In [1]:
```
#uploading in jupyter notebook
import pandas as pd
import numpy as np
```

In [2]:
```
pwd
```

Out[2]:
```
'C:\\Users\\DELL'
```

In [24]:
```
df = pd.read_csv('NationalNames.csv.crdownload')
```

In [25]:
```
df.head()
```

Out[25]:

	Id	Name	Year	Gender	Count
0	1	Mary	1880	F	7065.0
1	2	Anna	1880	F	2604.0
2	3	Emma	1880	F	2003.0
3	4	Elizabeth	1880	F	1939.0
4	5	Minnie	1880	F	1746.0

In [26]:
```
#Show the first and last 10 rows
df.info
```

Out[26]:
```
<bound method DataFrame.info of             Id       Name  Year Gender
Count
0            1       Mary  1880      F  7065.0
1            2       Anna  1880      F  2604.0
2            3       Emma  1880      F  2003.0
3            4  Elizabeth  1880      F  1939.0
4            5     Minnie  1880      F  1746.0
...        ...        ...   ...    ...     ...
932403  932404      Silpa  1983      F     6.0
932404  932405    Simmone  1983      F     6.0
932405  932406     Sinead  1983      F     6.0
932406  932407     Sioban  1983      F     6.0
932407  932408       Snow  1983    NaN     NaN

[932408 rows x 5 columns]>
```

In [27]:
```
#there more male or female names in the dataset
df['Gender'].value_counts()
```

Out[27]:
```
F    553010
M    379397
Name: Gender, dtype: int64
```

In [28]:
```
len(df[(df['Name'] == 'Yes') & (df['Gender'] == 'Female')])
```

Out[28]:
```
0
```

In [29]:
```
len(df['Gender'].value_counts())/len(df)*100
```

Out[29]:
```
0.0002144983741023243
```

In [12]:
```
df.columns
```

Out[12]:
```
Index(['Id', 'Name', 'Year', 'Gender', 'Count'], dtype='object')
```

In [30]:
```
#the total counts of each Name
df.groupby('Name').sum()
```

Out[30]:

Name	Id	Year	Count
Aage	127770	1915	7.0
Aagot	123994	1915	5.0
Aaisha	892274	1981	6.0
Aakash	1764335	3960	10.0
Aaliyah	6865111	15836	257.0
...	...	...	...
Zygmund	3202338	32652	168.0
Zygmunt	7375988	52092	363.0
Zylpha	1510688	26728	84.0
Zylphia	893923	9600	29.0
Zyndall	1155292	3919	14.0

45259 rows × 3 columns

In [14]:
```
df = pd.read_csv('titanic_data.csv')
```

In [33]:
```
import pandas as pd
```

In [51]:
```
#Create a dataframe named bank_client_df as shown below using dictionary

data = {
    'ClientID': [1, 2, 3, 4, 5],
    'Name': ['parnav', 'iti', 'prince', 'kevin', 'kasula'],
    'net worth': [28, 35, 22, 40, 32],
    'years': [5, 4, 5, 2, 6]
}
```

In [52]:
```
bank_client_df = pd.DataFrame(data)
```

In [53]:
```
print(bank_client_df)
```

```
   ClientID    Name  net worth  years
0         1  parnav         28      5
1         2     iti         35      4
2         3  prince         22      5
3         4   kevin         40      2
4         5  kasula         32      6
```

In [54]:
```
#Define a function that increases all clients networth (stocks) by a fixed
apply it on Net Worth column
function that increases all clients networth (stocks) by a fixed value of 1
apply it on Net Worth column
def increase_net_worth(net_worth):
    return net_worth * 1.10
```

In [55]:
```
bank_client_df['net worth'] = bank_client_df['net worth'].apply(increase_ne
```

In [56]:
```
print(bank_client_df)
```

```
   ClientID    Name  net worth  years
0         1  parnav       30.8      5
1         2     iti       38.5      4
2         3  prince       24.2      5
3         4   kevin       44.0      2
4         5  kasula       35.2      6
```

In [57]:
```
#total Networth

total_net_worth = bank_client_df['net worth'].sum()
print("Total net worth: $", total_net_worth)
```

```
Total net worth: $ 172.7
```

In [58]:
```
#Filter rows where Years with bank is greater than or equal to 5 years.

filtered_df = bank_client_df[bank_client_df['years'] >= 5]
print(filtered_df)
```

```
   ClientID    Name  net worth  years
0         1  parnav       30.8      5
2         3  prince       24.2      5
4         5  kasula       35.2      6
```

In [59]:
```
#Rename Yash into Rahul

bank_client_df['Name'] = bank_client_df['Name'].replace('kasula', 'kala pa
print(bank_client_df)
```

```
   ClientID         Name  net worth  years
0         1       parnav       30.8      5
1         2          iti       38.5      4
2         3       prince       24.2      5
3         4        kevin       44.0      2
4         5  kala papita       35.2      6
```

In [71]:
```
#Read titanic_data csv in jupyter notebook

df = pd.read_csv('titanic_data.csv')
```

In [72]:
```
import pandas as pd
```

In [73]:
```
df.head()
```

Out[73]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_ma
0	0	3	male	22.0	1	0	7.2500	S	Third	man	Tr
1	1	1	female	38.0	1	0	71.2833	C	First	woman	Fal
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	Fal
3	1	1	female	35.0	1	0	53.1000	S	First	woman	Fal
4	0	3	male	35.0	0	0	8.0500	S	Third	man	Tr

In [75]:
```
df.describe()
```

Out[75]:

	survived	pclass	age	sibsp	parch	fare
count	889.000000	889.000000	713.000000	889.000000	889.000000	889.000000
mean	0.384702	2.307087	29.698696	0.523060	0.382452	32.259059
std	0.486799	0.836367	14.536691	1.103729	0.806761	49.735870
min	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	0.000000	2.000000	20.000000	0.000000	0.000000	7.925000
50%	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

In [76]:
```
#total number of passengers from each class who survived?

passengers_df = pd.DataFrame(data)
```

In [79]:
```
df.groupby('survived').sum()
```

Out[79]:

survived	pclass	age	sibsp	parch	fare	adult_male	alone
0	1384	12955.50	303	181	12127.0741	447	373
1	667	8219.67	162	159	16551.2294	88	163

In [80]:
```
#the average fare for passengers from each embark_town?

df.groupby('embarked').mean()
```

Out[80]:

embarked	survived	pclass	age	sibsp	parch	fare	adult_male	alo
C	0.553571	1.886905	30.814769	0.386905	0.363095	59.954144	0.535714	0.5059
Q	0.394737	2.907895	28.089286	0.421053	0.171053	13.348741	0.473684	0.7500
S	0.337481	2.349922	29.444394	0.572317	0.413686	27.109647	0.636081	0.6096

In [81]:
```
#the average fare for male vs female passengers from each embark_town

df.groupby('sex').mean()
```

Out[81]:

sex	survived	pclass	age	sibsp	parch	fare	adult_male	alone
female	0.742038	2.159236	27.915709	0.694268	0.649682	44.479818	0.000000	0.401274
male	0.189565	2.387826	30.728252	0.429565	0.236522	25.585462	0.930435	0.713043

In [85]:
```
#Show the summary statistics of data
summary_stats = passengers_df.describe()
print(summary_stats)
```

```
       ClientID  net worth     years
count  5.000000   5.000000  5.000000
mean   3.000000  31.400000  4.400000
std    1.581139   6.841053  1.516575
min    1.000000  22.000000  2.000000
25%    2.000000  28.000000  4.000000
50%    3.000000  32.000000  5.000000
75%    4.000000  35.000000  5.000000
max    5.000000  40.000000  6.000000
```

In [ ]: