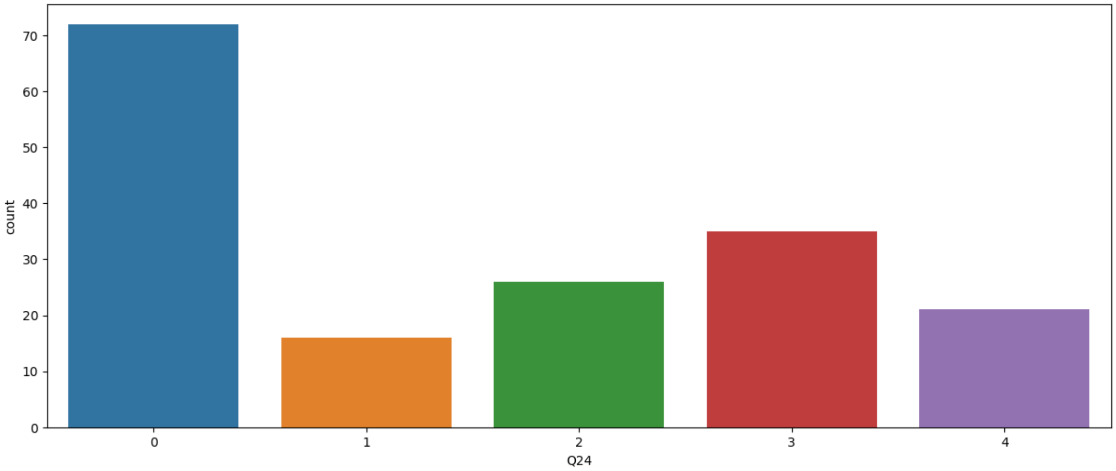

count
0
10
20
30
40
50
60
70
80
0
1
2
3
4
Q19

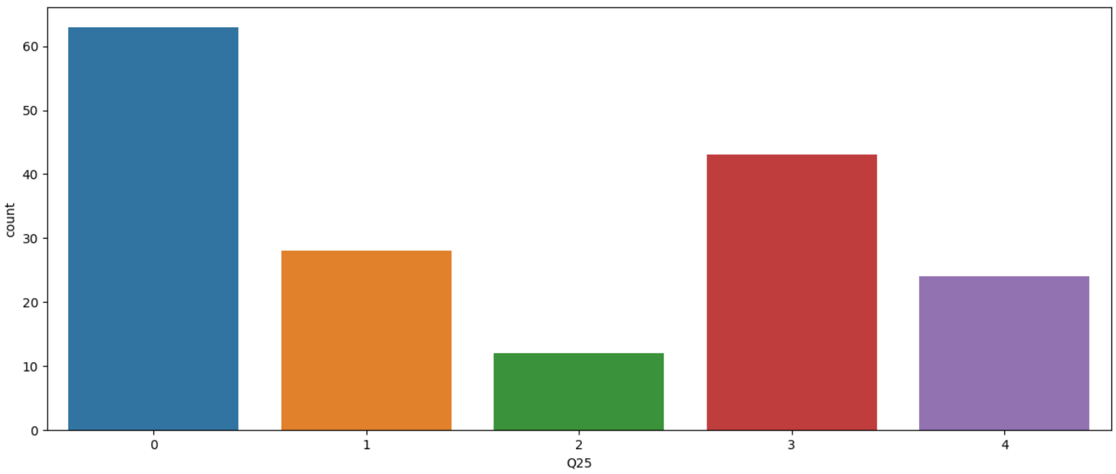

80
70
60
50
40
30
20
10
0
count
0
1
2
3
4
Q20


80
70
60
50
40
30
20
10
0
count
0
1
2
3
4
Q21

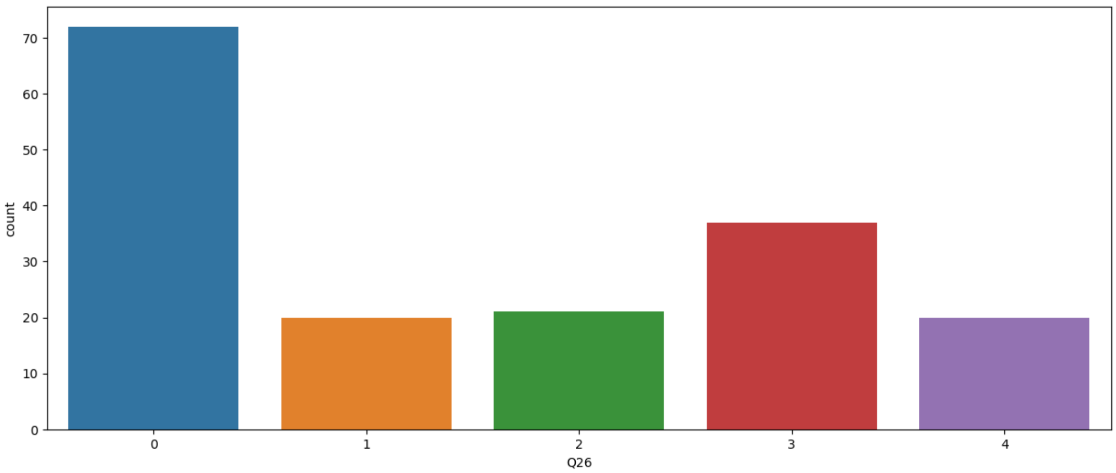

80
60
40
20
0
count
0
1
2
3
4
Q22


count
80
60
40
20
0
0
1
2
3
4
Q23

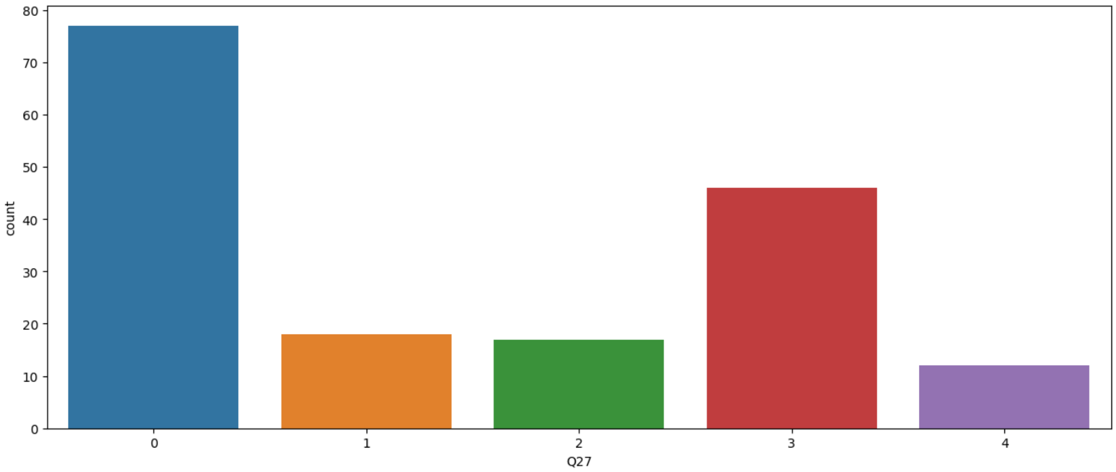

