Salsa_notebook

May 9, 2018

1 Salsa Science

The idea of this document is to understand the data collected from the client's assitance to the Salsa Colombiana workshops offered by Salsa Unicorns. Analyse the data, understand the trends and classify people into groups under certain criterion can give us a precise vision on where to move forward. This can help us to understand what the clients wants and in the marketing aspect we can offer them exactly what they are looking for.

```
In [43]: import pandas as pd
         from pandas import DataFrame, Series
         import matplotlib.pyplot as plt
         import numpy as np
         import seaborn as sns
         from collections import OrderedDict
```

```
In [44]: folder="/home/camilo/Documents/Data_Science_Courses/Salsa_Science/"
         df_liste = pd.read_csv(folder + 'Old_Model/Stats_Dataframe_Table.csv')
```

```
In [45]: df_liste.head(5)
```

```
Out[45]:      Nombre  Juli  August  September  Oktober  November  Dezember  Januar  \
         0  Client 1     1       0          2        0         0         0       0
         1  Client 2     1       0          0        1         0         0       0
         2  Client 3     0       4          0        0         0         0       0
         3  Client 4     1       0          2        0         0         0       0
         4  Client 5     0       0          0        1         0         0       0

            Total
         0      3
         1      2
         2      4
         3      3
         4      1
```

2 Find out the number of new participants per month

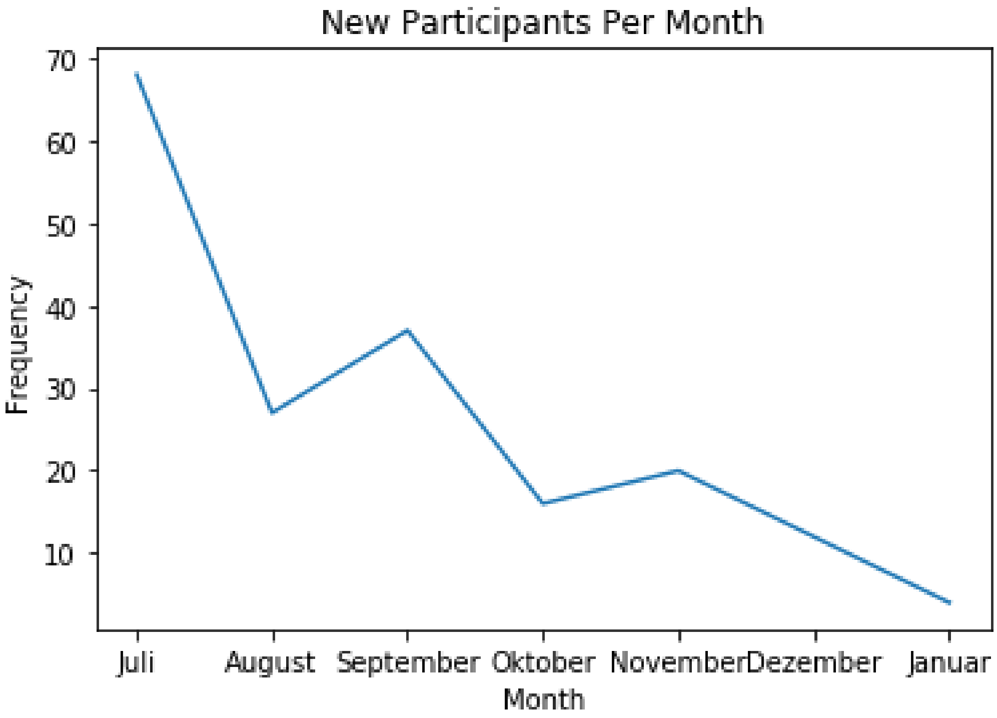
```
In [46]: newJuli = df_liste['Juli'][df_liste['Juli']>0].count()
         newAugust = df_liste['August'][(df_liste['Juli']==0) & (df_liste['August']>0)].count()
         newSeptember = df_liste['September'][(df_liste['Juli']==0) & (df_liste['August']==0) &
                                              (df_liste['September']>0)].count()
         newOktober = df_liste['Oktober'][(df_liste['Juli']==0) & (df_liste['August']==0) &
                                          (df_liste['September']==0) &
                                          (df_liste['Oktober']>0)].count()
         newNovember = df_liste['November'][(df_liste['Juli']==0) & (df_liste['August']==0) &
                                            (df_liste['September']==0) &
                                            (df_liste['Oktober']==0) &
                                            (df_liste['November']>0)].count()
         newDezember = df_liste['Dezember'][(df_liste['Juli']==0) & (df_liste['August']==0) &
                                            (df_liste['September']==0) &
                                            (df_liste['Oktober']==0) &
                                            (df_liste['November']==0) &
                                            (df_liste['Dezember']>0)].count()
         newJanuar = df_liste['Januar'][(df_liste['Juli']==0) & (df_liste['August']==0) &
                                        (df_liste['September']==0) & (df_liste['Oktober']==0) &
                                        (df_liste['November']==0) & (df_liste['Dezember']==0) &
                                        (df_liste['Januar']>0)].count()
```

```
In [47]: newParticipants=DataFrame(OrderedDict([('NewJuli', newJuli), ('NewAugust', newAugust),
                                ('NewSeptember', newSeptember), ('NewOktober', newOktober),
                                ('NewNovember', newNovember), ('NewDezember', newDezember),
                                ('NewJanuar', newJanuar)]), index=['# of new Participants'])
```

```
In [48]: newParticipants
```

```
Out[48]:                        NewJuli  NewAugust  NewSeptember  NewOktober  \
         # of new Participants       68         27            37          16

                                NewNovember  NewDezember  NewJanuar
         # of new Participants           20           12          4
```

```
In [49]: months = [7,8,9,10,11,12,13]
         #names = list(newParticipants.columns)
         values = list(newParticipants.iloc[0])
         names=['Juli','August','September','Oktober','November','Dezember','Januar']
         plt.plot(months, values)
         plt.xticks(months, names)
         plt.title('New Participants Per Month')
         plt.xlabel('Month')
         plt.ylabel('Frequency')
         plt.show()
```



