Baby Names: Exercise.

- Prepared by: Sagun Shakya.
- GITAM Institute of Science.

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import math
import os
```

DataSet URL:

- https://www.kaggle.com/kaggle/us-baby-names/version/1#StateNames.csv (https://www.kaggle.com/kaggle/us-baby-names/version/1#StateNames.csv)

In [51]:
```
os.chdir(r'D:\Sagun Shakya\Python\Data Sets')
```

In [3]:
```
baby = pd.read_csv('NationalNames.csv')
```

In [4]:
```
baby.head(10)
```

Out[4]:

	Id	Name	Year	Gender	Count
0	1	Mary	1880	F	7065
1	2	Anna	1880	F	2604
2	3	Emma	1880	F	2003
3	4	Elizabeth	1880	F	1939
4	5	Minnie	1880	F	1746
5	6	Margaret	1880	F	1578
6	7	Ida	1880	F	1472
7	8	Alice	1880	F	1414
8	9	Bertha	1880	F	1320
9	10	Sarah	1880	F	1288

In [5]:
```
#delete the column named Id.
del baby['Id']
```

In [6]:
```
baby.head()
```

Out[6]:

	Name	Year	Gender	Count
0	Mary	1880	F	7065
1	Anna	1880	F	2604
2	Emma	1880	F	2003
3	Elizabeth	1880	F	1939
4	Minnie	1880	F	1746

Males greater than females?

In [9]:
```
baby['Gender'].value_counts()
```

Out[9]:
```
F    1081683
M     743750
Name: Gender, dtype: int64
```

Group by names.

In [21]:
```
names_count = baby[['Name', 'Count']]

names_count.groupby('Name').sum().head(10)
```

Out[21]:

	Count
Name	
Aaban	72
Aabha	21
Aabid	5
Aabriella	10
Aadam	196
Aadan	112
Aadarsh	158
Aaden	3920
Aadesh	15
Aadhav	102

Sort by count.

In [22]:
```
names = names_count.groupby('Name').sum()

names.sort_values('Count', ascending = False).head(10)
```

Out[22]:

Name	Count
James	5129096
John	5106590
Robert	4816785
Michael	4330805
Mary	4130441
William	4071368
David	3590557
Joseph	2580687
Richard	2564867
Charles	2376700

Number of unique names.

In [23]:
```
len(names)
```

Out[23]: 93889

In [26]:
```
#Aliter
#SYNTAX: dataframe.column_name.unique()
uniques = baby.Name.unique()
len(uniques)
```

Out[26]: 93889

Name with the most occurences.

In [34]:
```
names.sort_values('Count', ascending = False).head(1)
```

Out[34]:

Name	Count
James	5129096

In [38]:
```
#Aliter
names[ names['Count'] == names['Count'].max() ]
```

Out[38]:

Name	Count
James	5129096

In [46]:
```
names_with_min_occ = names[ names['Count'] == names['Count'].min() ]

print(names_with_min_occ.shape)
print('\n')
print(names_with_min_occ.head(10))
```

```
(13393, 1)


         Count
Name
Aabid        5
Aadhyan      5
Aadian       5
Aadrian      5
Aadrit       5
Aafreen      5
Aagot        5
Aahron       5
Aaiyana      5
Aaja         5
```

Names with median number of occurences.

In [48]:
```
names[ names['Count'] == names['Count'].median() ]
```

Out[48]:

Name	Count
Abhijeet	45
Adaire	45
Adaleen	45
Adebola	45
Adream	45
Adwin	45
Ailia	45
Ajwa	45
Aleph	45
Aleshanee	45
Aleyiah	45
Aljandro	45
Allyzon	45
Alylah	45
Alyzon	45
Amary	45
Ambrey	45
Ameriyah	45
Amishi	45
Amiyha	45
Amoriah	45
Anachristina	45
Aneres	45
Anezka	45
Anglie	45
Annakaren	45
Anonda	45
Antanesha	45
Antravious	45
Antwanique	45
...	...
Treye	45
Trieste	45
Trinell	45
Tripton	45

Name	Count
Trulie	45
Tunesia	45
Tyquana	45
Tyres	45
Urmi	45
Vedad	45
Viviene	45
Waid	45
Wellman	45
Witold	45
Yaelis	45
Yagmur	45
Yanae	45
Yaricza	45
Yeili	45
Yurico	45
Zafiro	45
Zamyrah	45
Zarie	45
Zeki	45
Zelna	45
Zeni	45
Zhion	45
Zorka	45
Zyniyah	45
Zyrihanna	45

340 rows × 1 columns

Standard deviation of the names count.

In [49]:
```
names['Count'].std()
```

Out[49]:
```
55665.63350735194
```

Statistical Summary.

In [50]:
```
names.describe()
```

Out[50]:

	Count
count	9.388900e+04
mean	3.590787e+03
std	5.566563e+04
min	5.000000e+00
25%	1.100000e+01
50%	4.500000e+01
75%	2.370000e+02
max	5.129096e+06

The End