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

In [5]:
```
us_babies = pd.read_csv("us_baby_names.csv")
```

In [4]:
```
us_babies.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1825433 entries, 0 to 1825432
Data columns (total 5 columns):
 #   Column  Dtype 
---  ------  ----- 
 0   Id      int64 
 1   Name    object
 2   Year    int64 
 3   Gender  object
 4   Count   int64 
dtypes: int64(3), object(2)
memory usage: 69.6+ MB
```

In [6]:
```
us_babies.describe()
```

Out[6]:

	Id	Year	Count
count	1.825433e+06	1.825433e+06	1.825433e+06
mean	9.127170e+05	1.972620e+03	1.846879e+02
std	5.269573e+05	3.352891e+01	1.566711e+03
min	1.000000e+00	1.880000e+03	5.000000e+00
25%	4.563590e+05	1.949000e+03	7.000000e+00
50%	9.127170e+05	1.982000e+03	1.200000e+01
75%	1.369075e+06	2.001000e+03	3.200000e+01
max	1.825433e+06	2.014000e+03	9.968000e+04

In [7]:
```
us_babies
```

Out[7]:

	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
...	...	...	...	...	...
1825428	1825429	Zykeem	2014	M	5
1825429	1825430	Zymeer	2014	M	5
1825430	1825431	Zymiere	2014	M	5
1825431	1825432	Zyran	2014	M	5
1825432	1825433	Zyrin	2014	M	5

1825433 rows × 5 columns

Data Manipulation

Q. What were the 5 most popular baby names in 2014 in US? To answer this, we have 3 data manipulation steps:

1. Slicing out the rows for 2014
2. Sorting rows in descending order by count
3. Retrieving the first five rows.

We are now checking the year column in the dataset.

In [8]:
```
us_babies['Year']
```

Out[8]:
```
0          1880
1          1880
2          1880
3          1880
4          1880
           ...
1825428    2014
1825429    2014
1825430    2014
1825431    2014
1825432    2014
Name: Year, Length: 1825433, dtype: int64
```

The year ranges from 1880 to 2014. Now, we need to extract only the year 2014.

In [9]:
```
us_babies['Year']==2014
```

Out[9]:
```
0          False
1          False
2          False
3          False
4          False
           ...
1825428     True
1825429     True
1825430     True
1825431     True
1825432     True
Name: Year, Length: 1825433, dtype: bool
```

The data with the false value is not 2014 and the data with true value is 2014. Dropping the false values and keeping the true values to retrieve 2014 data.

In [10]:
```
us_babies_2014 = us_babies.loc[us_babies['Year']==2014, :] #the data with 201
```

In [11]:
```
us_babies_2014
```

Out[11]:

	Id	Name	Year	Gender	Count
1792389	1792390	Emma	2014	F	20799
1792390	1792391	Olivia	2014	F	19674
1792391	1792392	Sophia	2014	F	18490
1792392	1792393	Isabella	2014	F	16950
1792393	1792394	Ava	2014	F	15586
...	...	...	...	...	...
1825428	1825429	Zykeem	2014	M	5
1825429	1825430	Zymeer	2014	M	5
1825430	1825431	Zymiere	2014	M	5
1825431	1825432	Zyran	2014	M	5
1825432	1825433	Zyrin	2014	M	5

33044 rows × 5 columns

Only the data with year 2014 is retrieved. Now, second step is to sort the rows in descending order by count.

In [13]:
```
sorted_us_2014 = us_babies_2014.sort_values('Count',ascending = False)
```

In [14]:
```
sorted_us_2014
```

Out[14]:

	Id	Name	Year	Gender	Count
1792389	1792390	Emma	2014	F	20799
1792390	1792391	Olivia	2014	F	19674
1811456	1811457	Noah	2014	M	19144
1792391	1792392	Sophia	2014	F	18490
1811457	1811458	Liam	2014	M	18342
...	...	...	...	...	...
1810561	1810562	Melba	2014	F	5
1810560	1810561	Melaya	2014	F	5
1810559	1810560	Mel	2014	F	5
1810558	1810559	Mekhi	2014	F	5
1825432	1825433	Zyrin	2014	M	5

33044 rows × 5 columns

The datas are sorted now in descending order by count. Now, heading to step 3, which is tot retrieve the first five rows.

In [15]:
```
sorted_us_2014.head(5)
```

Out[15]:

	Id	Name	Year	Gender	Count
1792389	1792390	Emma	2014	F	20799
1792390	1792391	Olivia	2014	F	19674
1811456	1811457	Noah	2014	M	19144
1792391	1792392	Sophia	2014	F	18490
1811457	1811458	Liam	2014	M	18342

Result: The most popular US baby names are Emma, Olivia, Noah, Sophia, and Liam.