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
```

Step 2. Import the dataset from this address.

Step 3. Assign it to a variable called baby_names.

In [2]:
```
baby_names = pd.read_csv('https://raw.githubusercontent.com/guipsamora/pandas_exercises/master/06_Stats/US_Baby_Names/US_Baby_Names_right.csv')
baby_names.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1016395 entries, 0 to 1016394
Data columns (total 7 columns):
 #   Column       Non-Null Count    Dtype 
---  ------       --------------    ----- 
 0   Unnamed: 0   1016395 non-null  int64 
 1   Id           1016395 non-null  int64 
 2   Name         1016395 non-null  object
 3   Year         1016395 non-null  int64 
 4   Gender       1016395 non-null  object
 5   State        1016395 non-null  object
 6   Count        1016395 non-null  int64 
dtypes: int64(4), object(3)
memory usage: 54.3+ MB
```

Step 4. See the first 10 entries

In [3]:
```
baby_names.head(10)
```

Out[3]:

	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 [4]:
```
baby_names.drop(["Unnamed: 0","Id"], axis =1, inplace=True)
baby_names
```

Out[4]:

	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 [5]:
```
baby_names.groupby("Gender").count()
```

Out[5]:

	Name	Year	State	Count
Gender				
F	558846	558846	558846	558846
M	457549	457549	457549	457549

In [6]:
```
baby_names['Gender'].value_counts()
```

Out[6]:
```
F    558846
M    457549
Name: Gender, dtype: int64
```

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

In [7]:
```
baby_names.groupby("Name").sum()
```

Out[7]:

	Year	Count
Name		
Aaban	4027	12
Aadan	8039	23
Aadarsh	2009	5
Aaden	393963	3426
Aadhav	2014	6
...	...	...
Zyra	14085	42
Zyrah	4024	11
Zyren	2013	6
Zyria	20089	59
Zyriah	18087	58

17632 rows × 2 columns

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

In [8]:
```
baby_names["Name"].nunique()
```

Out[8]:
```
17632
```

Step 9. What is the name with most occurrences?

In [9]:
```
baby_names["Name"].value_counts().sort_values(ascending=False)
```

Out[9]:
```
Riley       1112
Avery       1080
Jordan      1073
Peyton      1064
Hayden      1049
            ... 
Man            1
Cordale        1
Kenson         1
Lofton         1
Augustas       1
Name: Name, Length: 17632, dtype: int64
```

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

In [10]:
```
baby_names["Name"].value_counts().sort_values(ascending=True)
```

Out[10]:
```
Augustas       1
Lofton         1
Kenson         1
Cordale        1
Man            1
            ... 
Hayden      1049
Peyton      1064
Jordan      1073
Avery       1080
Riley       1112
Name: Name, Length: 17632, dtype: int64
```

Step 11. What is the median name occurrence?

In [11]:
```
baby_names[baby_names.Count == baby_names.Count.median()]
```

Out[11]:

	Name	Year	Gender	State	Count
71	Makayla	2004	F	AK	11
72	Maria	2004	F	AK	11
73	Mary	2004	F	AK	11
74	Michelle	2004	F	AK	11
259	Alexandra	2005	F	AK	11
...	...	...	...	...	...
1016276	Christopher	2014	M	WY	11
1016277	Corbin	2014	M	WY	11
1016278	Gavin	2014	M	WY	11
1016279	Greyson	2014	M	WY	11
1016280	Isaiah	2014	M	WY	11

36199 rows × 5 columns

Step 12. What is the standard deviation of names?

In [12]:
```
baby_names["Count"].std()
```

Out[12]:
```
97.39734648625934
```

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

In [13]:
```
baby_names.describe()
```

Out[13]:

	Year	Count
count	1.016395e+06	1.016395e+06
mean	2.009053e+03	3.485012e+01
std	3.138293e+00	9.739735e+01
min	2.004000e+03	5.000000e+00
25%	2.006000e+03	7.000000e+00
50%	2.009000e+03	1.100000e+01
75%	2.012000e+03	2.600000e+01
max	2.014000e+03	4.167000e+03