US - Baby Names

Introduction:

We are going to use a subset of US Baby Names from Kaggle.
In the file it will be names from 2004 until 2014

Step 1. Import the necessary libraries

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

Step 2. Import the dataset from this address.

Step 3. Assign it to a variable called baby_names.

In [19]:
```
bn = pd.read_csv("https://raw.githubusercontent.com/guipsamora/pandas_exercises/master/06_Stats/US_Baby_Names/US_Baby_Names_righ
bn
```

Out[19]:

	Unnamed: 0	Id	Name	Year	Gender	State	Count
0	11349	11350	Emma	2004	F	AK	62
1	11350	11351	Madison	2004	F	AK	48
2	11351	11352	Hannah	2004	F	AK	46
3	11352	11353	Grace	2004	F	AK	44
4	11353	11354	Emily	2004	F	AK	41
...	...	...	...	...	...	...	...
1016390	5647421	5647422	Seth	2014	M	WY	5
1016391	5647422	5647423	Spencer	2014	M	WY	5
1016392	5647423	5647424	Tyce	2014	M	WY	5
1016393	5647424	5647425	Victor	2014	M	WY	5
1016394	5647425	5647426	Waylon	2014	M	WY	5

1016395 rows × 7 columns

Step 4. See the first 10 entries

In [20]:
```
bn.head(10)
```

Out[20]:

	Unnamed: 0	Id	Name	Year	Gender	State	Count
0	11349	11350	Emma	2004	F	AK	62
1	11350	11351	Madison	2004	F	AK	48
2	11351	11352	Hannah	2004	F	AK	46
3	11352	11353	Grace	2004	F	AK	44
4	11353	11354	Emily	2004	F	AK	41
5	11354	11355	Abigail	2004	F	AK	37
6	11355	11356	Olivia	2004	F	AK	33
7	11356	11357	Isabella	2004	F	AK	30
8	11357	11358	Alyssa	2004	F	AK	29
9	11358	11359	Sophia	2004	F	AK	28

Step 5. Delete the column 'Unnamed: 0' and 'Id'

In [21]:
```
bn = bn.drop(["Unnamed: 0" ,"Id" ], axis =1)
bn
```

Out[21]:

	Name	Year	Gender	State	Count
0	Emma	2004	F	AK	62
1	Madison	2004	F	AK	48
2	Hannah	2004	F	AK	46
3	Grace	2004	F	AK	44
4	Emily	2004	F	AK	41
...	...	...	...	...	...
1016390	Seth	2014	M	WY	5
1016391	Spencer	2014	M	WY	5
1016392	Tyce	2014	M	WY	5
1016393	Victor	2014	M	WY	5
1016394	Waylon	2014	M	WY	5

1016395 rows × 5 columns

Step 6. Is there more male or female names in the dataset?

In [22]:
```
bn.Gender.value_counts()
```

Out[22]:
```
F    558846
M    457549
Name: Gender, dtype: int64
```

In [23]:
```
bn.Gender.describe()
```

Out[23]:
```
count      1016395
unique           2
top              F
freq        558846
Name: Gender, dtype: object
```

Step 7. Group the dataset by name and assign to names

In [26]:
```
n = bn.groupby("Name")
n = bn.drop('Year', axis = 1).groupby('Name').sum()
n
```

Out[26]:

Name	Count
Aaban	12
Aadan	23
Aadarsh	5
Aaden	3426
Aadhav	6
...	...
Zyra	42
Zyrah	11
Zyren	6
Zyria	59
Zyriah	58

17632 rows × 1 columns

Step 8. How many different names exist in the dataset?

In [28]:
```
n.index.nunique()
```

Out[28]: 17632

In [29]:
```
len(n)
```

Out[29]: 17632

Step 9. What is the name with most occurrences?

In [32]:
```
n.groupby('Name').Count.sum().sort_values(ascending = False)
```

Out[32]:
```
Name
Jacob          242874
Emma           214852
Michael        214405
Ethan          209277
Isabella       204798
                  ...
Eniola              5
Atlantis            5
Marci               5
Simarpreet          5
Nita                5
Name: Count, Length: 17632, dtype: int64
```

Step 10. How many different names have the least occurrences?

In [35]:
```
(n.Count == n.Count.min()).sum()
```

Out[35]: 2578

Step 11. What is the median name occurrence?

In [41]:
```
(n.Count == n.Count.median()).sum()
```

Out[41]: 66

Step 12. What is the standard deviation of names?

In [38]:
```
(n.Count == n.Count.median()).sum()
```

Out[38]:
```
Count    11006.069468
dtype: float64
```

In [45]:
```
n.std()
```

Out[45]:
```
Count    11006.069468
dtype: float64
```

Step 13. Get a summary with the mean, min, max, std and quartiles.

In [46]:
```
n.describe()
```

Out[46]:

	Count
count	17632.000000
mean	2008.932169
std	11006.069468
min	5.000000
25%	11.000000
50%	49.000000
75%	337.000000
max	242874.000000

In [ ]: