DVA_lab-6a_205229118_Mahalakshmi.S

April 21, 2021

0.0.1 Lab 6a: Fundamental Pre processing functions in Pandas

0.0.2 1. Import NumPy and Pandas.

[1]:
```
import numpy as np
import pandas as pd
```

0.0.3 2. Redad data.csv file.

[2]:
```
data = pd.read_csv("data.csv")
data
```

[2]:
```
      RowNumber  CustomerId    Surname  CreditScore Geography  Gender  Age  \
0             1    15634602   Hargrave          619    France  Female   42
1             2    15647311       Hill          608     Spain  Female   41
2             3    15619304       Onio          502    France  Female   42
3             4    15701354       Boni          699    France  Female   39
4             5    15737888   Mitchell          850     Spain  Female   43
…           …          …          …           …          …      …   …
9995       9996    15606229   Obijiaku          771    France    Male   39
9996       9997    15569892  Johnstone          516    France    Male   35
9997       9998    15584532        Liu          709    France  Female   36
9998       9999    15682355  Sabbatini          772   Germany    Male   42
9999      10000    15628319     Walker          792    France  Female   28

      Tenure    Balance  NumOfProducts  HasCrCard  IsActiveMember  \
0          2       0.00              1          1               1
1          1   83807.86              1          0               1
2          8  159660.80              3          1               0
3          1       0.00              2          0               0
4          2  125510.82              1          1               1
…       …         …              …          …               …
9995       5       0.00              2          1               0
9996      10   57369.61              1          1               1
9997       7       0.00              1          0               1
9998       3   75075.31              2          1               0
9999       4  130142.79              1          1               0

      EstimatedSalary  Exited
```

```
0          101348.88       1
1          112542.58       0
2          113931.57       1
3           93826.63       0
4           79084.10       0
…                 …      …
9995        96270.64       0
9996       101699.77       0
9997        42085.58       1
9998        92888.52       1
9999        38190.78       0

[10000 rows x 14 columns]
```

0.0.4 3. Display the shape of data frame.

[3]:
```
data.shape
```

[3]:
```
(10000, 14)
```

0.0.5 4. Display all colum names.

[4]:
```
data.columns
```

[4]:
```
Index(['RowNumber', 'CustomerId', 'Surname', 'CreditScore', 'Geography',
       'Gender', 'Age', 'Tenure', 'Balance', 'NumOfProducts', 'HasCrCard',
       'IsActiveMember', 'EstimatedSalary', 'Exited'],
      dtype='object')
```

0.0.6 5. Remove following colums from data set 'RowNumber', 'CustomerId', 'Surname', 'CreditScore' and display shape of data frame.

[5]:
```
data.drop(['RowNumber', 'CustomerId', 'Surname', 'CreditScore'],axis=1,␣
 ↪inplace=True)
```

[6]:
```
data
```

[6]:
```
     Geography  Gender  Age  Tenure    Balance  NumOfProducts  HasCrCard  \
0       France  Female   42       2       0.00              1          1
1        Spain  Female   41       1   83807.86              1          0
2       France  Female   42       8  159660.80              3          1
3       France  Female   39       1       0.00              2          0
4        Spain  Female   43       2  125510.82              1          1
…         …     …   …    …        …               …          …
9995    France    Male   39       5       0.00              2          1
9996    France    Male   35      10   57369.61              1          1
9997    France  Female   36       7       0.00              1          0
```

```
9998   Germany    Male   42        3   75075.31              2            1
9999    France  Female   28        4  130142.79              1            1

      IsActiveMember  EstimatedSalary  Exited
0                  1        101348.88       1
1                  1        112542.58       0
2                  0        113931.57       1
3                  0         93826.63       0
4                  1         79084.10       0
…                …               …     …
9995               0         96270.64       0
9996               1        101699.77       0
9997               1         42085.58       1
9998               0         92888.52       1
9999               0         38190.78       0

[10000 rows x 10 columns]
```

[7]:
```python
data.shape
```

[7]:
```
(10000, 10)
```

0.0.7 6. Read following colums from csv file 'Gender', 'Age', 'Tenure', 'Balance'.

[8]:
```python
data_read = pd.read_csv("data.csv", usecols=['Gender', 'Age', 'Tenure',␣
↪'Balance'])
```

[9]:
```python
data_read
```

[9]:
```
      Gender  Age  Tenure    Balance
0     Female   42       2       0.00
1     Female   41       1   83807.86
2     Female   42       8  159660.80
3     Female   39       1       0.00
4     Female   43       2  125510.82
…        …   …      …         …
9995    Male   39       5       0.00
9996    Male   35      10   57369.61
9997  Female   36       7       0.00
9998    Male   42       3   75075.31
9999  Female   28       4  130142.79

[10000 rows x 4 columns]
```

0.0.8 7. Read first 3000 rows from csv file.

[10]:
```
data_par = data[:3000]
```

[11]:
```
print(data_par)
```

```
     Geography  Gender  Age  Tenure    Balance  NumOfProducts  HasCrCard  \
0       France  Female   42       2       0.00              1          1
1        Spain  Female   41       1   83807.86              1          0
2       France  Female   42       8  159660.80              3          1
3       France  Female   39       1       0.00              2          0
4        Spain  Female   43       2  125510.82              1          1
…          …       …  …     …       …          …          …
2995    France  Female   29       2  112367.34              1          1
2996     Spain  Female   45       7   91091.06              2          1
2997    France  Female   26       7  106198.50              1          0
2998    France    Male   47       5  142669.93              2          1
2999    France    Male   33       1  112833.35              1          0

      IsActiveMember  EstimatedSalary  Exited
0                  1        101348.88       1
1                  1        112542.58       0
2                  0        113931.57       1
3                  0         93826.63       0
4                  1         79084.10       0
…               …              …     …
2995               0        185630.76       0
2996               0         71133.12       0
2997               1         32020.42       0
2998               0        162760.96       0
2999               1        175178.56       0

[3000 rows x 10 columns]
```

0.0.9 8. Take 1000 rows as a sample data from data frame.

[12]:
```
data_sam = data.sample(n=1000)
data_sam
```

[12]:
```
     Geography  Gender  Age  Tenure    Balance  NumOfProducts  HasCrCard  \
1568    France    Male   40       8  114005.78              1          1
3615    France    Male   46       9  134950.19              3          0
6343     Spain  Female   55       6       0.00              1          0
7235    France    Male   35       5  133087.76              1          1
7433   Germany  Female   37       2  128389.63              1          1
…          …       …  …     …       …          …          …
650     France  Female   49       4       0.00              2          1
3712     Spain  Female   39       1  141789.15              1          1
```

```
6892    France  Female   56       8  156974.26              1            1
6544    France  Female   45       3  104118.50              1            0
1514     Spain    Male   38       8   71460.67              2            1

      IsActiveMember  EstimatedSalary  Exited
1568               1         67998.45       0
3615               0        178587.36       1
6343               0         91943.94       1
7235               0         64771.61       0
7433               1          6589.16       1
…                …                …      …
650                1        196335.48       0
3712               0         92455.96       0
6892               0         89405.26       1
6544               1        174032.00       0
1514               1         10074.05       0

[1000 rows x 10 columns]
```

0.0.10 9. Take 10% of rows as a sample data from data frame.

[13]:
```python
data_sam2 = data.sample(frac=0.1)
data_sam2
```

[13]:
```
     Geography  Gender  Age  Tenure    Balance  NumOfProducts  HasCrCard  \
662     France    Male   31       2       0.00              2          1
1916   Germany    Male   48       1  100900.50              1          0
610     France    Male   30      10  129755.99              1          0
6743   Germany  Female   51       9  138214.50              1          1
272    Germany  Female   34       1  149297.19              2          1
…         …      …    …     …       …          …            …
5156     Spain    Male   33       5  127343.40              1          0
1356    France    Male   49       4  154344.49              2          1
3943    France    Male   46       5       0.00              2          1
6897    France  Female   48       6  127253.98              1          1
6773    France    Male   43       4  122351.29              1          1

      IsActiveMember  EstimatedSalary  Exited
662                1         58803.28       0
1916               0         33310.72       1
610                0        172749.65       0
6743               0        198715.27       1
272                1        186339.74       0
…                …                …      …
5156               1        121789.30       0
1356               1         38794.57       0
3943               1         76946.60       0
```

```
6897                 1          92144.09       1
6773                 0          71216.60       0

[1000 rows x 10 columns]
```

0.0.11 10. Count NA value in all colums.

