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

```
import pandas as pd
```

Step 2. Import the dataset from this address.

```
baby_names = pd.read_csv('us_baby.tsv', sep=',')
```

Step 3. Assign it to a variable called baby_names.

```
baby_names.head()
```

	Unnamed: 0	Id	Name	Year	Gender	State	Count
0	11349	11350	Emma	2004.0	F	AK	62.0
1	11350	11351	Madison	2004.0	F	AK	48.0
2	11351	11352	Hannah	2004.0	F	AK	46.0
3	11352	11353	Grace	2004.0	F	AK	44.0
4	11353	11354	Emily	2004.0	F	AK	41.0

Step 4. See the first 10 entries

```
print(baby_names.head(10))
```

```
   Unnamed: 0     Id      Name    Year Gender State  Count
0       11349  11350      Emma  2004.0      F    AK   62.0
1       11350  11351   Madison  2004.0      F    AK   48.0
2       11351  11352    Hannah  2004.0      F    AK   46.0
3       11352  11353     Grace  2004.0      F    AK   44.0
4       11353  11354     Emily  2004.0      F    AK   41.0
5       11354  11355   Abigail  2004.0      F    AK   37.0
6       11355  11356    Olivia  2004.0      F    AK   33.0
7       11356  11357  Isabella  2004.0      F    AK   30.0
8       11357  11358    Alyssa  2004.0      F    AK   29.0
9       11358  11359    Sophia  2004.0      F    AK   28.0
```

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

```
baby_names = baby_names.drop(columns=[col for col in ['Unnamed: 0', 'Id'] if col in baby_names.columns])
```

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

```
gender_counts = baby_names['Gender'].value_counts()
print("Gender counts:\n", gender_counts)
```

```
Gender counts:
 Gender
F    527809
M    426379
Name: count, dtype: int64
```

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

```
names = baby_names.groupby('Name').agg({'Count': 'sum'})
```

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

```
num_unique_names = names.shape[0]
print(f"Different names: {num_unique_names}")
```

```
Different names: 17604
```

Step 9. What is the name with most occurrences?

```
most_common_name = names['Count'].idxmax()
most_common_count = names['Count'].max()
print(f"Most common name: {most_common_name} ({most_common_count} occurrences)")
```

```
Most common name: Jacob (230414.0 occurrences)
```

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

```
least_common_count = names['Count'].min()
least_common_names = names[names['Count'] == least_common_count]
print(f"Number of names with least occurrences ({least_common_count}): {least_common_names.shape[0]}")
```

```
Number of names with least occurrences (5.0): 2567
```

Step 11. What is the median name occurrence?

```
median_occurrence = names['Count'].median()
print(f"Median name occurrence: {median_occurrence}")
```

```
Median name occurrence: 48.0
```

Step 12. What is the standard deviation of names?

```
std_occurrence = names['Count'].std()
print(f"Standard deviation of name occurrences: {std_occurrence}")
```

```
Standard deviation of name occurrences: 10461.874438928102
```

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

```
summary = names['Count'].describe()
print("Summary statistics:\n", summary)
```

```
Summary statistics:
 count     17604.000000
mean       1916.914792
std       10461.874439
min           5.000000
25%          11.000000
50%          48.000000
75%         332.000000
max      230414.000000
Name: Count, dtype: float64
```