Group Number = 9

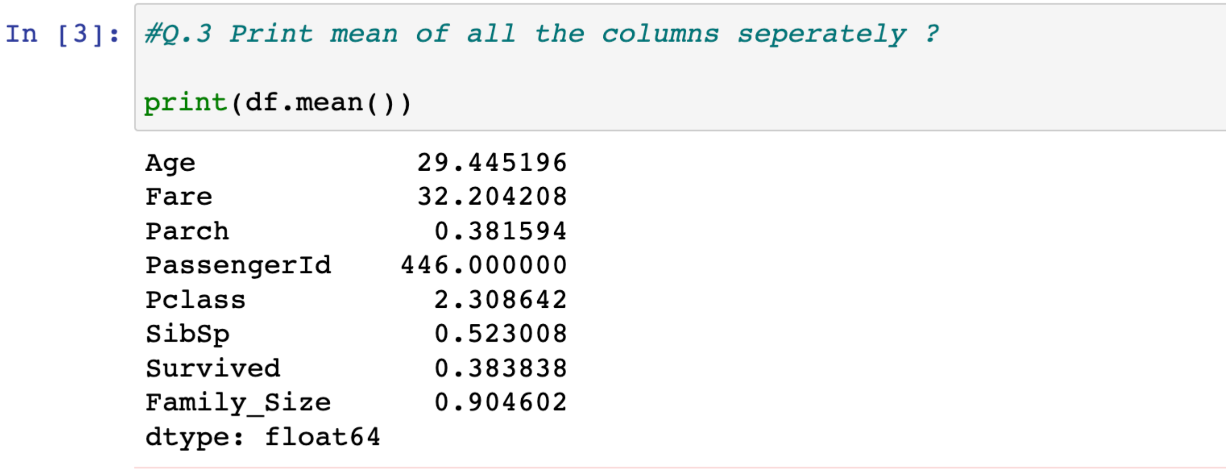
230 (Aditya Babar)
231 (Janhavi Kawadkar)
236 (Rohan Agrawal)

```
In [1]: #Q.1 Print whole dataset using pandas ?

        import pandas as pd

        df = pd.read_csv("titanic.csv")

        print(df)
```

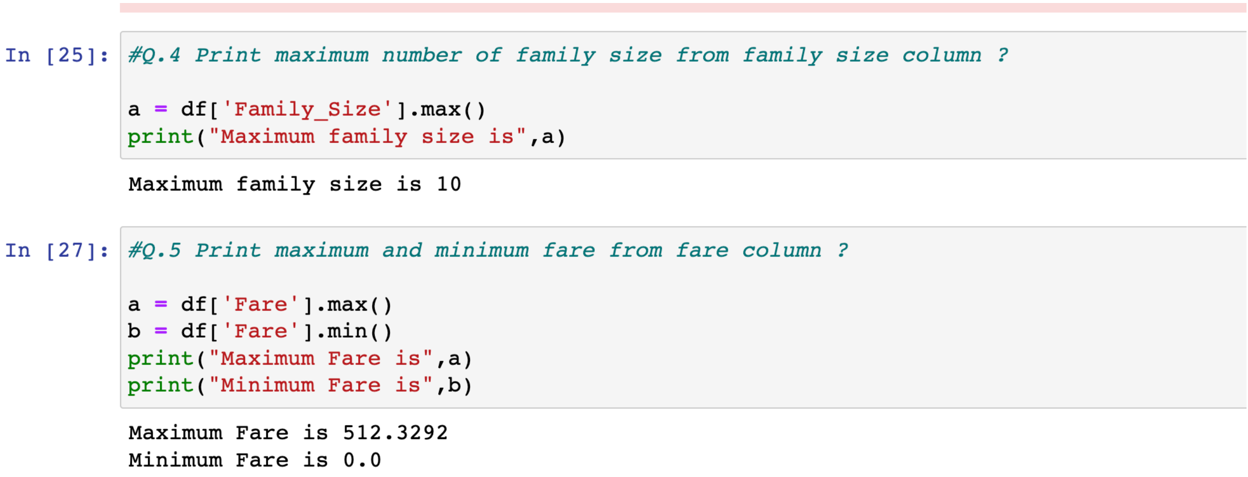
```
      Age Cabin Embarked     Fare  \
0    22.0   NaN        S   7.2500
1    38.0   C85        C  71.2833
2    26.0   NaN        S   7.9250
3    35.0  C123        S  53.1000
4    35.0   NaN        S   8.0500
..    ...   ...      ...      ...
886  27.0   NaN        S  13.0000
887  19.0   B42        S  30.0000
888  22.0   NaN        S  23.4500
889  26.0  C148        C  30.0000
890  32.0   NaN        Q   7.7500

                                                  Name  Parch  PassengerId  \
0                              Braund, Mr. Owen Harris      0            1
1    Cumings, Mrs. John Bradley (Florence Briggs Th...      0            2
2                               Heikkinen, Miss. Laina      0            3
3         Futrelle, Mrs. Jacques Heath (Lily May Peel)      0            4
```

```
In [2]: #Q.2 Print all attributes like count,mean,std,min of numerical columns using one function ?

        print(df.describe())
```

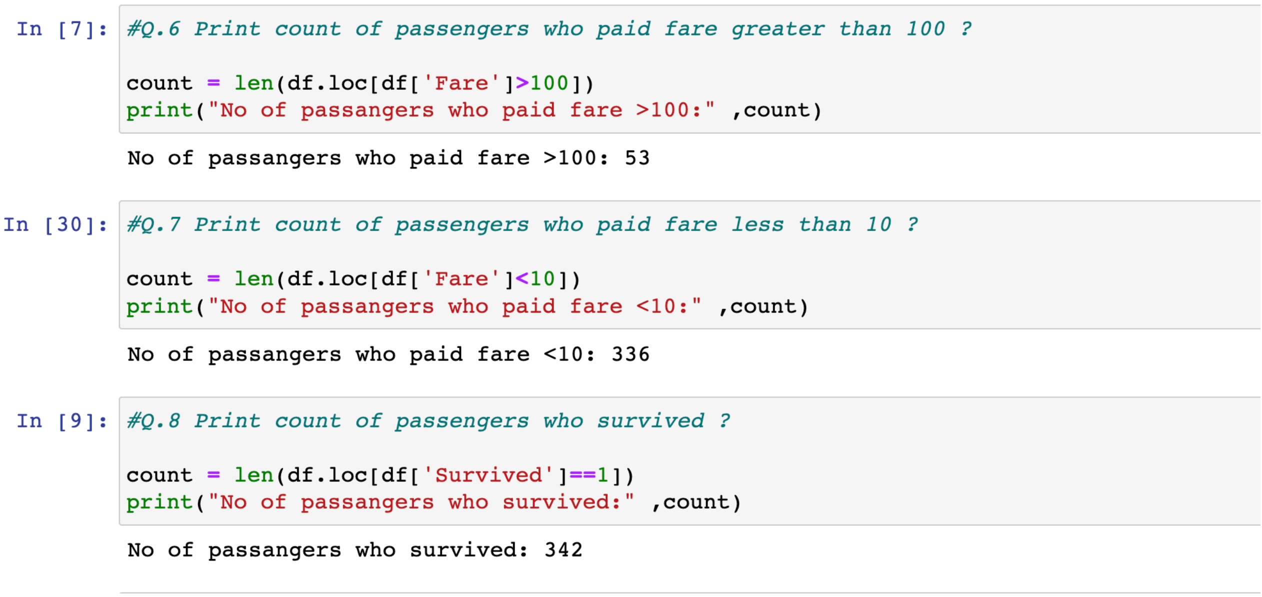
```
              Age        Fare       Parch  PassengerId      Pclass  \
count  891.000000  891.000000  891.000000   891.000000  891.000000
mean    29.445196   32.204208    0.381594   446.000000    2.308642
std     13.244896   49.693429    0.806057   257.353842    0.836071
min      0.420000    0.000000    0.000000     1.000000    1.000000
25%     22.000000    7.910400    0.000000   223.500000    2.000000
50%     30.000000   14.454200    0.000000   446.000000    3.000000
75%     35.500000   31.000000    0.000000   668.500000    3.000000
max     80.000000  512.329200    6.000000   891.000000    3.000000

            SibSp    Survived  Family_Size
count  891.000000  891.000000   891.000000
mean     0.523008    0.383838     0.904602
std      1.102743    0.486592     1.613459
min      0.000000    0.000000     0.000000
25%      0.000000    0.000000     0.000000
50%      0.000000    0.000000     0.000000
75%      1.000000    1.000000     1.000000
max      8.000000    1.000000    10.000000
```



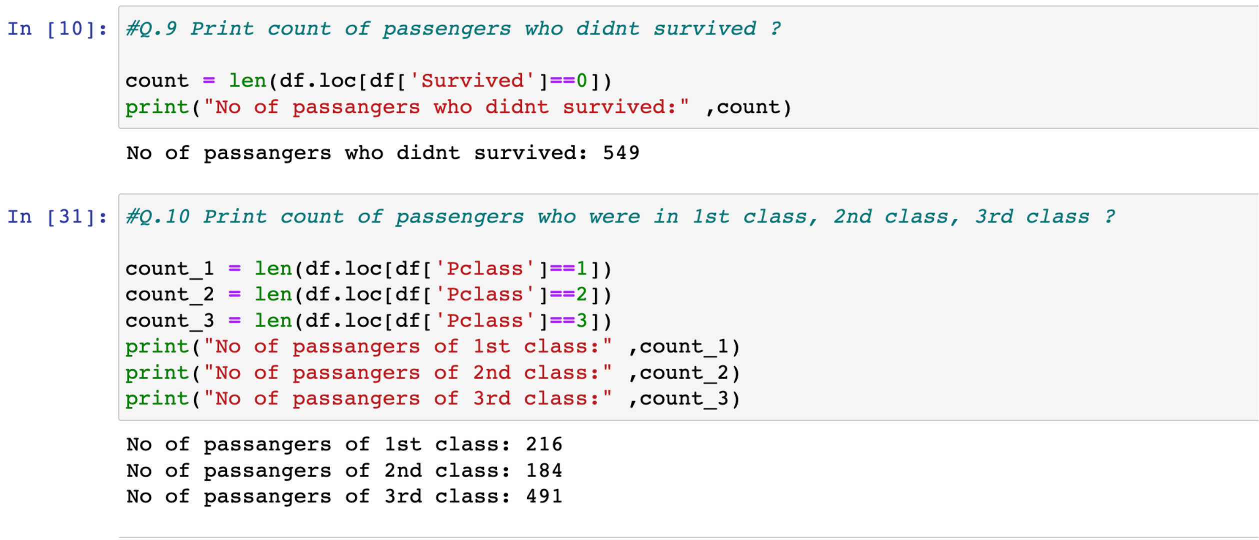
```
In [3]: #Q.3 Print mean of all the columns seperately ?

        print(df.mean())

        Age             29.445196
        Fare            32.204208
        Parch            0.381594
        PassengerId    446.000000
        Pclass           2.308642
        SibSp            0.523008
        Survived         0.383838
        Family_Size      0.904602
        dtype: float64
```



```
In [25]: #Q.4 Print maximum number of family size from family size column ?

         a = df['Family_Size'].max()
         print("Maximum family size is",a)

         Maximum family size is 10

In [27]: #Q.5 Print maximum and minimum fare from fare column ?

         a = df['Fare'].max()
         b = df['Fare'].min()
         print("Maximum Fare is",a)
         print("Minimum Fare is",b)

         Maximum Fare is 512.3292
         Minimum Fare is 0.0
```



```
In [7]: #Q.6 Print count of passengers who paid fare greater than 100 ?

        count = len(df.loc[df['Fare']>100])
        print("No of passangers who paid fare >100:" ,count)

        No of passangers who paid fare >100: 53

In [30]: #Q.7 Print count of passengers who paid fare less than 10 ?

        count = len(df.loc[df['Fare']<10])
        print("No of passangers who paid fare <10:" ,count)

        No of passangers who paid fare <10: 336

In [9]: #Q.8 Print count of passengers who survived ?

        count = len(df.loc[df['Survived']==1])
        print("No of passangers who survived:" ,count)

        No of passangers who survived: 342
```

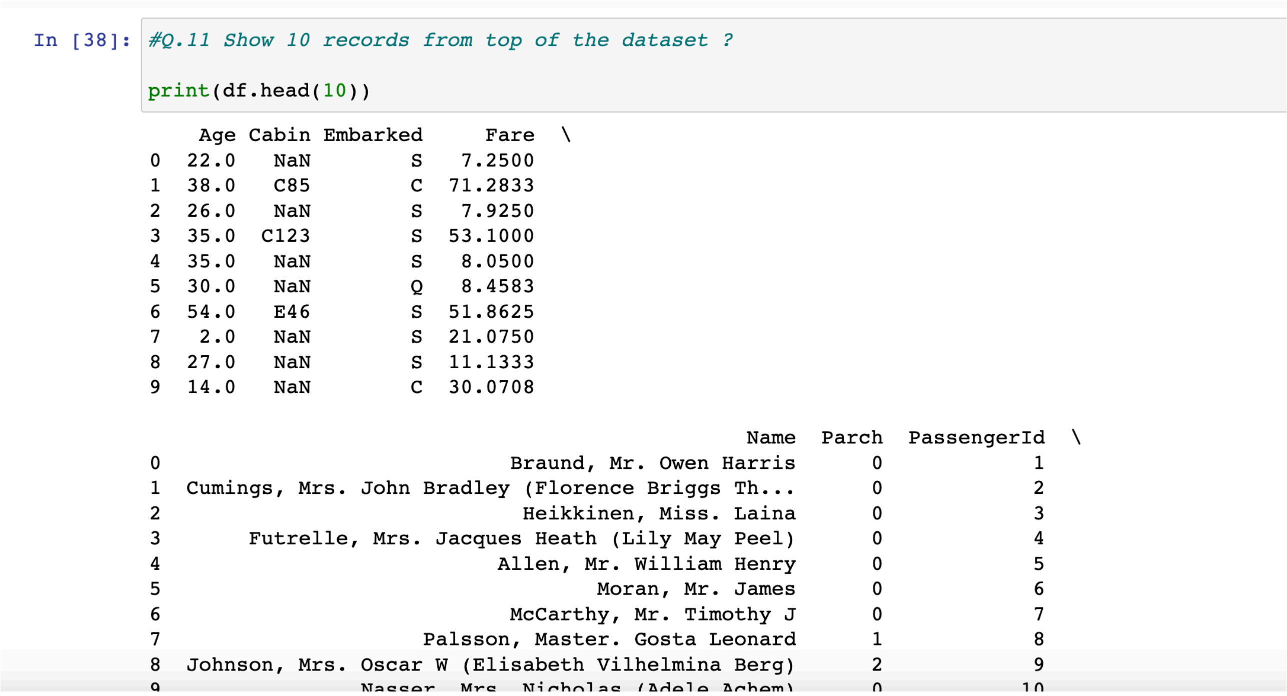


In [10]:
```
#Q.9 Print count of passengers who didnt survived ?

count = len(df.loc[df['Survived']==0])
print("No of passangers who didnt survived:" ,count)
```

```
No of passangers who didnt survived: 549
```

In [31]:
```
#Q.10 Print count of passengers who were in 1st class, 2nd class, 3rd class ?

count_1 = len(df.loc[df['Pclass']==1])
count_2 = len(df.loc[df['Pclass']==2])
count_3 = len(df.loc[df['Pclass']==3])
print("No of passangers of 1st class:" ,count_1)
print("No of passangers of 2nd class:" ,count_2)
print("No of passangers of 3rd class:" ,count_3)
```

```
No of passangers of 1st class: 216
No of passangers of 2nd class: 184
No of passangers of 3rd class: 491
```



In [38]:
```
#Q.11 Show 10 records from top of the dataset ?

print(df.head(10))
```

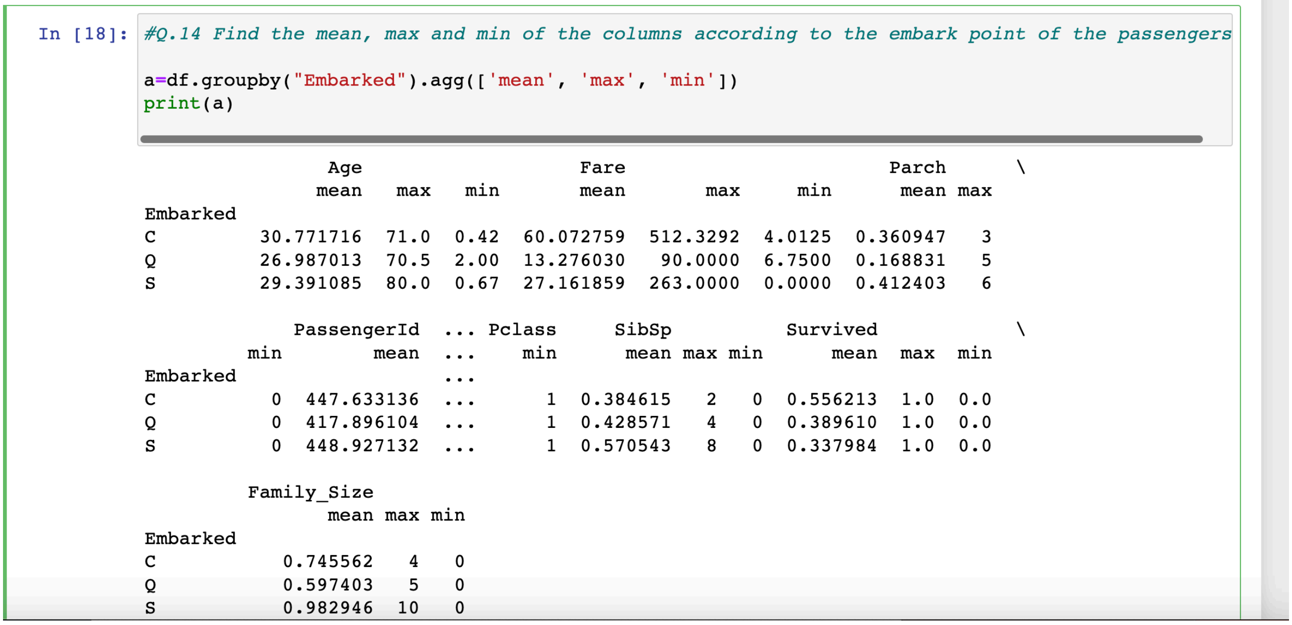
```
    Age Cabin Embarked     Fare  \
0  22.0   NaN        S   7.2500   
1  38.0   C85        C  71.2833   
2  26.0   NaN        S   7.9250   
3  35.0  C123        S  53.1000   
4  35.0   NaN        S   8.0500   
5  30.0   NaN        Q   8.4583   
6  54.0   E46        S  51.8625   
7   2.0   NaN        S  21.0750   
8  27.0   NaN        S  11.1333   
9  14.0   NaN        C  30.0708   

                                                Name  Parch  PassengerId  \
0                            Braund, Mr. Owen Harris      0            1   
1  Cumings, Mrs. John Bradley (Florence Briggs Th...      0            2   
2                             Heikkinen, Miss. Laina      0            3   
3       Futrelle, Mrs. Jacques Heath (Lily May Peel)      0            4   
4                           Allen, Mr. William Henry      0            5   
5                                   Moran, Mr. James      0            6   
6                            McCarthy, Mr. Timothy J      0            7   
7                     Palsson, Master. Gosta Leonard      1            8   
8  Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)      2            9   
9                Nasser, Mrs. Nicholas (Adele Achem)      0           10   
```

```
In [15]: #Q.12 Print the value of total fare collected ?

         print(df['Fare'].sum())
```

```
28693.9493
```

```
In [16]: #Q.13 Use groupby function to count the number of passengers embarked from C,Q,S spot ?

         a=df.groupby("Embarked").count()
         print(a)
```

```
          Age  Cabin  Fare  Name  Parch  PassengerId  Pclass  Sex  SibSp  \
Embarked
C         169     70   169   169    169          169     169  169    169
Q          77      4    77    77     77           77      77   77     77
S         645    130   645   645    645          645     645  645    645

          Survived  Ticket  Title  Family_Size
Embarked
C              169     169    169          169
Q               77      77     77           77
S              645     645    645          645
```



```
In [18]: #Q.14 Find the mean, max and min of the columns according to the embark point of the passengers

         a=df.groupby("Embarked").agg(['mean', 'max', 'min'])
         print(a)
```

```
                Age                   Fare                       Parch      \
               mean   max   min       mean       max     min      mean max
Embarked
C         30.771716  71.0  0.42  60.072759  512.3292  4.0125  0.360947   3
Q         26.987013  70.5  2.00  13.276030   90.0000  6.7500  0.168831   5
S         29.391085  80.0  0.67  27.161859  263.0000  0.0000  0.412403   6

             PassengerId  ... Pclass     SibSp          Survived            \
         min        mean  ...    min      mean max min      mean  max  min
Embarked                  ...
C          0  447.633136  ...      1  0.384615   2   0  0.556213  1.0  0.0
Q          0  417.896104  ...      1  0.428571   4   0  0.389610  1.0  0.0
S          0  448.927132  ...      1  0.570543   8   0  0.337984  1.0  0.0

         Family_Size
                mean max min
Embarked
C           0.745562   4   0
Q           0.597403   5   0
S           0.982946  10   0
```

```
In [55]: #Q.15 Find Correlation and Covarience between different columns ?

         a = df.corr()
         b = df.cov()
         print("Correlation \n",a,"\n")
         print("Covariance \n",b)
```

```
Correlation 
                   Age      Fare     Parch  PassengerId    Pclass     SibSp  \
Age          1.000000  0.097578 -0.188611     0.042645 -0.352827 -0.263826   
Fare         0.097578  1.000000  0.216225     0.012658 -0.549500  0.159651   
Parch       -0.188611  0.216225  1.000000    -0.001652  0.018443  0.414838   
PassengerId  0.042645  0.012658 -0.001652     1.000000 -0.035144 -0.057527   
Pclass      -0.352827 -0.549500  0.018443    -0.035144  1.000000  0.083081   
SibSp       -0.263826  0.159651  0.414838    -0.057527  0.083081  1.000000   
Survived    -0.077653  0.257307  0.081629    -0.005007 -0.338481 -0.035322   
Family_Size -0.274543  0.217138  0.783111    -0.040143  0.065997  0.890712   

             Survived  Family_Size  
Age         -0.077653    -0.274543  
Fare         0.257307     0.217138  
Parch        0.081629     0.783111  
PassengerId -0.005007    -0.040143  
Pclass      -0.338481     0.065997  
SibSp       -0.035322     0.890712  
Survived     1.000000     0.016639  
Family_Size  0.016639     1.000000  
```

```
Covariance 
                    Age         Fare     Parch   PassengerId     Pclass  \
Age          175.540463    64.245317 -2.014288    145.408989  -3.908351   
Fare          64.245317  2469.436846  8.661052    161.883369 -22.830196   
Parch         -2.014288     8.661052  0.649728     -0.342697   0.012429   
PassengerId  145.408989   161.883369 -0.342697  66231.000000  -7.561798   
Pclass        -3.908351   -22.830196  0.012429     -7.561798   0.699015   
SibSp         -3.854612     8.748734  0.368739    -16.325843   0.076599   
Survived      -0.500624     6.221787  0.032017     -0.626966  -0.137703   
Family_Size   -5.868900    17.409785  1.018467    -16.668539   0.089028   

                 SibSp  Survived  Family_Size  
Age          -3.854612 -0.500624    -5.868900  
Fare          8.748734  6.221787    17.409785  
Parch         0.368739  0.032017     1.018467  
PassengerId -16.325843 -0.626966   -16.668539  
Pclass        0.076599 -0.137703     0.089028  
SibSp         1.216043 -0.018954     1.584782  
Survived     -0.018954  0.236772     0.013063  
Family_Size   1.584782  0.013063     2.603248  
```

```
In [50]: #Q.16 Find 0.25,0.5,0.75 of fare column and age column ?

         a = df['Fare'].quantile([0.25, 0.5, 0.75])
         b = df['Age'].quantile([0.25, 0.5, 0.75])

         print("Fare \n",a,"\n")
         print("Age \n",b)
```

```
Fare 
 0.25     7.9104
0.50    14.4542
0.75    31.0000
Name: Fare, dtype: float64 

Age 
 0.25    22.0
0.50    30.0
0.75    35.5
Name: Age, dtype: float64
```

```
In [52]: #Q.17 Find missing data and fill it with 0 ?

         b = df.isnull()
         print(b)
         print("\n")
         d = df.fillna(0)
         print(d)
```

```
       Age  Cabin  Embarked   Fare   Name  Parch  PassengerId  Pclass    Sex  \
0    False   True     False  False  False  False        False   False  False   
1    False  False     False  False  False  False        False   False  False   
2    False   True     False  False  False  False        False   False  False   
3    False  False     False  False  False  False        False   False  False   
4    False   True     False  False  False  False        False   False  False   
..     ...    ...       ...    ...    ...    ...          ...     ...    ...   
886  False   True     False  False  False  False        False   False  False   
887  False  False     False  False  False  False        False   False  False   
888  False   True     False  False  False  False        False   False  False   
889  False  False     False  False  False  False        False   False  False   
890  False   True     False  False  False  False        False   False  False   

     SibSp  Survived  Ticket  Title  Family_Size  
0    False     False   False  False        False  
1    False     False   False  False        False  
2    False     False   False  False        False  
3    False     False   False  False        False  
4    False     False   False  False        False  
```

```
      Age Cabin Embarked     Fare  \
0    22.0     0        S   7.2500   
1    38.0   C85        C  71.2833   
2    26.0     0        S   7.9250   
3    35.0  C123        S  53.1000   
4    35.0     0        S   8.0500   
..    ...   ...      ...      ...   
886  27.0     0        S  13.0000   
887  19.0   B42        S  30.0000   
888  22.0     0        S  23.4500   
889  26.0  C148        C  30.0000   
890  32.0     0        Q   7.7500   

                                                  Name  Parch  PassengerId  \
0                              Braund, Mr. Owen Harris      0            1   
1    Cumings, Mrs. John Bradley (Florence Briggs Th...      0            2   
2                               Heikkinen, Miss. Laina      0            3   
3         Futrelle, Mrs. Jacques Heath (Lily May Peel)      0            4   
4                             Allen, Mr. William Henry      0            5   
..                                                 ...    ...          ...   
886                              Montvila, Rev. Juozas      0          887   
887                       Graham, Miss. Margaret Edith      0          888   
888           Johnston, Miss. Catherine Helen "Carrie"      2          889   
889                              Behr, Mr. Karl Howell      0          890   
890                                Dooley, Mr. Patrick      0          891   
```

In [53]:

```
#Q.18 Convert the datatype of Age column from float to integer ?

a = df['Age'] = df['Age'].astype('int')
print(a)
```

```
0      22
1      38
2      26
3      35
4      35
       ..
886    27
887    19
888    22
889    26
890    32
Name: Age, Length: 891, dtype: int64
```

In [22]:

```
#Q.19 Concat name and title column and print it ?

df1 = df["Name"]
df2 = df["Title"]
merged_df = pd.concat([df1, df2], axis=0)
print(merged_df)
```

```
0                               Braund, Mr. Owen Harris
1       Cumings, Mrs. John Bradley (Florence Briggs Th...
2                                Heikkinen, Miss. Laina
3            Futrelle, Mrs. Jacques Heath (Lily May Peel)
4                              Allen, Mr. William Henry
                             ...
886                                                 Rev
887                                                Miss
888                                                Miss
889                                                  Mr
890                                                  Mr
Length: 1782, dtype: object
```

In [29]:

```
#Q.20 Do square operation on fare column and print its output in a seperate new column ?

df['Value_square'] = df['Fare'] ** 2
print(df)
```

```
     Age Cabin Embarked     Fare  \
0     22   NaN        S   7.2500   
1     38   C85        C  71.2833   
2     26   NaN        S   7.9250   
3     35  C123        S  53.1000   
4     35   NaN        S   8.0500   
..   ...   ...      ...      ...   
886   27   NaN        S  13.0000   
887   19   B42        S  30.0000   
888   22   NaN        S  23.4500   
889   26  C148        C  30.0000   
890   32   NaN        Q   7.7500   

                                                  Name  Parch  PassengerId  \
0                              Braund, Mr. Owen Harris      0            1   
1    Cumings, Mrs. John Bradley (Florence Briggs Th...      0            2   
2                               Heikkinen, Miss. Laina      0            3   
3         Futrelle, Mrs. Jacques Heath (Lily May Peel)      0            4   
4                             Allen, Mr. William Henry      0            5   
..                                                 ...    ...          ...   
886                              Montvila, Rev. Juozas      0          887   
887                       Graham, Miss. Margaret Edith      0          888   
```

```
     Pclass     Sex  SibSp  Survived            Ticket  Title  Family_Size  \
0         3    Male      1       0.0         A/5 21171     Mr            1   
1         1  Female      1       1.0          PC 17599    Mrs            1   
2         3  Female      0       1.0  STON/O2. 3101282   Miss            0   
3         1  Female      1       1.0            113803    Mrs            1   
4         3    Male      0       0.0            373450     Mr            0   
..      ...     ...    ...       ...               ...    ...          ...   
886       2    Male      0       0.0            211536    Rev            0   
887       1  Female      0       1.0            112053   Miss            0   
888       3  Female      1       0.0        W./C. 6607   Miss            3   
889       1    Male      0       1.0            111369     Mr            0   
890       3    Male      0       0.0            370376     Mr            0   

     Value_square  
0       52.562500  
1     5081.308859  
2       62.805625  
3     2819.610000  
4       64.802500  
..            ...  
886    169.000000  
887    900.000000  
888    549.902500  
889    900.000000  
890     60.062500  

[891 rows x 15 columns]
```