count
70
60
50
40
30
20
10
0
0
1
2
3
4
Q24



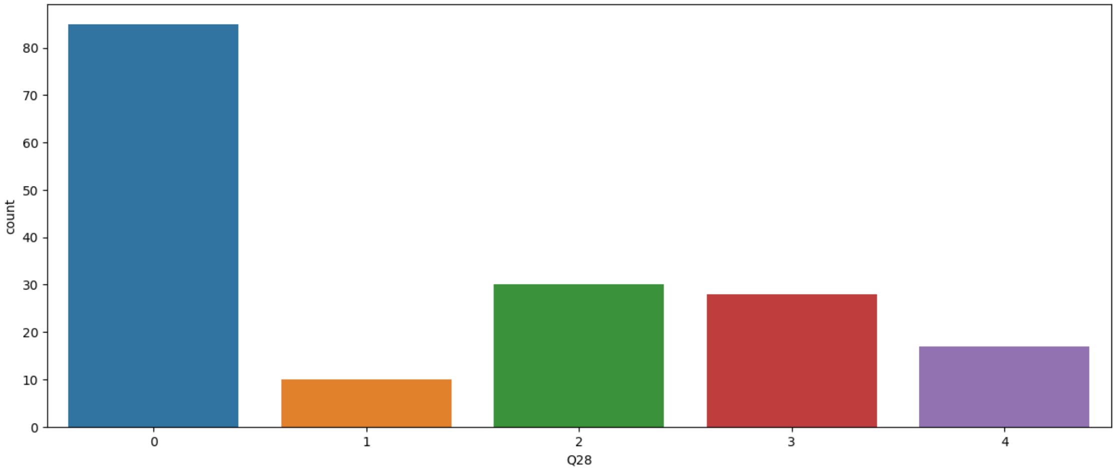
count
60
50
40
30
20
10
0
0
1
2
3
4
Q25

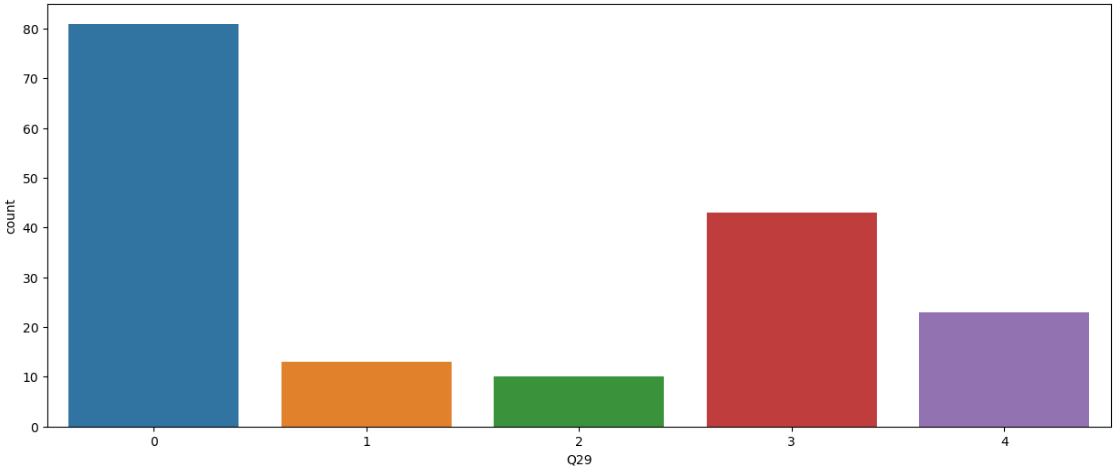


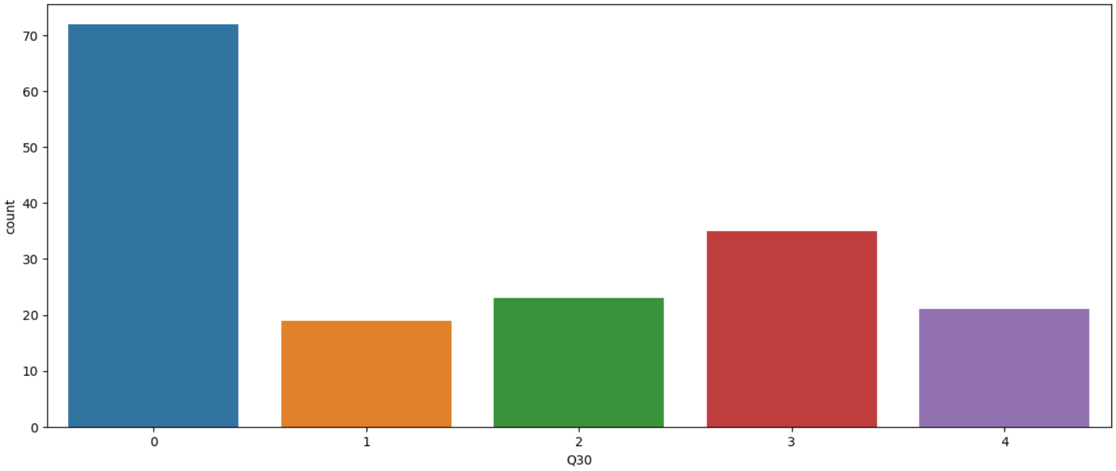
count
70
60
50
40
30
20
10
0
0
1
2
3
4
Q26

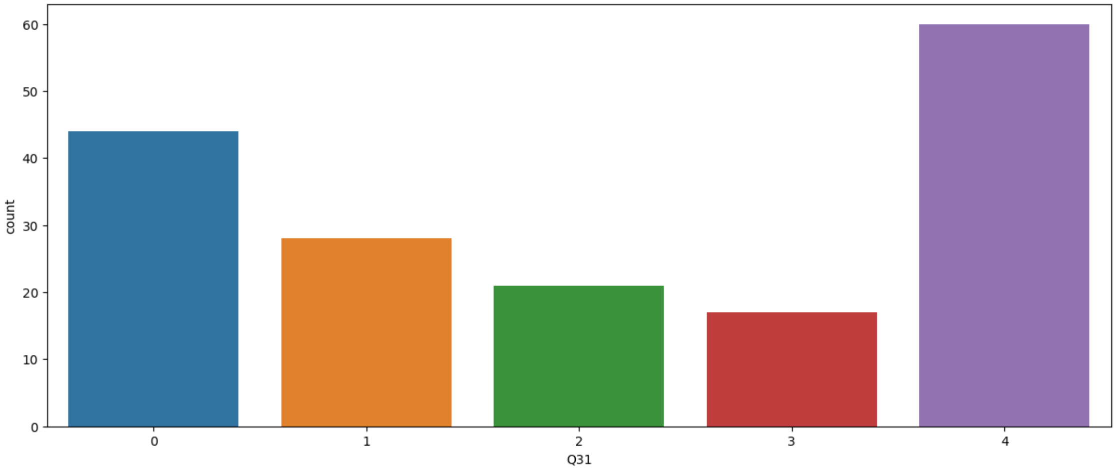


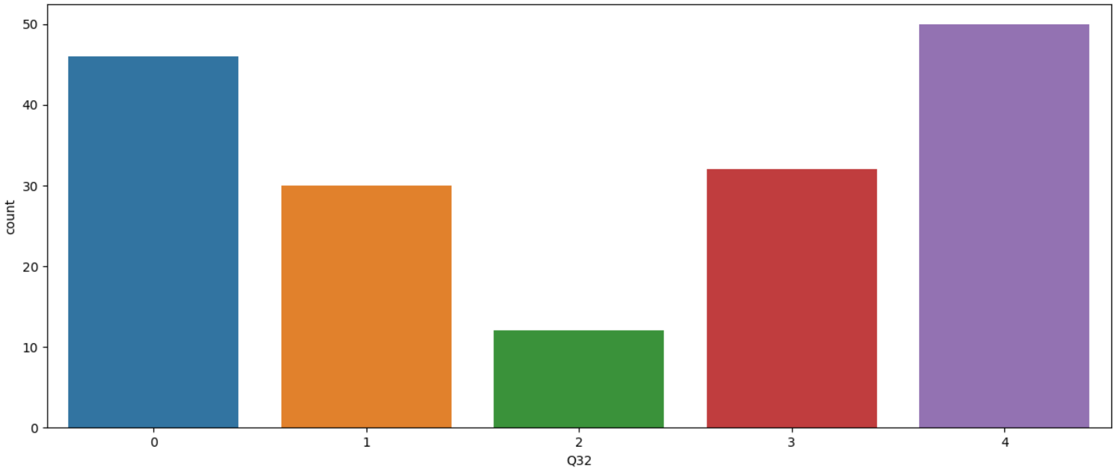
count
80
70
60
50
40
30
20
10
0
0
1
2
3
4
Q27

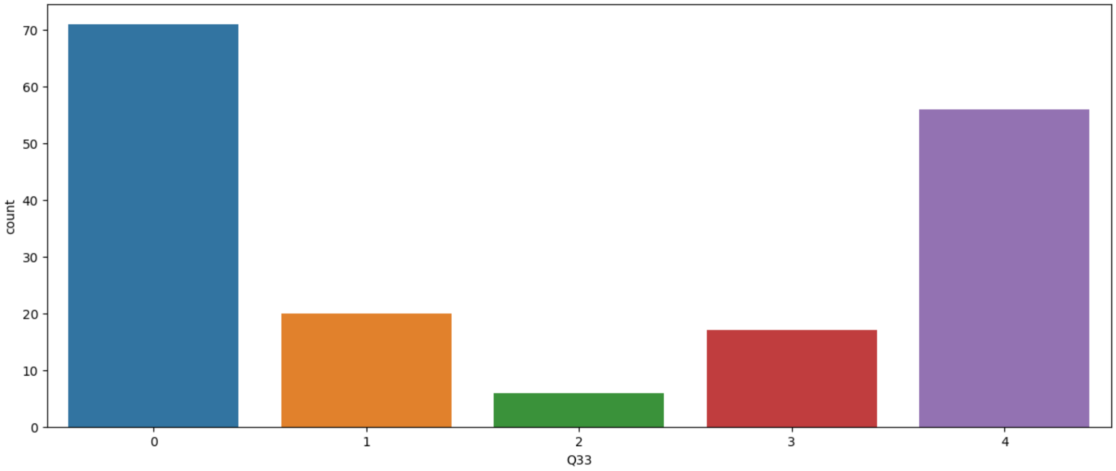

count
0
10
20
30
40
50
60
70
80
0
1
2
3
4
Q28

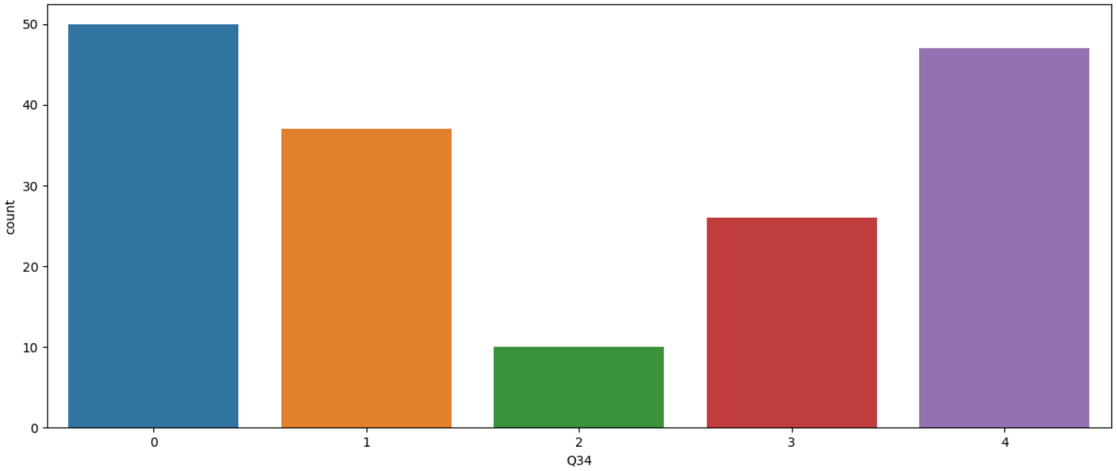

count
0
10
20
30
40
50
60
70
80
0
1
2
3
4
Q29

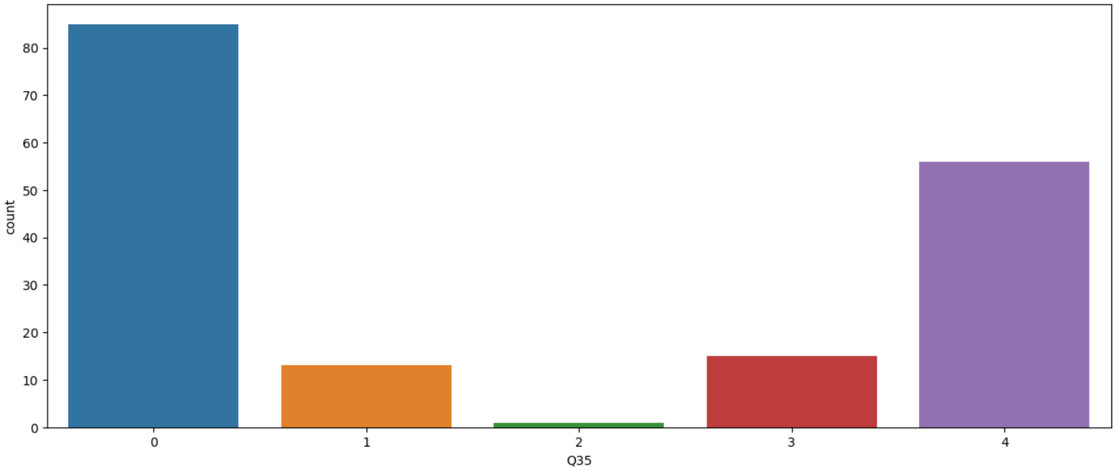

count
0
10
20
30
40
50
60
70
0
1
2
3
4
Q30

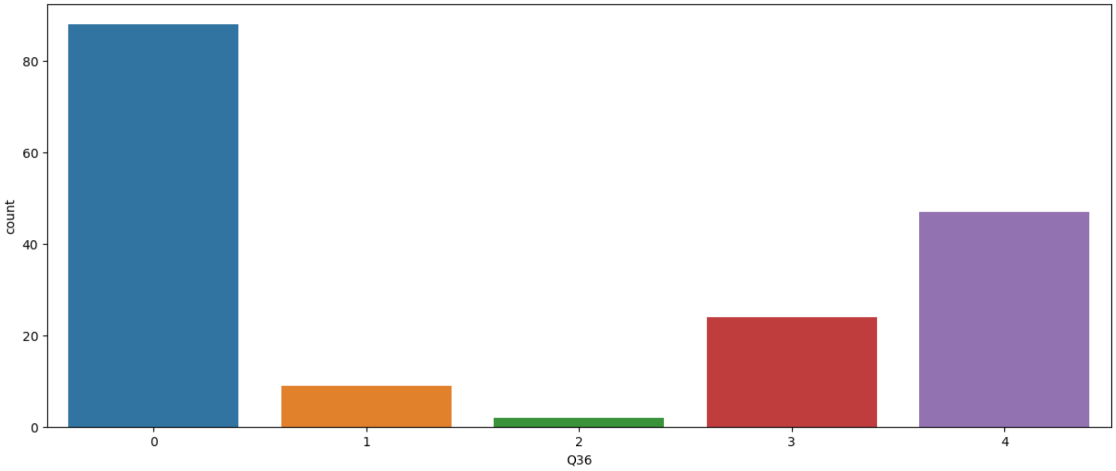

count
0
10
20
30
40
50
60
0
1
2
3
4
Q31

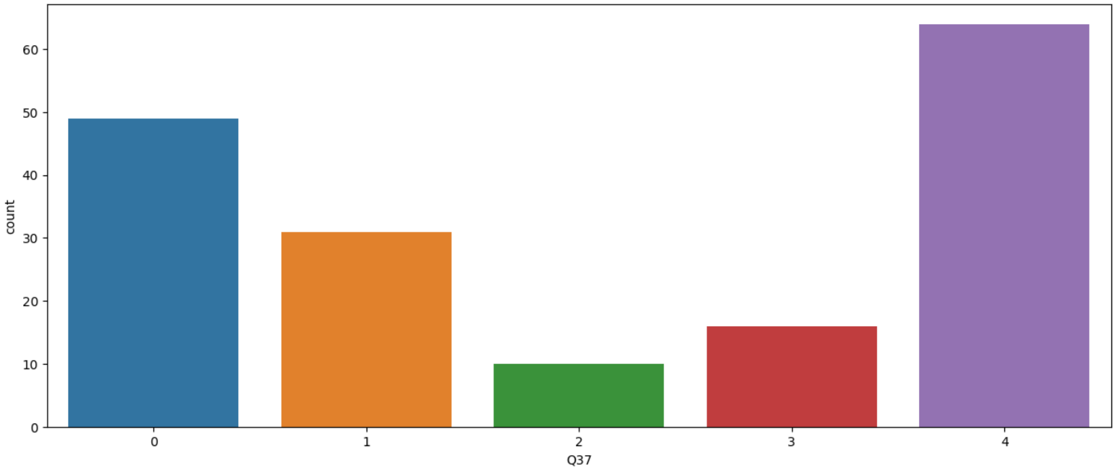

count
0
10
20
30
40
50
0
1
2
3
4
Q32

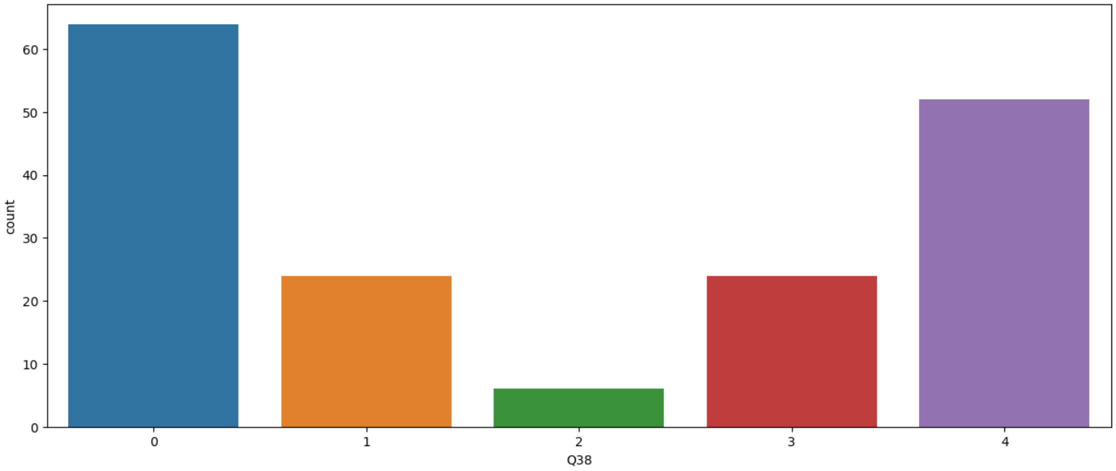

count
0
10
20
30
40
50
60
70
0
1
2
3
4
Q33

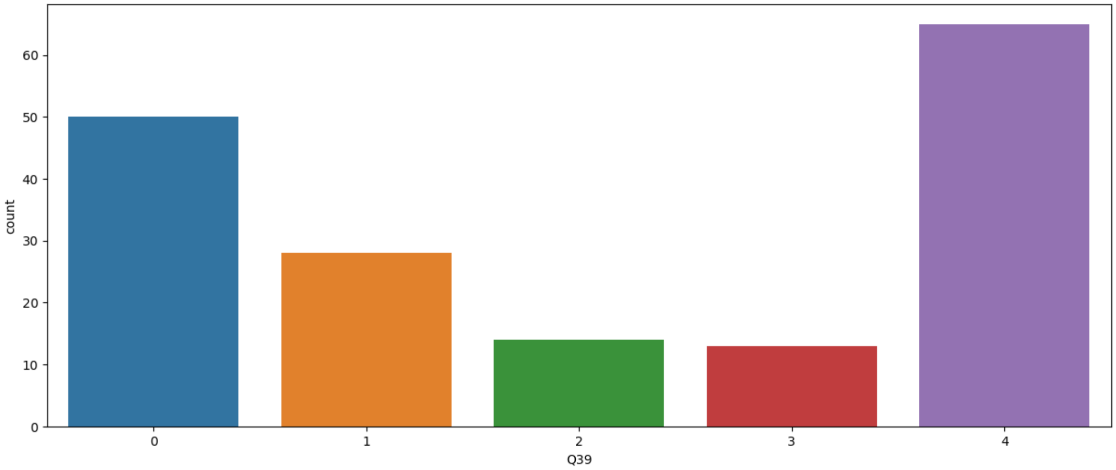

count
0
10
20
30
40
50
0
1
2
3
4
Q34

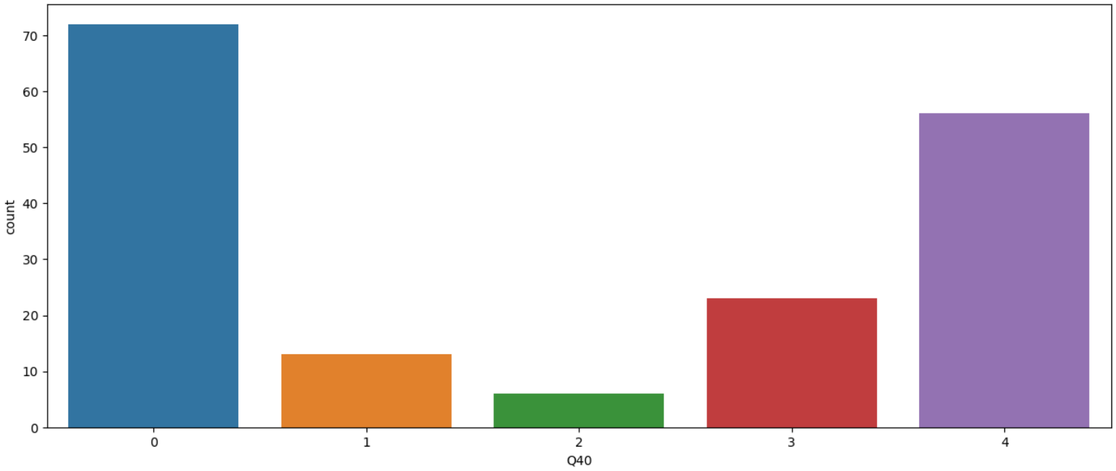

count
0
10
20
30
40
50
60
70
80
0
1
2
3
4
Q35

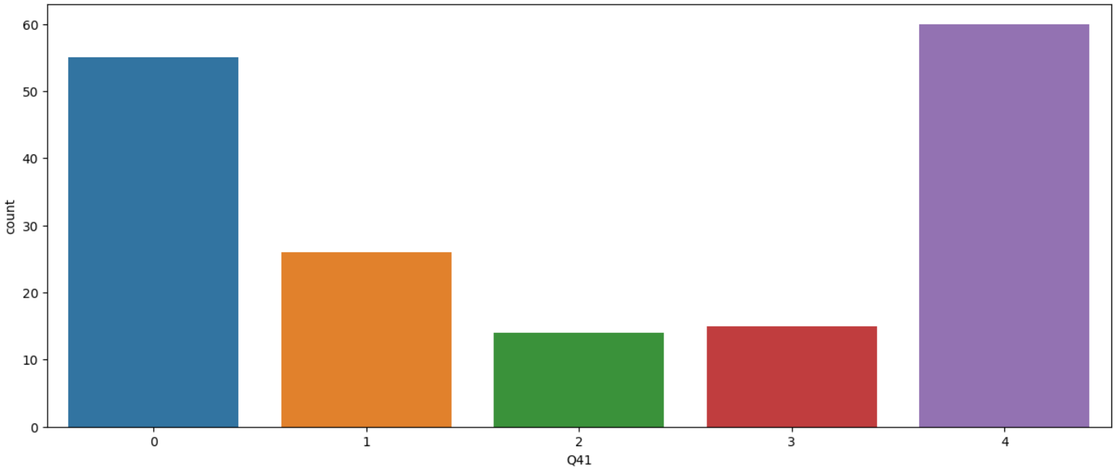

count
0
20
40
60
80
0
1
2
3
4
Q36

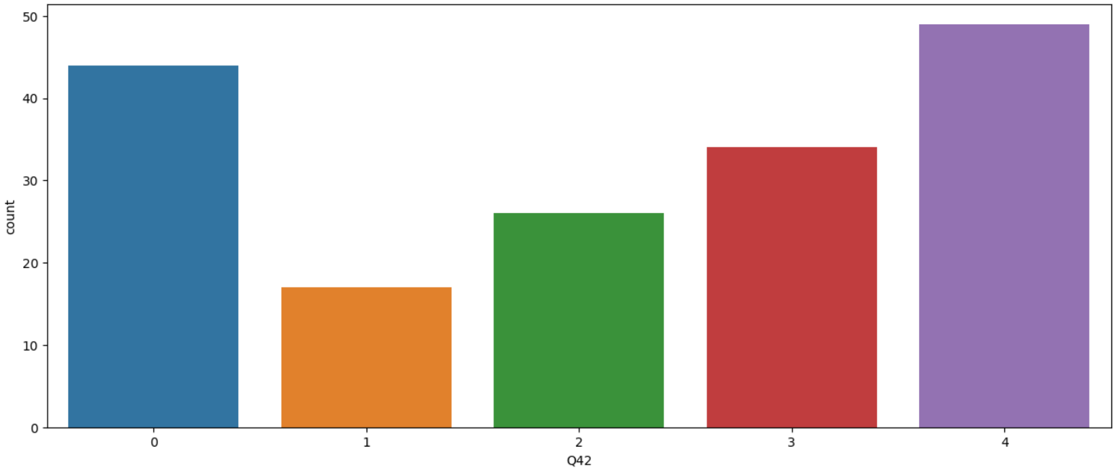

count
0
10
20
30
40
50
60
0
1
2
3
4
Q37

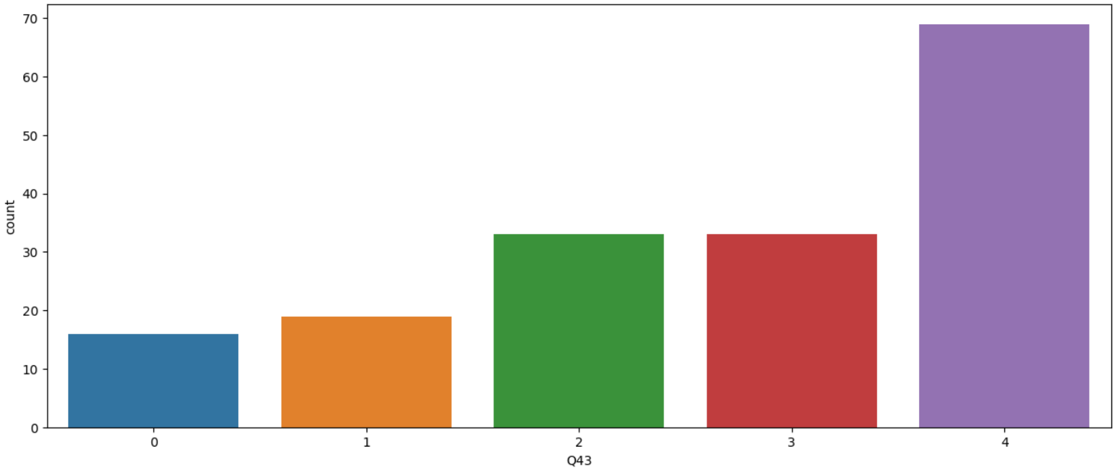

count
0
10
20
30
40
50
60
0
1
2
3
4
Q38

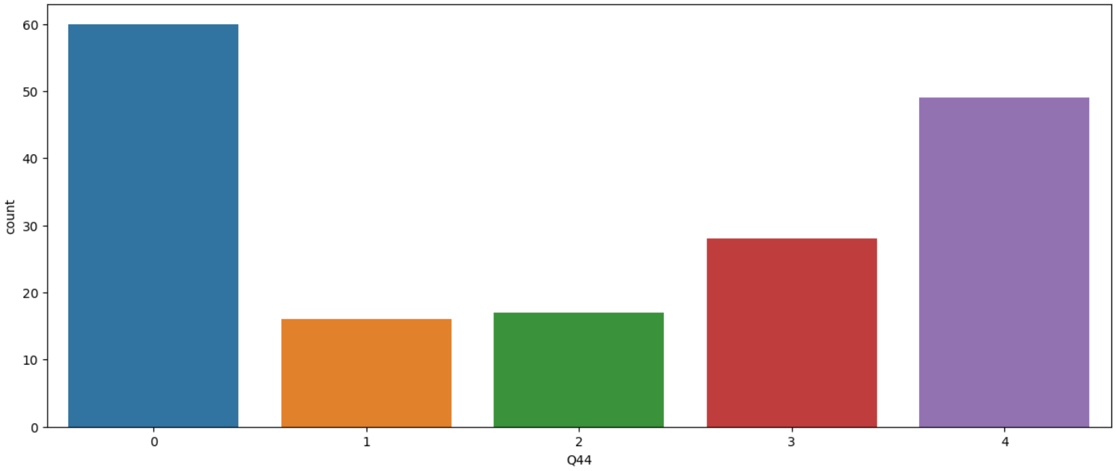

count
0
10
20
30
40
50
60
0
1
2
3
4
Q39


count
0
10
20
30
40
50
60
70
0
1
2
3
4
Q40

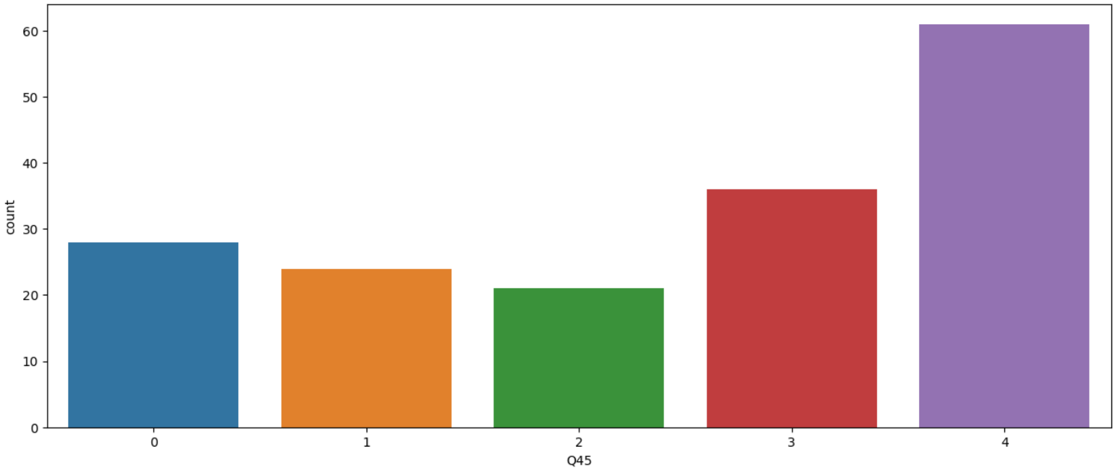

count
0
10
20
30
40
50
60
0
1
2
3
4
Q41


count
0
10
20
30
40
50
0
1
2
3
4
Q42

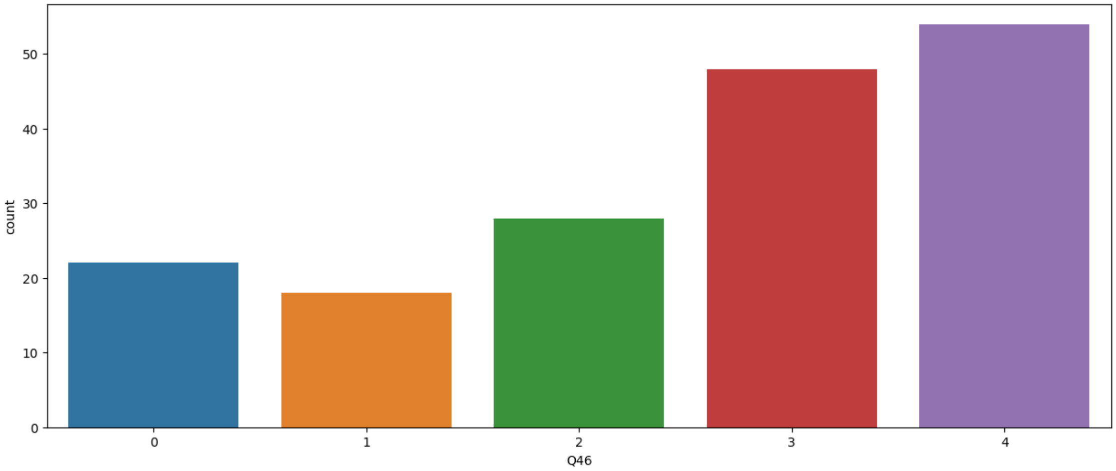

count
0
10
20
30
40
50
60
70
0
1
2
3
4
Q43



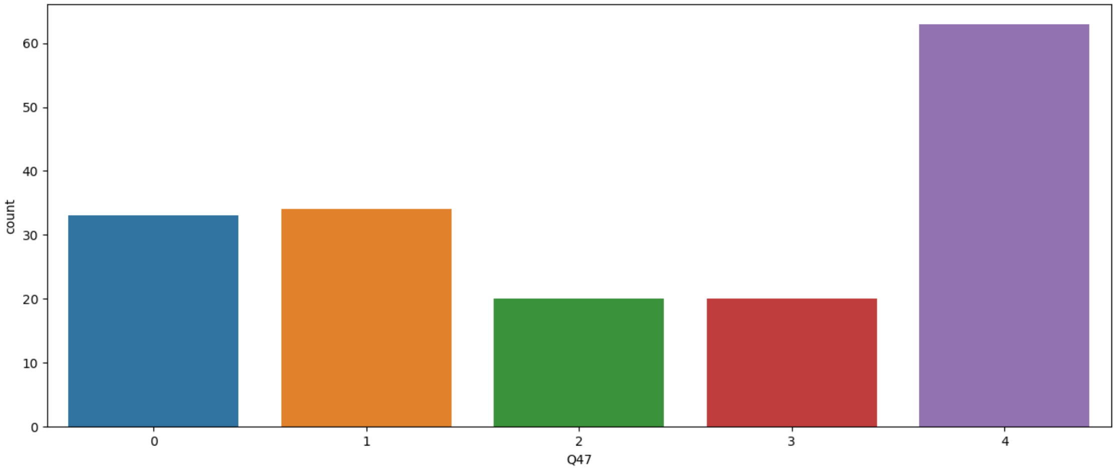
count
0
10
20
30
40
50
60
0
1
2
3
4
Q44

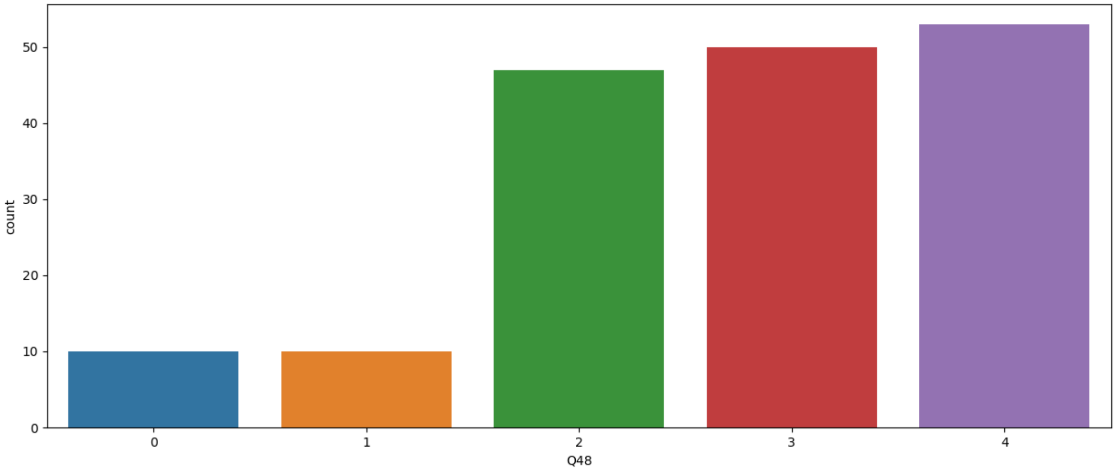


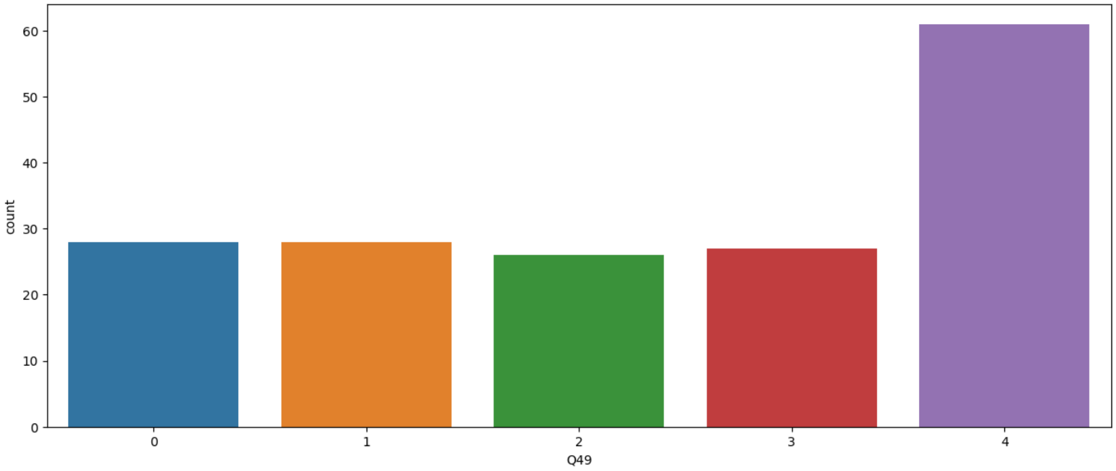
count
0
10
20
30
40
50
60
0
1
2
3
4
Q45

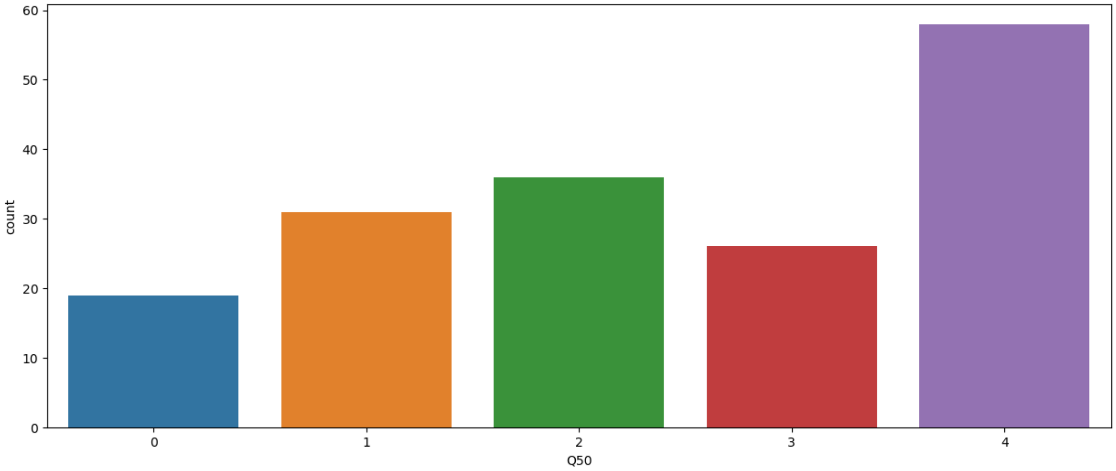


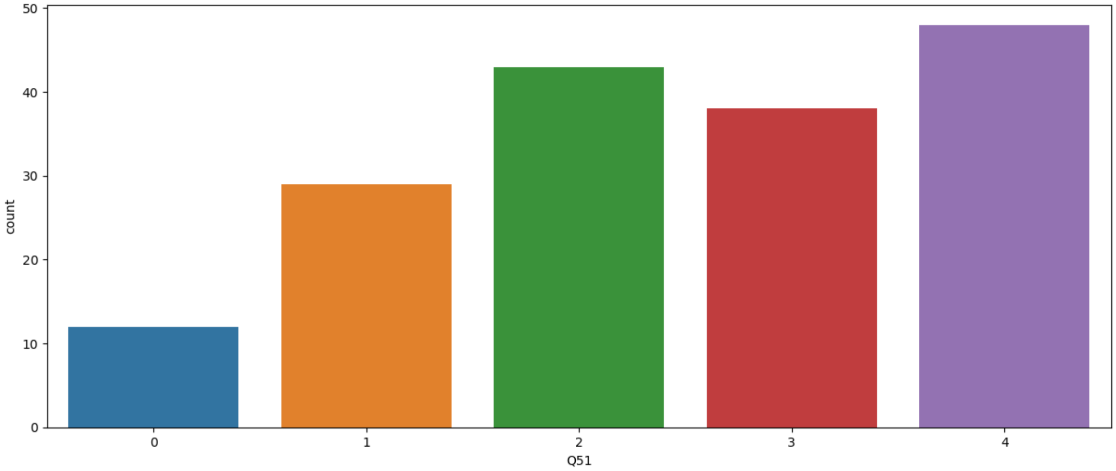
count
0
10
20
30
40
50
0
1
2
3
4
Q46

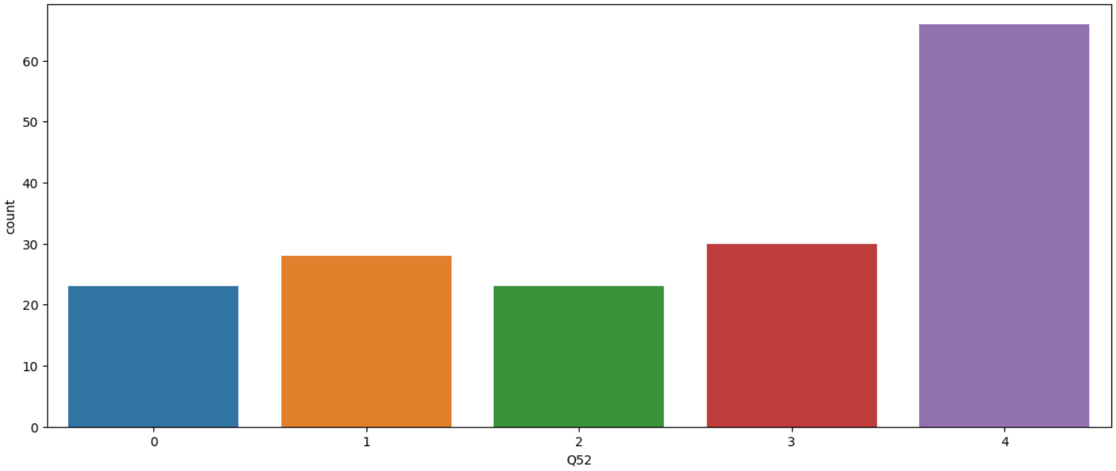

count
0
10
20
30
40
50
60
0
1
2
3
4
Q47

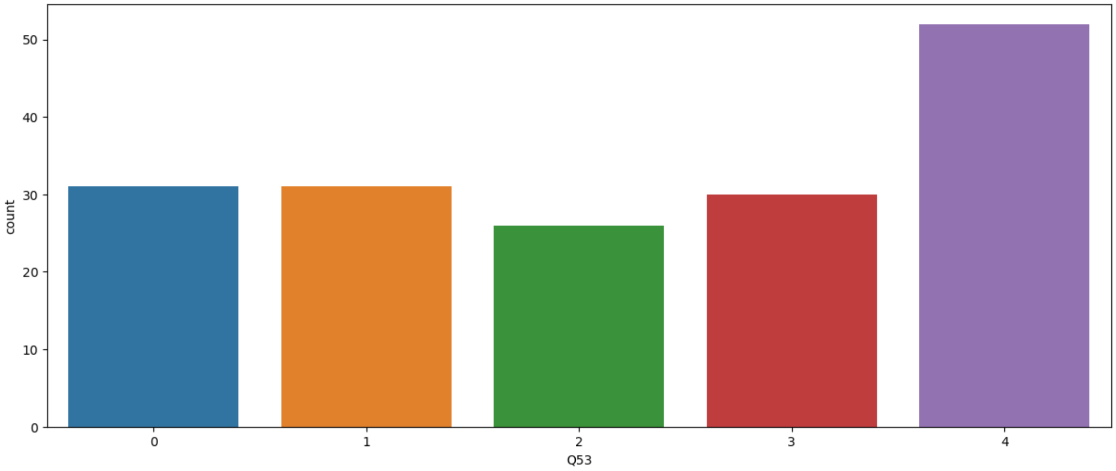

count
0
10
20
30
40
50
0
1
2
3
4
Q48

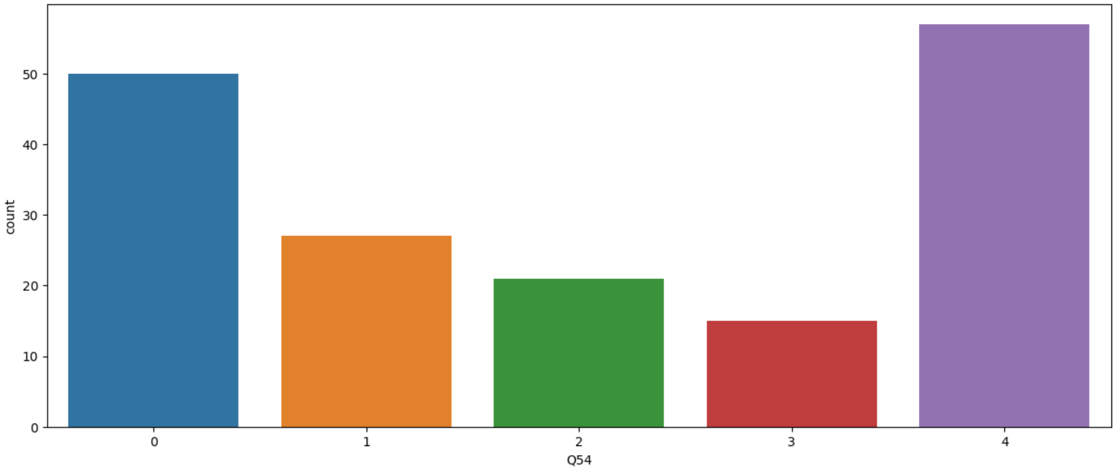

count
60
50
40
30
20
10
0
0
1
2
3
4
Q49

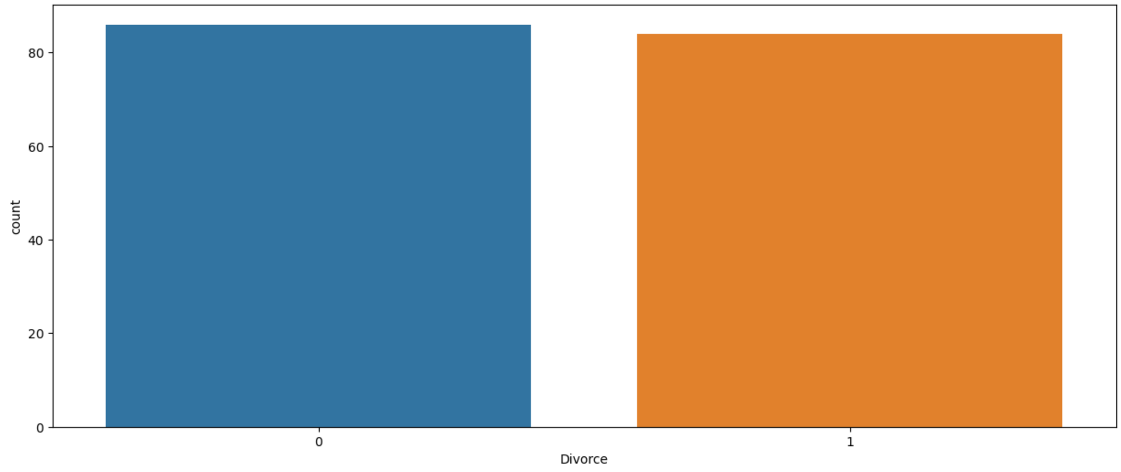

count
60
50
40
30
20
10
0
0
1
2
3
4
Q50


count
50
40
30
20
10
0
0
1
2
3
4
Q51


count
0
10
20
30
40
50
60
0
1
2
3
4
Q52


count
0
10
20
30
40
50
0
1
2
3
4
Q53


count
0
10
20
30
40
50
0
1
2
3
4
Q54



```
[14]: divo['Divorce'].values
```

```
[14]: array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
             1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
             1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
             1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], dtype=int64)
```

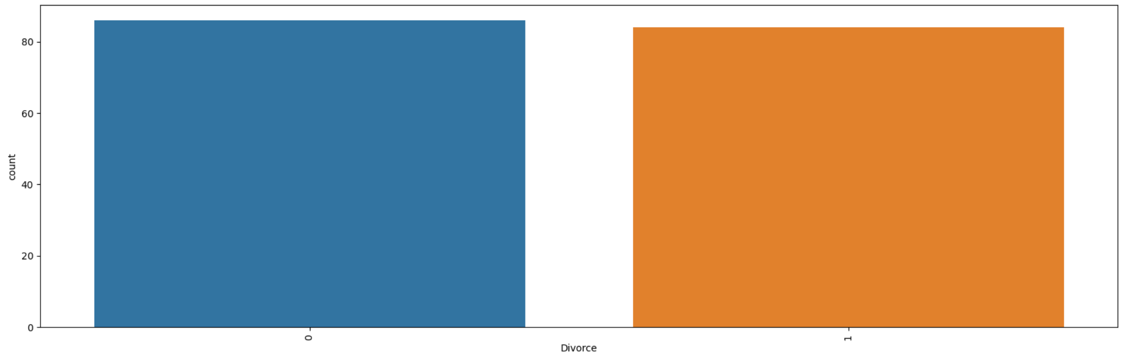
```
[15]: divo['Divorce'].unique()
```

```
[15]: array([1, 0], dtype=int64)
```

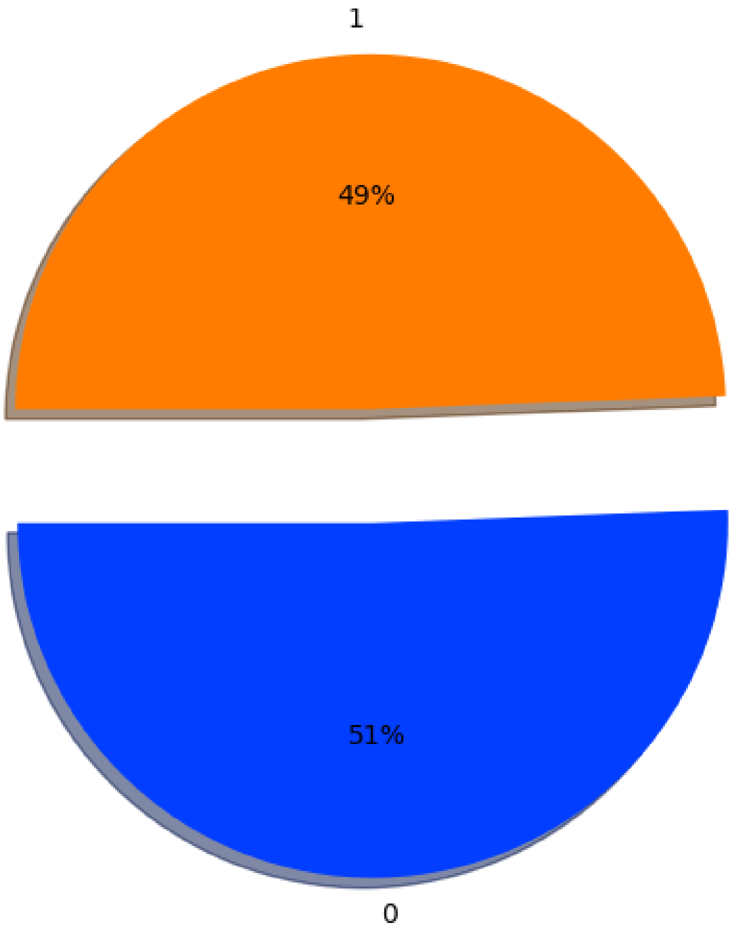
```
[16]: divo['Divorce'].value_counts()
```

```
[16]: 0    86
      1    84
      Name: Divorce, dtype: int64
```

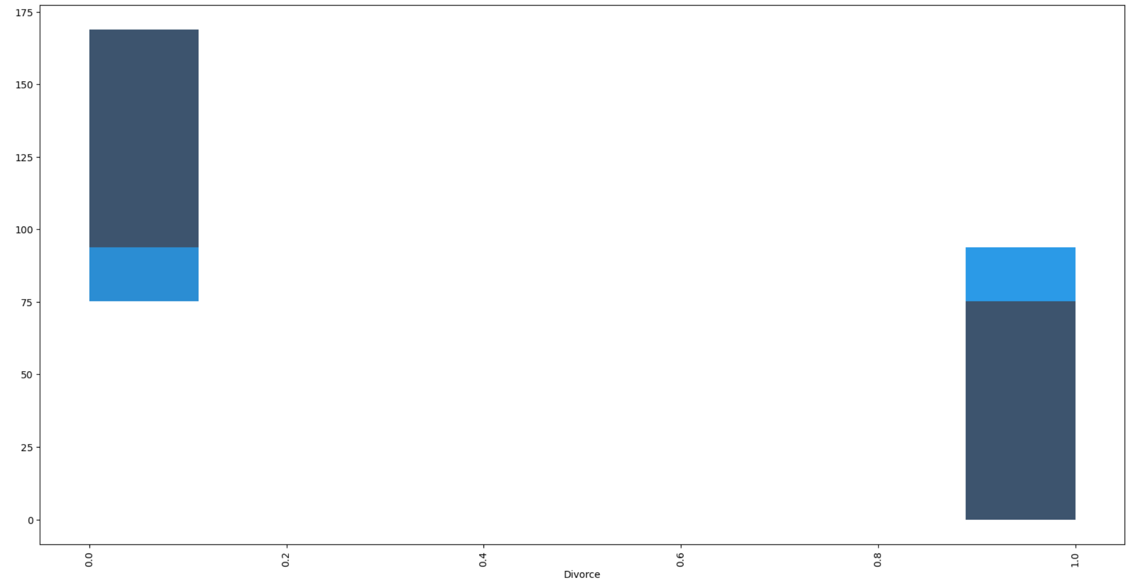
```
[17]: plt.figure(figsize=(20,6))
      sns.countplot('Divorce', data=divo)
      plt.xticks(rotation = 90)
      plt.show()
```



```
[18]: plt.figure(figsize=(20,6))
      colors = sns.color_palette('bright')
      explode = [0.3, 0.02]
      plt.pie(divo['Divorce'].value_counts(), colors = colors,
       labels = [0, 1], autopct = '%0.0f%%', shadow = 'True',
       explode = explode , startangle = 180)
      plt.show()
```



[19]:
```
plt.figure(figsize=(20,10))
sns.histplot(x=divo['Divorce'],y=divo.index)
plt.xticks(rotation=90)
plt.show()
```



```
[21]: divo.corr()
```

```
[21]:            Q1        Q2        Q3        Q4        Q5        Q6        Q7  \
      Q1   1.000000  0.819066  0.832508  0.825066  0.881272  0.287140  0.427989
      Q2   0.819066  1.000000  0.805876  0.791313  0.819360  0.102843  0.417616
      Q3   0.832508  0.805876  1.000000  0.806709  0.800774  0.263032  0.464071
      Q4   0.825066  0.791313  0.806709  1.000000  0.818472  0.185963  0.474806
      Q5   0.881272  0.819360  0.800774  0.818472  1.000000  0.297834  0.381378
      Q6   0.287140  0.102843  0.263032  0.185963  0.297834  1.000000  0.424212
      Q7   0.427989  0.417616  0.464071  0.474806  0.381378  0.424212  1.000000
      Q8   0.802357  0.864284  0.757264  0.798347  0.877584  0.184019  0.412807
      Q9   0.845916  0.827711  0.816653  0.829053  0.916327  0.301342  0.517522
      Q10  0.790183  0.782286  0.753017  0.873636  0.823659  0.266076  0.498266
      Q11  0.892253  0.823380  0.805915  0.808533  0.936955  0.340135  0.432479
      Q12  0.794307  0.862835  0.780258  0.793992  0.846513  0.209801  0.511761
      Q13  0.842996  0.791073  0.758969  0.751623  0.915033  0.305109  0.373361
      Q14  0.817099  0.875800  0.750602  0.757000  0.845576  0.224459  0.491021
      Q15  0.848754  0.801316  0.806909  0.794184  0.879461  0.323787  0.494110
      Q16  0.831822  0.806497  0.775528  0.878416  0.853561  0.311056  0.573290
      Q17  0.895970  0.822317  0.808161  0.809968  0.947429  0.377330  0.461450
      Q18  0.853739  0.883856  0.797395  0.835296  0.894474  0.251856  0.544550
      Q19  0.900446  0.829422  0.798999  0.832750  0.943349  0.365227  0.469995
      Q20  0.840966  0.884176  0.807892  0.815896  0.892909  0.230486  0.544207
      Q21  0.815708  0.790468  0.796069  0.775132  0.871994  0.273564  0.409827
      Q22  0.785280  0.795406  0.727933  0.839534  0.840265  0.220010  0.378915
      Q23  0.822534  0.773018  0.706585  0.744783  0.888584  0.246478  0.254912
      Q24  0.813233  0.868240  0.740476  0.776640  0.833608  0.191458  0.446469
```

```
Q25      0.822084  0.769244  0.724506  0.736228  0.888740  0.291159  0.288867
Q26      0.803507  0.861421  0.728653  0.762765  0.836194  0.200634  0.443149
Q27      0.829037  0.817364  0.797595  0.767206  0.883768  0.283895  0.444643
Q28      0.762102  0.776943  0.689914  0.827847  0.809789  0.254858  0.351262
Q29      0.858139  0.789827  0.755491  0.781792  0.925601  0.309302  0.349379
Q30      0.792257  0.844007  0.752391  0.772562  0.837501  0.266464  0.448569
Q31      0.699223  0.661210  0.652188  0.661251  0.785038  0.247634  0.334308
Q32      0.739679  0.735763  0.747669  0.746677  0.832032  0.316605  0.442306
Q33      0.799735  0.757286  0.726481  0.764381  0.879037  0.292037  0.395764
Q34      0.749774  0.714360  0.702500  0.729022  0.827560  0.279789  0.328700
Q35      0.796413  0.753566  0.730290  0.770813  0.878289  0.276539  0.349076
Q36      0.812867  0.781295  0.744390  0.794636  0.887498  0.287708  0.370158
Q37      0.786890  0.747088  0.736984  0.760451  0.859581  0.281458  0.431979
Q38      0.804129  0.751705  0.740642  0.790350  0.852601  0.297791  0.401769
Q39      0.817035  0.787768  0.759820  0.763502  0.866293  0.296121  0.477063
Q40      0.838355  0.788200  0.781657  0.798520  0.871809  0.351433  0.501758
Q41      0.804182  0.780757  0.739967  0.768706  0.864434  0.329765  0.445483
Q42      0.642307  0.648539  0.569293  0.639671  0.737922  0.227993  0.333211
Q43      0.482223  0.503894  0.385152  0.452479  0.613142  0.171599  0.149930
Q44      0.752972  0.699765  0.661830  0.707212  0.799453  0.339918  0.425874
Q45      0.510160  0.489062  0.427409  0.446798  0.591656  0.094820  0.199548
Q46      0.400296  0.389519  0.308149  0.340240  0.470758  0.127759  0.069850
Q47      0.582693  0.616884  0.544863  0.552301  0.719899  0.212979  0.254225
Q48      0.633564  0.643762  0.638256  0.630205  0.659220  0.200673  0.311110
Q49      0.674843  0.659841  0.647961  0.699069  0.762257  0.201091  0.291325
Q50      0.725443  0.680538  0.663995  0.685263  0.795960  0.221100  0.332370
Q51      0.684143  0.636558  0.600603  0.624015  0.742664  0.179119  0.349920
Q52      0.575463  0.536294  0.491803  0.534264  0.663855  0.205056  0.243104
Q53      0.611422  0.610726  0.598749  0.588390  0.719493  0.258092  0.313725
Q54      0.768522  0.728897  0.673012  0.698264  0.836799  0.292428  0.347493
Divorce  0.861324  0.820774  0.806709  0.819583  0.893180  0.420913  0.544835

               Q8        Q9       Q10  …       Q46       Q47       Q48  \
Q1       0.802357  0.845916  0.790183  …  0.400296  0.582693  0.633564
Q2       0.864284  0.827711  0.782286  …  0.389519  0.616884  0.643762
Q3       0.757264  0.816653  0.753017  …  0.308149  0.544863  0.638256
Q4       0.798347  0.829053  0.873636  …  0.340240  0.552301  0.630205
Q5       0.877584  0.916327  0.823659  …  0.470758  0.719899  0.659220
Q6       0.184019  0.301342  0.266076  …  0.127759  0.212979  0.200673
Q7       0.412807  0.517522  0.498266  …  0.069850  0.254225  0.311110
Q8       1.000000  0.915301  0.828031  …  0.433541  0.675584  0.588531
Q9       0.915301  1.000000  0.852385  …  0.434318  0.693839  0.611726
Q10      0.828031  0.852385  1.000000  …  0.342315  0.559998  0.550313
Q11      0.889795  0.911557  0.855596  …  0.411962  0.645995  0.616833
Q12      0.890338  0.869088  0.847474  …  0.374833  0.632118  0.587084
Q13      0.840350  0.873048  0.819715  …  0.441968  0.654707  0.607921
Q14      0.888822  0.868122  0.797964  …  0.417249  0.622207  0.613143
```

```
Q15      0.873804  0.949041  0.853898  …  0.419023  0.652757  0.570107
Q16      0.865680  0.893377  0.922320  …  0.404929  0.619932  0.560172
Q17      0.881005  0.922307  0.843101  …  0.463868  0.692801  0.611383
Q18      0.941084  0.925543  0.867250  …  0.422372  0.645743  0.577324
Q19      0.873546  0.916472  0.837476  …  0.459280  0.687382  0.634737
Q20      0.922465  0.902254  0.846435  …  0.427054  0.666745  0.623265
Q21      0.861939  0.909428  0.793875  …  0.506472  0.714223  0.609352
Q22      0.857010  0.849978  0.862670  …  0.503282  0.689916  0.592974
Q23      0.845731  0.850241  0.750494  …  0.597487  0.768102  0.623255
Q24      0.896841  0.851000  0.796453  …  0.484264  0.692690  0.593157
Q25      0.809110  0.838754  0.792520  …  0.535168  0.670306  0.593997
Q26      0.883414  0.850286  0.802167  …  0.466021  0.659920  0.584190
Q27      0.848766  0.903959  0.796466  …  0.522207  0.702745  0.616280
Q28      0.822361  0.818032  0.848007  …  0.485621  0.645103  0.579382
Q29      0.860194  0.878845  0.790977  …  0.537866  0.738302  0.629746
Q30      0.902820  0.854455  0.801220  …  0.428798  0.635199  0.597375
Q31      0.716731  0.745699  0.704880  …  0.443638  0.634671  0.601569
Q32      0.762425  0.803397  0.759551  …  0.381200  0.623136  0.690888
Q33      0.818682  0.844909  0.755521  …  0.476596  0.687061  0.642480
Q34      0.780778  0.810135  0.737276  …  0.479888  0.625707  0.649640
Q35      0.827441  0.854943  0.755314  …  0.550157  0.726969  0.628374
Q36      0.845435  0.871648  0.781450  …  0.568555  0.715604  0.675947
Q37      0.800964  0.839067  0.779488  …  0.421546  0.614817  0.639337
Q38      0.815830  0.849469  0.796215  …  0.488014  0.626556  0.635144
Q39      0.797134  0.850638  0.782768  …  0.423210  0.652620  0.641082
Q40      0.822302  0.875661  0.819700  …  0.485531  0.626907  0.655541
Q41      0.821081  0.852477  0.777444  …  0.449828  0.689292  0.624703
Q42      0.699571  0.737409  0.670209  …  0.548272  0.717083  0.618045
Q43      0.555187  0.585679  0.459405  …  0.561868  0.695560  0.409197
Q44      0.760016  0.808607  0.723545  …  0.572467  0.690103  0.594348
Q45      0.542547  0.575326  0.435798  …  0.592041  0.720692  0.413413
Q46      0.433541  0.434318  0.342315  …  1.000000  0.664794  0.448773
Q47      0.675584  0.693839  0.559998  …  0.664794  1.000000  0.602951
Q48      0.588531  0.611726  0.550313  …  0.448773  0.602951  1.000000
Q49      0.674776  0.711503  0.659604  …  0.530939  0.649803  0.594826
Q50      0.729668  0.755509  0.672236  …  0.523832  0.658809  0.640171
Q51      0.690190  0.713750  0.618666  …  0.490177  0.600896  0.697531
Q52      0.658613  0.652376  0.513842  …  0.550210  0.571223  0.513645
Q53      0.705071  0.699211  0.592641  …  0.505164  0.623877  0.637911
Q54      0.807911  0.810977  0.698528  …  0.483424  0.689735  0.588135
Divorce  0.869569  0.912368  0.834897  …  0.443465  0.656409  0.619830

              Q49       Q50       Q51       Q52       Q53       Q54   Divorce
Q1       0.674843  0.725443  0.684143  0.575463  0.611422  0.768522  0.861324
Q2       0.659841  0.680538  0.636558  0.536294  0.610726  0.728897  0.820774
Q3       0.647961  0.663995  0.600603  0.491803  0.598749  0.673012  0.806709
Q4       0.699069  0.685263  0.624015  0.534264  0.588390  0.698264  0.819583
```