[14]:
```
data.isna().sum()
```

[14]:
```
Geography          0
Gender             0
Age                0
Tenure             0
Balance            0
NumOfProducts      0
HasCrCard          0
IsActiveMember     0
EstimatedSalary    0
Exited             0
dtype: int64
```

0.0.12 11. Create 20 random indices as 'missing_index' .

[15]:
```
missing_index = np.random.randint(10000, size=20)
```

[16]:
```
missing_index
```

[16]:
```
array([8406, 2291, 6785, 8820, 1509,  272, 4180, 3895, 3266, 3366,  757,
       9788, 6805,  380, 9319, 7230, 4764, 7719, 8979, 4107])
```

0.0.13 12. Create 20 missing values in the "Balance" and "Geography" columns using 'missing_index'.

[17]:
```
data.loc[missing_index, ['Balance','Geography']] = np.nan
```

[18]:
```
data.loc[missing_index, ['Balance','Geography']]   ##20 missing values in␣
 ↪balance and geography columns.
```

[18]:
```
      Balance Geography
8406      NaN       NaN
2291      NaN       NaN
6785      NaN       NaN
8820      NaN       NaN
1509      NaN       NaN
272       NaN       NaN
4180      NaN       NaN
3895      NaN       NaN
```

```
3266      NaN       NaN
3366      NaN       NaN
757       NaN       NaN
9788      NaN       NaN
6805      NaN       NaN
380       NaN       NaN
9319      NaN       NaN
7230      NaN       NaN
4764      NaN       NaN
7719      NaN       NaN
8979      NaN       NaN
4107      NaN       NaN
```

0.0.14 13. Create missing values in the index of the last column using 'missing_index'.

[19]:
```
data.iloc[missing_index, -1] = np.nan
```

[20]:
```
data.iloc[missing_index, -1]
```

[20]:
```
8406    NaN
2291    NaN
6785    NaN
8820    NaN
1509    NaN
272     NaN
4180    NaN
3895    NaN
3266    NaN
3366    NaN
757     NaN
9788    NaN
6805    NaN
380     NaN
9319    NaN
7230    NaN
4764    NaN
7719    NaN
8979    NaN
4107    NaN
Name: Exited, dtype: float64
```

0.0.15 14. Find the most common value in the 'geography' column.

[21]:
```
data_geo = data['Geography'].value_counts()
```

[22]:
```
data_geo
```

```
[22]: France     5006
      Germany    2503
      Spain      2471
      Name: Geography, dtype: int64
```

0.0.16 15. Find mean of the 'balance' column, and replace mean value with its missing values.

```
[23]: mea = data['Balance'].mean()
```

```
[24]: mea
```

```
[24]: 76500.49372444849
```

```
[25]: data['Balance'].fillna(value=mea)
```

```
[25]: 0            0.00
      1        83807.86
      2       159660.80
      3            0.00
      4       125510.82
                ...
      9995         0.00
      9996     57369.61
      9997         0.00
      9998     75075.31
      9999    130142.79
      Name: Balance, Length: 10000, dtype: float64
```

0.0.17 16. Drop missing values in the last colum.

```
[26]: data.Exited.isnull().sum()
```

```
[26]: 20
```

0.0.18 17. Count selecting rows based on following conditions (Geography == 'France') & (Exited == 1).

```
[27]: fran = data[(data.Geography == 'France') & (data.Exited == 1)]
      fran
```

```
[27]:      Geography  Gender  Age  Tenure    Balance  NumOfProducts  HasCrCard  \
      0       France  Female   42       2       0.00              1          1
      2       France  Female   42       8  159660.80              3          1
      35      France  Female   45       0  134264.04              1          1
      41      France  Female   51       8  122522.32              1          0
      43      France  Female   49       2  131394.56              1          0
      …         …       …  …    …        …            …          …
```

```
9920   France  Female   49       3  204510.94              1            0
9947   France    Male   34       1   83503.11              2            1
9956   France  Female   46      10   85216.61              1            1
9991   France  Female   53       4   88381.21              1            1
9997   France  Female   36       7       0.00              1            0

      IsActiveMember  EstimatedSalary  Exited
0                  1        101348.88     1.0
2                  0        113931.57     1.0
35                 0         27822.99     1.0
41                 0        181297.65     1.0
43                 0        194365.76     1.0
…                …                …     …
9920               1           738.88     1.0
9947               1         73124.53     1.0
9956               0        117369.52     1.0
9991               0         69384.71     1.0
9997               1         42085.58     1.0

[809 rows x 10 columns]
```

[28]:
```
fran.Geography.value_counts()
```

[28]:
```
France    809
Name: Geography, dtype: int64
```

[29]:
```
fran.Exited.value_counts()
```

[29]:
```
1.0    809
Name: Exited, dtype: int64
```

0.0.19 18. Display what are the rows have 4,6,9, and 10 years in 'Tenure' column.

[30]:
```
data[data['Tenure'].isin([4,6,9,10])]
```

[30]:
```
     Geography  Gender  Age  Tenure    Balance  NumOfProducts  HasCrCard  \
7      Germany  Female   29       4  115046.74              4          1
8       France    Male   44       4  142051.07              2          0
10      France    Male   31       6  102016.72              2          0
12      France  Female   34      10       0.00              2          1
17       Spain  Female   24       9       0.00              2          1
…        …     …   …    …       …          …        …
9988    France    Male   30       4       0.00              2          1
9989     Spain    Male   28       4       0.00              2          1
9991    France  Female   53       4   88381.21              1          1
9996    France    Male   35      10   57369.61              1          1
9999    France  Female   28       4  130142.79              1          1
```

```
      IsActiveMember  EstimatedSalary  Exited
7                  0        119346.88     1.0
8                  1         74940.50     0.0
10                 0         80181.12     0.0
12                 0         26260.98     0.0
17                 1         14406.41     0.0
…                …              …      …
9988               0         49337.84     0.0
9989               1        179436.60     0.0
9991               0         69384.71     1.0
9996               1        101699.77     0.0
9999               0         38190.78     0.0

[3430 rows x 10 columns]
```

0.0.20 19. Group the 'Exited' colum value according to the colum 'Geography' and 'Gender'.

[31]:
```
data.groupby(['Exited','Geography','Gender']).sum()
```

[31]:
```
                            Age  Tenure       Balance  NumOfProducts  \
Exited Geography Gender
0.0    France    Female   66675    8912  1.052222e+08           2797
                 Male     89616   12104  1.482193e+08           3672
       Germany   Female   27688    3644  8.815290e+07           1141
                 Male     35279    4825  1.135910e+08           1478
       Spain     Female   32550    4359  4.855836e+07           1339
                 Male     45362    6159  7.423509e+07           1839
1.0    France    Female   20813    2261  3.107532e+07            694
                 Male     15705    1786  2.649879e+07            501
       Germany   Female   20083    2274  5.361389e+07            657
                 Male     16461    1801  4.436002e+07            527
       Spain     Female   10074    1081  1.663146e+07            369
                 Male      8159     842  1.331655e+07            256

                          HasCrCard  IsActiveMember  EstimatedSalary
Exited Geography Gender
0.0    France    Female        1253             987     1.772353e+08
                 Male          1714            1302     2.394005e+08
       Germany   Female         523             391     7.714630e+07
                 Male           687             557     9.573961e+07
       Spain     Female         612             485     8.467429e+07
                 Male           826             682     1.182979e+08
1.0    France    Female         321             171     4.750332e+07
                 Male           248             126     3.611784e+07
       Germany   Female         317             165     4.474824e+07
```

```
          Male         260            131     3.535253e+07
Spain     Female       157             77     2.484269e+07
          Male         121             64     1.795632e+07
```

0.0.21 20. Find group and Aggregate 'mean','count' in the 'Exited' colum value according to the colum 'Geography' and 'Gender'.

```
[32]: data[['Geography','Gender','Exited']].groupby(['Geography','Gender']).
      ↪agg(['mean','count'])
```

```
[32]:                     Exited
                        mean count
      Geography Gender
      France    Female  0.203457  2256
                Male    0.127273  2750
      Germany   Female  0.376471  1190
                Male    0.278751  1313
      Spain     Female  0.212511  1087
                Male    0.131503  1384
```

0.0.22 21. Find number of churned Customers in 'Exited' colum and mean of 'Balance' coloum according to the group of common 'Geography' value. In addtion rename index as Exited':'Number of churned customers', 'Balance':'Average Balance of Customers'.

```
[33]: chur = data[['Geography','Exited','Balance']].groupby('Geography')\
      .agg({'Exited':'sum', 'Balance':'mean'})
      chur
```

```
[33]:            Exited        Balance
      Geography
      France      809.0   62128.569872
      Germany     814.0  119743.446336
      Spain       413.0   61813.622181
```

```
[34]: chur.rename(columns={'Exited':'Number of churned customers', 'Balance':'Average␣
      ↪Balance of Customers'})
      chur
```

```
[34]:            Exited        Balance
      Geography
      France      809.0   62128.569872
      Germany     814.0  119743.446336
      Spain       413.0   61813.622181
```

0.0.23 22. Set a 'Geography' column as the index.

```
[35]: data.set_index('Geography')
```

```
[35]:           Gender  Age  Tenure    Balance  NumOfProducts  HasCrCard  \
Geography
France      Female   42       2       0.00              1          1
Spain       Female   41       1   83807.86              1          0
France      Female   42       8  159660.80              3          1
France      Female   39       1       0.00              2          0
Spain       Female   43       2  125510.82              1          1
…             …   …       …          …            …        …
France        Male   39       5       0.00              2          1
France        Male   35      10   57369.61              1          1
France      Female   36       7       0.00              1          0
Germany       Male   42       3   75075.31              2          1
France      Female   28       4  130142.79              1          1

           IsActiveMember  EstimatedSalary  Exited
Geography
France                  1        101348.88     1.0
Spain                   1        112542.58     0.0
France                  0        113931.57     1.0
France                  0         93826.63     0.0
Spain                   1         79084.10     0.0
…                     …              …     …
France                  0         96270.64     0.0
France                  1        101699.77     0.0
France                  1         42085.58     1.0
Germany                 0         92888.52     1.0
France                  0         38190.78     0.0

[10000 rows x 9 columns]
```

0.0.24 23. Insert a new column with the value of random integer between 0 and 9 with size 6.

```
[36]: new_c = np.random.randint(10, size=10000)  # i have got error in size=6 len is␣
      ↪10000 rows.
```

```
[37]: data['Group'] = new_c
      data
```

```
[37]:     Geography  Gender  Age  Tenure    Balance  NumOfProducts  HasCrCard  \
0        France  Female   42       2       0.00              1          1
1         Spain  Female   41       1   83807.86              1          0
2        France  Female   42       8  159660.80              3          1
3        France  Female   39       1       0.00              2          0
```

```
4         Spain  Female   43        2  125510.82             1           1
…          …      …  …     …        …           …         …
9995    France     Male   39        5       0.00             2           1
9996    France     Male   35       10   57369.61             1           1
9997    France   Female   36        7       0.00             1           0
9998   Germany     Male   42        3   75075.31             2           1
9999    France   Female   28        4  130142.79             1           1

      IsActiveMember  EstimatedSalary  Exited  Group
0                  1        101348.88     1.0      8
1                  1        112542.58     0.0      9
2                  0        113931.57     1.0      1
3                  0         93826.63     0.0      6
4                  1         79084.10     0.0      3
…               …              …     …      …
9995               0         96270.64     0.0      5
9996               1        101699.77     0.0      0
9997               1         42085.58     1.0      7
9998               0         92888.52     1.0      6
9999               0         38190.78     0.0      0

[10000 rows x 11 columns]
```

0.0.25 24. Change newely added colum position to 0.

[38]:
```python
data=data.drop(['Group'],axis=1)
```

[39]:
```python
data.insert(loc=0,column='Group',value=new_c)
data
```

[39]:
```
      Group Geography  Gender  Age  Tenure    Balance  NumOfProducts  \
0         8    France  Female   42       2       0.00              1
1         9     Spain  Female   41       1   83807.86              1
2         1    France  Female   42       8  159660.80              3
3         6    France  Female   39       1       0.00              2
4         3     Spain  Female   43       2  125510.82              1
…       …       …      …     …      …         …              …
9995      5    France    Male   39       5       0.00              2
9996      0    France    Male   35      10   57369.61              1
9997      7    France  Female   36       7       0.00              1
9998      6   Germany    Male   42       3   75075.31              2
9999      0    France  Female   28       4  130142.79              1

      HasCrCard  IsActiveMember  EstimatedSalary  Exited
0             1               1        101348.88     1.0
1             0               1        112542.58     0.0
2             1               0        113931.57     1.0
```

```
3             0               0         93826.63     0.0
4             1               1         79084.10     0.0
…          …             …             …      …
9995          1               0         96270.64     0.0
9996          1               1        101699.77     0.0
9997          0               1         42085.58     1.0
9998          1               0         92888.52     1.0
9999          1               0         38190.78     0.0

[10000 rows x 11 columns]
```

0.0.26 25. Create new dataframe 'newdf' from exsisting dataframe. You have to set the balance to 0 for customers who belong to a 'group' that is less than 6 in the 'newdf'.

[40]:
```python
data['newdf'] = data['Balance'].where(data['Group'] <= 6, 0)
data
```

[40]:
```
      Group Geography  Gender  Age  Tenure    Balance  NumOfProducts  \
0         8    France  Female   42       2       0.00              1
1         9     Spain  Female   41       1   83807.86              1
2         1    France  Female   42       8  159660.80              3
3         6    France  Female   39       1       0.00              2
4         3     Spain  Female   43       2  125510.82              1
…       …         …      …     …     …         …             …
9995      5    France    Male   39       5       0.00              2
9996      0    France    Male   35      10   57369.61              1
9997      7    France  Female   36       7       0.00              1
9998      6   Germany    Male   42       3   75075.31              2
9999      0    France  Female   28       4  130142.79              1

      HasCrCard  IsActiveMember  EstimatedSalary  Exited      newdf
0             1               1        101348.88     1.0       0.00
1             0               1        112542.58     0.0       0.00
2             1               0        113931.57     1.0  159660.80
3             0               0         93826.63     0.0       0.00
4             1               1         79084.10     0.0  125510.82
…          …             …             …      …        …
9995          1               0         96270.64     0.0       0.00
9996          1               1        101699.77     0.0   57369.61
9997          0               1         42085.58     1.0       0.00
9998          1               0         92888.52     1.0   75075.31
9999          1               0         38190.78     0.0  130142.79

[10000 rows x 12 columns]
```

0.0.27 26. Ranking the values of 'Balance' column. Let's create a column that ranks the customers according to their balances named as 'Rank'.

[41]:
```
data['Rank'] = data['Balance'].rank(method='first').astype('float')
data.Rank,data.Balance
```

[41]:
```
(0          1.0
 1       4351.0
 2       9392.0
 3          2.0
 4       7296.0
          …
 9995    3609.0
 9996    3733.0
 9997    3610.0
 9998    4036.0
 9999    7712.0
 Name: Rank, Length: 10000, dtype: float64,
 0            0.00
 1        83807.86
 2       159660.80
 3            0.00
 4       125510.82
           …
 9995         0.00
 9996     57369.61
 9997         0.00
 9998     75075.31
 9999    130142.79
 Name: Balance, Length: 10000, dtype: float64)
```

0.0.28 27. Find the number of unique values in the colum 'Geography'.

[42]:
```
data.Geography.nunique()
```

[42]:
```
3
```

0.0.29 28. Find the memory usage of Dataframe

[43]:
```
data.info(memory_usage='deep')
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 13 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Group            10000 non-null  int32
 1   Geography        9980 non-null   object
 2   Gender           10000 non-null  object
 3   Age              10000 non-null  int64
 4   Tenure           10000 non-null  int64
 5   Balance          9980 non-null   float64
 6   NumOfProducts    10000 non-null  int64
 7   HasCrCard        10000 non-null  int64
 8   IsActiveMember   10000 non-null  int64
 9   EstimatedSalary  10000 non-null  float64
 10  Exited           9980 non-null   float64
 11  newdf            9983 non-null   float64
 12  Rank             9980 non-null   float64
dtypes: float64(5), int32(1), int64(5), object(2)
memory usage: 2.0 MB
```

0.0.30 29. Categorize 'Geography' column values according to its data type.

[44]:
```
geo1=data.groupby(['Geography']).sum()
geo1
```

[44]:
```
           Group     Age  Tenure       Balance  NumOfProducts  HasCrCard  \
Geography
France     22687  192809   25063  3.110156e+08           7664       3536
Germany    11395   99511   12544  2.997178e+08           3803       1787
Spain      11378   96145   12441  1.527415e+08           3803       1716

           IsActiveMember  EstimatedSalary  Exited         newdf        Rank
Geography
France               2586     5.002570e+08   809.0  2.171376e+08  22015445.0
Germany              1244     2.529867e+08   814.0  2.069454e+08  17008583.0
Spain                1308     2.457712e+08   413.0  1.042581e+08  10781162.0
```

0.0.31 30. Replace values in a dataframe, The value as 'v1' have to replace where dataframe 'Group' column has value as '0'.

[45]:
```
data['Group'].replace(0,'v1')
```

[45]:
```
0        8
1        9
2        1
3        6
4        3
        ..
9995     5
9996    v1
9997     7
9998     6
9999    v1
```

```
Name: Group, Length: 10000, dtype: object
```

0.0.32 31. Adjust numver of decimal places in newdf as 3.

[46]:
```
data['newdf'] = data['newdf'].map("{:,.3f}".format)
data
```

[46]:
```
      Group Geography  Gender  Age  Tenure    Balance  NumOfProducts  \
0         8    France  Female   42       2       0.00              1
1         9     Spain  Female   41       1   83807.86              1
2         1    France  Female   42       8  159660.80              3
3         6    France  Female   39       1       0.00              2
4         3     Spain  Female   43       2  125510.82              1
…       …        …      …     …      …       …              …
9995      5    France    Male   39       5       0.00              2
9996      0    France    Male   35      10   57369.61              1
9997      7    France  Female   36       7       0.00              1
9998      6   Germany    Male   42       3   75075.31              2
9999      0    France  Female   28       4  130142.79              1

      HasCrCard  IsActiveMember  EstimatedSalary  Exited        newdf    Rank
0             1               1        101348.88     1.0        0.000     1.0
1             0               1        112542.58     0.0        0.000  4351.0
2             1               0        113931.57     1.0  159,660.800  9392.0
3             0               0         93826.63     0.0        0.000     2.0
4             1               1         79084.10     0.0  125,510.820  7296.0
…           …               …              …     …        …      …
9995          1               0         96270.64     0.0        0.000  3609.0
9996          1               1        101699.77     0.0   57,369.610  3733.0
9997          0               1         42085.58     1.0        0.000  3610.0
9998          1               0         92888.52     1.0   75,075.310  4036.0
9999          1               0         38190.78     0.0  130,142.790  7712.0

[10000 rows x 13 columns]
```

0.0.33 32. Filter the rows in which the customer name starts with 'J'.

[47]:
```
data_j=pd.read_csv('data.csv')
data_j[(data_j.Surname.str.startswith('J'))]
```

[47]:
```
      RowNumber  CustomerId   Surname  CreditScore Geography  Gender  Age  \
62           63    15702014  Jeffrey           555     Spain    Male   33
64           65    15592461   Jackson          603   Germany    Male   26
198         199    15656176   Jenkins          501    France    Male   57
201         202    15622911      Jude          759    France    Male   42
238         239    15794056  Johnston          668    France  Female   46
…           …          …        …            …          …     …   …
```

```
9674       9675   15578098   Jamieson          600    France    Male   31
9723       9724   15612832   Jamieson          526    France    Male   32
9842       9843   15746704    Jibunoh          638     Spain    Male   30
9946       9947   15618171      James          669    France  Female   33
9996       9997   15569892  Johnstone          516    France    Male   35

      Tenure    Balance  NumOfProducts  HasCrCard  IsActiveMember  \
62         1   56084.69              2          0               0
64         4  109166.37              1          1               1
198       10       0.00              2          1               1
201        4  105420.18              1          0               1
238        2       0.00              3          1               0
…        …        …            …        …             …
9674       8       0.00              2          1               1
9723       7  125540.05              1          0               0
9842       9  136808.53              2          1               1
9946       9       0.00              2          0               1
9996      10   57369.61              1          1               1

      EstimatedSalary  Exited
62          178798.13       0
64           92840.67       0
198          47847.19       0
201         121409.06       0
238          89048.46       1
…                 …     …
9674        121555.51       0
9723         86786.41       0
9842        106642.97       0
9946        107221.03       0
9996        101699.77       0

[163 rows x 14 columns]
```

0.0.34 33. Highlight the minimum values of each colum using style property.

```
[ ]:
```

```
[ ]:
```