3 Find out the number of participants who came different times per Month

```
In [50]: valuesJuli=df_liste['Juli'][df_liste['Juli']>0].value_counts()
         valuesAugust=df_liste['August'][df_liste['August']>0].value_counts()
         valuesSeptember=df_liste['September'][df_liste['September']>0].value_counts()
         valuesOktober=df_liste['Oktober'][df_liste['Oktober']>0].value_counts()
         valuesNovember=df_liste['November'][df_liste['November']>0].value_counts()
         valuesDezember=df_liste['Dezember'][df_liste['Dezember']>0].value_counts()
         valuesJanuar=df_liste['Januar'][df_liste['Januar']>0].value_counts()
```

```
In [51]: assistanceNumberTable=pd.concat([valuesJuli, valuesAugust, valuesSeptember,
                                          valuesOktober, valuesNovember, valuesDezember,
                                          valuesJanuar], axis=1)
```

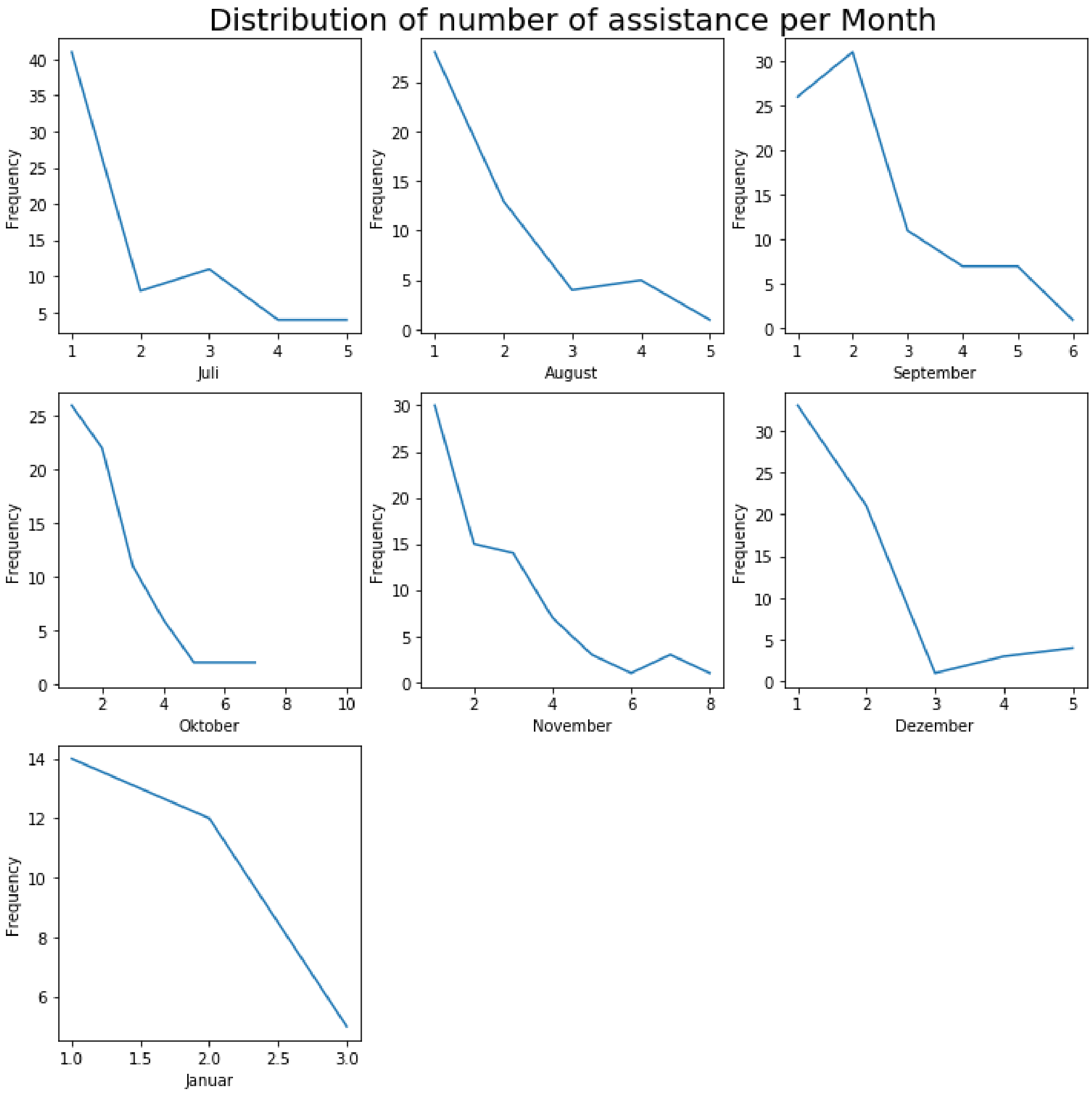
```
In [52]: assistanceNumberTable
```

```
Out[52]:     Juli  August  September  Oktober  November  Dezember  Januar
         1   41.0    28.0       26.0     26.0      30.0      33.0    14.0
         2    8.0    13.0       31.0     22.0      15.0      21.0    12.0
         3   11.0     4.0       11.0     11.0      14.0       1.0     5.0
         4    4.0     5.0        7.0      6.0       7.0       3.0     NaN
         5    4.0     1.0        7.0      2.0       3.0       4.0     NaN
```

```
6     NaN     NaN         1.0       2.0        1.0        NaN     NaN