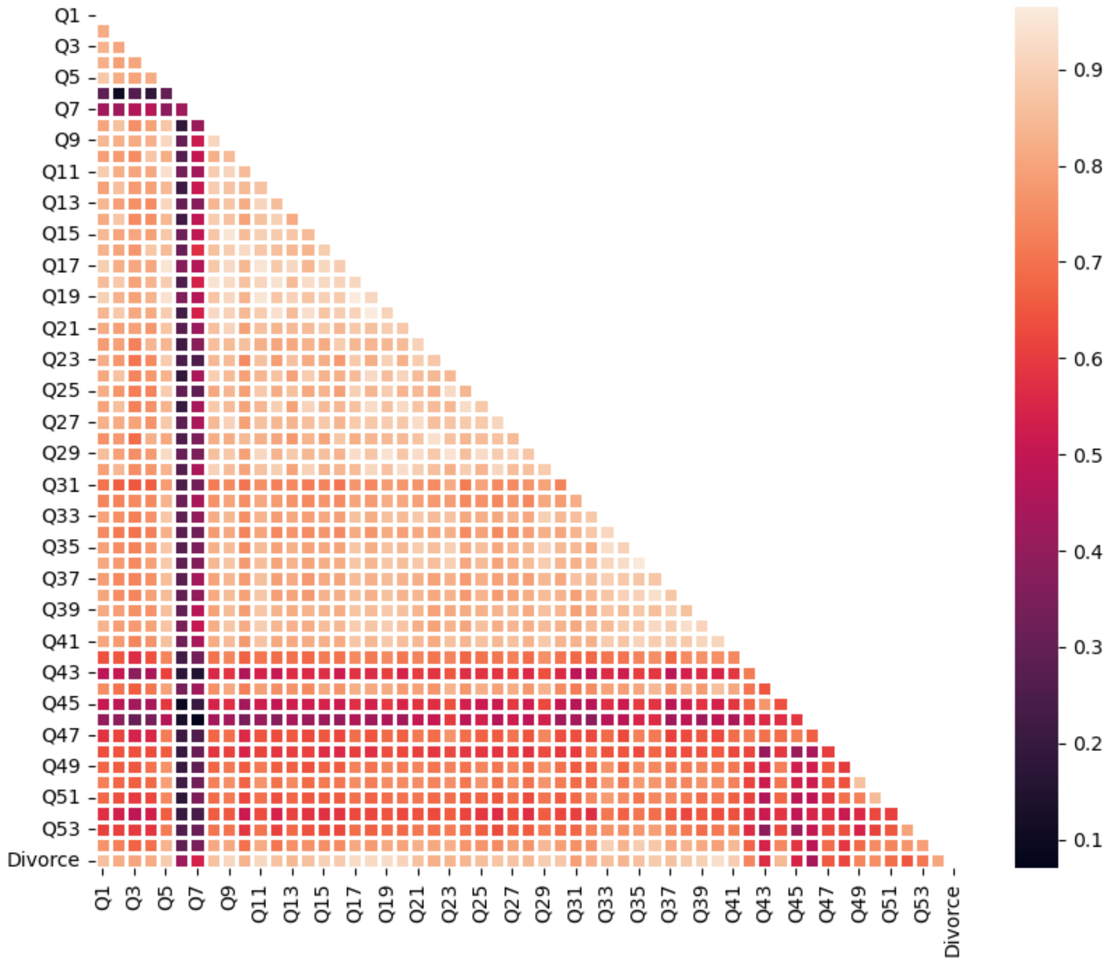
Q5	0.762257	0.795960	0.742664	0.663855	0.719493	0.836799	0.893180
Q6	0.201091	0.221100	0.179119	0.205056	0.258092	0.292428	0.420913
Q7	0.291325	0.332370	0.349920	0.243104	0.313725	0.347493	0.544835
Q8	0.674776	0.729668	0.690190	0.658613	0.705071	0.807911	0.869569
Q9	0.711503	0.755509	0.713750	0.652376	0.699211	0.810977	0.912368
Q10	0.659604	0.672236	0.618666	0.513842	0.592641	0.698528	0.834897
Q11	0.716001	0.766418	0.696282	0.631223	0.703364	0.819972	0.918386
Q12	0.662506	0.700609	0.633361	0.541752	0.613401	0.707153	0.868983
Q13	0.666462	0.746365	0.677543	0.613943	0.682513	0.761607	0.844743
Q14	0.637968	0.687542	0.649200	0.583448	0.642306	0.764686	0.864316
Q15	0.672676	0.713697	0.663087	0.579265	0.632804	0.757981	0.901220
Q16	0.658464	0.711698	0.659871	0.583625	0.623016	0.750224	0.886260
Q17	0.731158	0.754691	0.698285	0.626845	0.694300	0.812494	0.929346
Q18	0.703300	0.726645	0.671822	0.620433	0.666990	0.781632	0.923208
Q19	0.740252	0.752400	0.689729	0.646107	0.691421	0.814552	0.928627
Q20	0.718281	0.746519	0.700391	0.623030	0.688278	0.801546	0.907008
Q21	0.735288	0.753011	0.706653	0.642550	0.711991	0.831327	0.864519
Q22	0.706148	0.736448	0.688090	0.626827	0.700597	0.810719	0.825938
Q23	0.731493	0.754818	0.695089	0.660679	0.721908	0.828007	0.837504
Q24	0.699272	0.710019	0.660250	0.617920	0.689712	0.771216	0.839392
Q25	0.739600	0.768037	0.672088	0.593523	0.663172	0.769315	0.857052
Q26	0.707706	0.708283	0.649172	0.566313	0.628569	0.747339	0.872868
Q27	0.741492	0.756652	0.697969	0.629278	0.675915	0.791929	0.869788
Q28	0.685723	0.701984	0.627694	0.573582	0.659895	0.767266	0.846606
Q29	0.761812	0.766783	0.704671	0.658174	0.722518	0.846811	0.892954
Q30	0.699638	0.719207	0.658454	0.619378	0.686362	0.792509	0.874531
Q31	0.655936	0.687783	0.687045	0.611062	0.679921	0.759989	0.792607
Q32	0.673165	0.691838	0.741075	0.572254	0.730081	0.766907	0.829056
Q33	0.766631	0.776502	0.785674	0.709548	0.768211	0.901159	0.861328
Q34	0.683027	0.713617	0.748990	0.668027	0.757072	0.836086	0.835167
Q35	0.750962	0.781901	0.707727	0.706958	0.775465	0.891120	0.862624
Q36	0.741410	0.772204	0.743875	0.694532	0.787044	0.877572	0.886497
Q37	0.719160	0.754757	0.743814	0.654428	0.736013	0.808107	0.863597
Q38	0.713444	0.738703	0.718405	0.620069	0.713120	0.835082	0.883311
Q39	0.738208	0.763041	0.690443	0.650786	0.688828	0.812679	0.896180
Q40	0.717965	0.756052	0.712333	0.630086	0.716196	0.796685	0.938684
Q41	0.712495	0.729924	0.666691	0.636622	0.717272	0.849093	0.894356
Q42	0.618445	0.616437	0.614892	0.580543	0.562994	0.673308	0.739629
Q43	0.546105	0.523015	0.459008	0.513790	0.387796	0.543614	0.566242
Q44	0.724733	0.708330	0.676761	0.585740	0.626303	0.741815	0.847336
Q45	0.530254	0.540485	0.487817	0.495944	0.422336	0.586874	0.546450
Q46	0.530939	0.523832	0.490177	0.550210	0.505164	0.483424	0.443465
Q47	0.649803	0.658809	0.600896	0.571223	0.623877	0.689735	0.656409
Q48	0.594826	0.640171	0.697531	0.513645	0.637911	0.588135	0.619830
Q49	1.000000	0.863896	0.730045	0.560331	0.541626	0.685991	0.740704
Q50	0.863896	1.000000	0.852839	0.622362	0.613642	0.747848	0.755248
Q51	0.730045	0.852839	1.000000	0.610769	0.650090	0.737811	0.692681

```
Q52      0.560331  0.622362  0.610769  1.000000  0.803221  0.771315  0.651478
Q53      0.541626  0.613642  0.650090  0.803221  1.000000  0.788348  0.711176
Q54      0.685991  0.747848  0.737811  0.771315  0.788348  1.000000  0.806765
Divorce  0.740704  0.755248  0.692681  0.651478  0.711176  0.806765  1.000000

[55 rows x 55 columns]
```

[22]:
```
plt.figure(figsize=(10, 8))
matrix=np.triu(divo.corr())
sns.heatmap(divo.corr(), annot=False,linewidth=.8, mask=matrix, cmap="rocket");
plt.show
```

[22]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



[23]:
```
x = divo.drop('Divorce',axis =1)
y = divo['Divorce']
```

[24]:
```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=0.
 ↪15,random_state=42)
```

[25]:
```
# importing module
from sklearn.linear_model import LogisticRegression
```

[26]:
```
LR=LogisticRegression()
```

[27]:
```
LR.fit(x_train,y_train)
```

[27]:
```
LogisticRegression()
```

[28]:
```
y_prediction=LR.predict(x_test)
y_prediction
```

[28]:
```
array([0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1,
       1, 1, 1, 1], dtype=int64)
```

[29]:
```
print("Training Accuracy :", LR.score(x_train, y_train))
print("Testing Accuracy :", LR.score(x_test, y_test))
```

```
Training Accuracy : 1.0
Testing Accuracy : 1.0
```

[30]:
```
from sklearn.tree import DecisionTreeClassifier
dt = DecisionTreeClassifier()
dt.fit(x_train, y_train)
```

[30]:
```
DecisionTreeClassifier()
```

[31]:
```
y_prediction = dt.predict(x_test)
y_prediction
```

[31]:
```
array([0, 1, 0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1,
       1, 1, 1, 0], dtype=int64)
```

[32]:
```
print("Training Accuracy :", dt.score(x_train, y_train))
print("Testing Accuracy :", dt.score(x_test, y_test))
```

```
Training Accuracy : 1.0
Testing Accuracy : 0.9230769230769231
```

[37]:
```
from sklearn.ensemble import RandomForestClassifier
```

[38]:
```
clf = RandomForestClassifier()
```

[39]:
```
clf.fit(x_train, y_train)
```

[39]:

```
RandomForestClassifier()
```

[40]:

```
y_pred = clf.predict(x_test)
```

[41]:

```
print("Training Accuracy :", clf.score(x_train, y_train))
print("Testing Accuracy :", clf.score(x_test, y_test))
```

```
Training Accuracy : 1.0
Testing Accuracy : 0.9615384615384616
```

[42]:

```
from sklearn import metrics
print()
# using metrics module for accuracy calculation
print("ACCURACY OF THE MODEL: ", metrics.accuracy_score(y_test, y_pred))
```

```

ACCURACY OF THE MODEL:  0.9615384615384616
```

[ ]:

```

```