7     NaN     NaN         NaN       2.0        3.0        NaN     NaN
8     NaN     NaN         NaN       NaN        1.0        NaN     NaN
10    NaN     NaN         NaN       1.0        NaN        NaN     NaN
```

```
In [53]: fig = plt.gcf().set_size_inches(12, 12)
         x=assistanceNumberTable.index
         y=assistanceNumberTable.Juli
         plt.subplot(331)
         plt.plot(x, y)
         plt.xlabel('Juli')
         plt.ylabel('Frequency')

         y=assistanceNumberTable.August
         plt.subplot(332)
         plt.plot(x, y)
         plt.xlabel('August')
         plt.title('Distribution of number of assistance per Month', fontsize=20)
         plt.ylabel('Frequency')

         y=assistanceNumberTable.September
         plt.subplot(333)
         plt.plot(x, y)
         plt.xlabel('September')
         plt.ylabel('Frequency')

         y=assistanceNumberTable.Oktober
         plt.subplot(334)
         plt.plot(x, y)
         plt.xlabel('Oktober')
         plt.ylabel('Frequency')

         y=assistanceNumberTable.November
         plt.subplot(335)
         plt.plot(x, y)
         plt.xlabel('November')
         plt.ylabel('Frequency')

         y=assistanceNumberTable.Dezember
         plt.subplot(336)
         plt.plot(x, y)
         plt.xlabel('Dezember')
         plt.ylabel('Frequency')

         y=assistanceNumberTable.Januar
         plt.subplot(337)
         plt.plot(x, y)
         plt.xlabel('Januar')
         plt.ylabel('Frequency')
```

```
Out[53]: <matplotlib.text.Text at 0x7f7e3cd4a7d0>
```



4 Filter out the people that came in June, since June is not taken into account for calculations

Although the June column is not there, the names of the people who came only in June and not anymore is there. In order to filtrate out these people, it is enough to look at the total column = 0. This means these people came only in June and did not come back anymore.

```
In [54]: df_filter = df_liste.drop(df_liste[df_liste.Total == 0].index)
In [55]: df_filter.head(5)
```

```
Out[55]:      Nombre  Juli  August  September  Oktober  November  Dezember  Januar  \
         0  Client 1     1       0          2        0         0         0       0
         1  Client 2     1       0          0        1         0         0       0
         2  Client 3     0       4          0        0         0         0       0
         3  Client 4     1       0          2        0         0         0       0
         4  Client 5     0       0          0        1         0         0       0

            Total
         0      3
         1      2
         2      4
         3      3
         4      1
```

```
In [56]: assisOne=df_liste['Total'][df_liste['Total']==1].count()
         assisTwo=df_liste['Total'][df_liste['Total']==2].count()
         assisThree=df_liste['Total'][df_liste['Total']==3].count()
         assisFour=df_liste['Total'][df_liste['Total']==4].count()
         assisFive=df_liste['Total'][df_liste['Total']==5].count()
         assisSix=df_liste['Total'][df_liste['Total']==6].count()
         assisSeven=df_liste['Total'][df_liste['Total']==7].count()
         assisEight=df_liste['Total'][df_liste['Total']==8].count()
         assisGreater8=df_liste['Total'][df_liste['Total']>8].count()
```

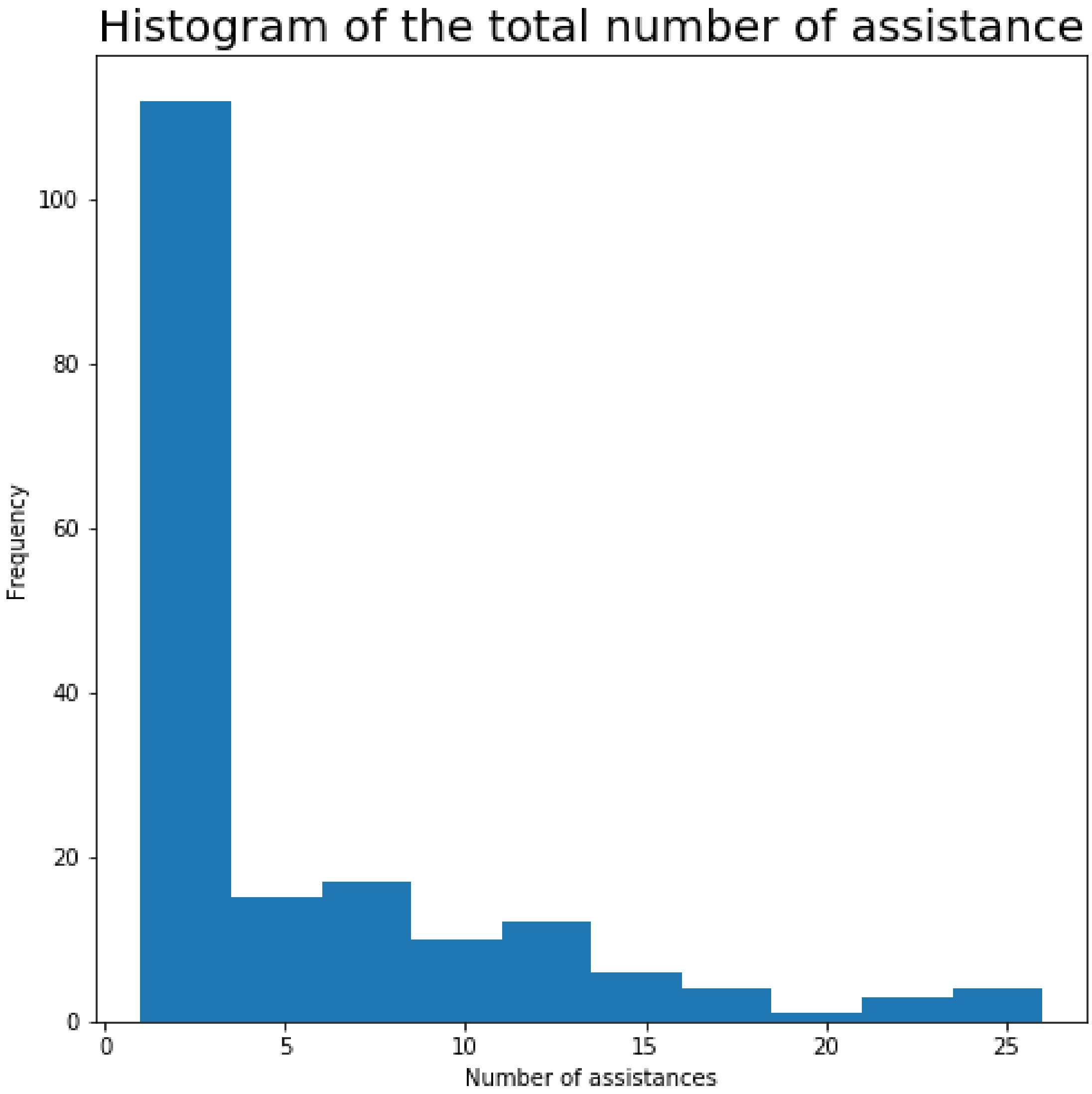
5 Find out the Number of Assistance for the whole period of six months

```
In [57]: DataFrame(OrderedDict([('OneTime', assisOne), ('TwoTimes', assisTwo),
                               ('ThreeTimes', assisThree), ('FourTimes', assisFour),
                               ('FiveTimes', assisFive), ('SixTimes', assisSix),
                               ('SevenTimes', assisSeven), ('EightTimes', assisEight),
                               ('Greater8Times', assisGreater8)]),
                               index=['Number of Assistance'])
```

```
Out[57]:                       OneTime  TwoTimes  ThreeTimes  FourTimes  FiveTimes  \
         Number of Assistance       78        20          14          6          9

                               SixTimes  SevenTimes  EightTimes  Greater8Times
         Number of Assistance         7           7           3             40
```

```
In [58]: fig=plt.gcf().set_size_inches(8, 8)
         plt.hist(x=df_filter.Total, bins=10)
         plt.title('Histogram of the total number of assistance', fontsize=20)
         plt.xlabel('Number of assistances')
         plt.ylabel('Frequency')
```

```
Out[58]: <matplotlib.text.Text at 0x7f7e3cbdca90>
```



6 Criterion to create groups

This is taken under personal criterion based on the distribution of the data in the Histogram: 1. Group 1 (the faithful): People who came 8 or more times during the 6 months period. 2. Group 2 (the irregular): People who came between 3 and 7 times during the 6 months period. 3. Group 3 (the indecisive): People who came less or equal than 2 times during the 6 months period.

```
In [59]: group1cnt=df_filter['Total'][df_filter['Total']>=8].count()
         group2cnt=df_filter['Total'][(df_filter['Total']>=3) &
                                      (df_filter['Total']<=7)].count()
         group3cnt=df_filter['Total'][df_filter['Total']<=2].count()
```

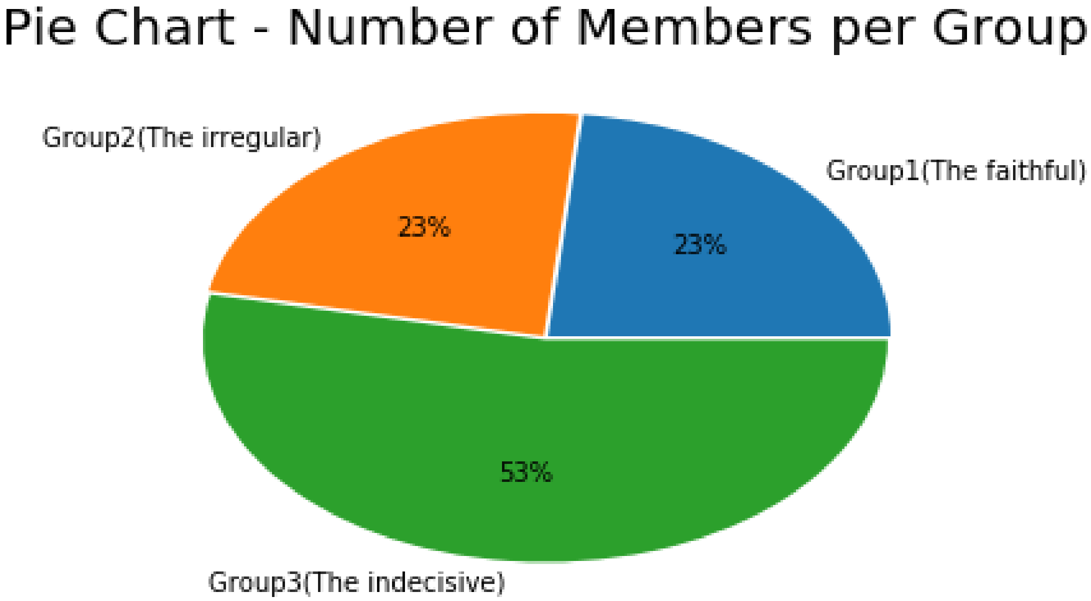
```
In [60]: groupsCount=DataFrame(OrderedDict([('Group1(the faithful)', group1cnt),
                                            ('Group2(the irregular)', group2cnt),
                                            ('Group3(the indecisive)', group3cnt)]),
                                            index=['Number of Members'])
```

```
In [61]: groupsCount
```

```
Out[61]:                    Group1(the faithful)  Group2(the irregular)  \
         Number of Members                    43                     43

                            Group3(the indecisive)
         Number of Members                      98
```

```
In [62]: x=groupsCount.loc['Number of Members']
         labels = ['Group1(The faithful)','Group2(The irregular)','Group3(The indecisive)']
         explode = (0.01, 0.01, 0.01)
         plt.pie(x, labels=labels, explode=explode, autopct='%.0f%%')
         plt.title('Pie Chart - Number of Members per Group', fontsize=20)
```

```
Out[62]: <matplotlib.text.Text at 0x7f7e3cab1110>
```



6.1 GROUP 1 - The faithful

```
In [63]: group1=df_filter[df_filter['Total']>=8]
```

```
In [64]: group1.head(5)
```

```
Out[64]:      Nombre  Juli  August  September  Oktober  November  Dezember  Januar  \
         7    Client 8     0       0          2        2         3         1       2
         9   Client 10     0       0          2        5         5         2       0
         16  Client 17     0       1          2        3         2         2       2
         17  Client 18     0       0          2        2         2         2       0
         29  Client 30     0       2          3        3         1         2       1

             Total
         7      10
         9      14
         16     12
         17      8
         29     12
```

6.2 GROUP 2 - The irregular

```
In [65]: group2=df_filter[(df_filter['Total']>=3) & (df_filter['Total']<=7)]
```

```
In [66]: #print(group2.to_string())
         group2.head(5)
```

```
Out[66]:      Nombre  Juli  August  September  Oktober  November  Dezember  Januar  \
         0    Client 1     1       0          2        0         0         0       0
         2    Client 3     0       4          0        0         0         0       0
         3    Client 4     1       0          2        0         0         0       0
         10  Client 11     0       1          2        1         0         0       0
         12  Client 13     4       1          2        0         0         0       0

             Total
         0       3
         2       4
         3       3
         10      4
         12      7
```

6.3 GROUP 3 - The indecisive

```
In [67]: group3=df_filter[df_filter['Total']<=2]
```

```
In [68]: group3.head(5)
```

```
Out[68]:     Nombre  Juli  August  September  Oktober  November  Dezember  Januar  \
         1  Client 2     1       0          0        1         0         0       0
         4  Client 5     0       0          0        1         0         0       0
         5  Client 6     0       0          0        0         0         0       1
         6  Client 7     0       0          0        0         1         0       0
         8  Client 9     0       0          2        0         0         0       0

            Total
```

```
1      2
4      1
5      1
6      1
8      2
```