

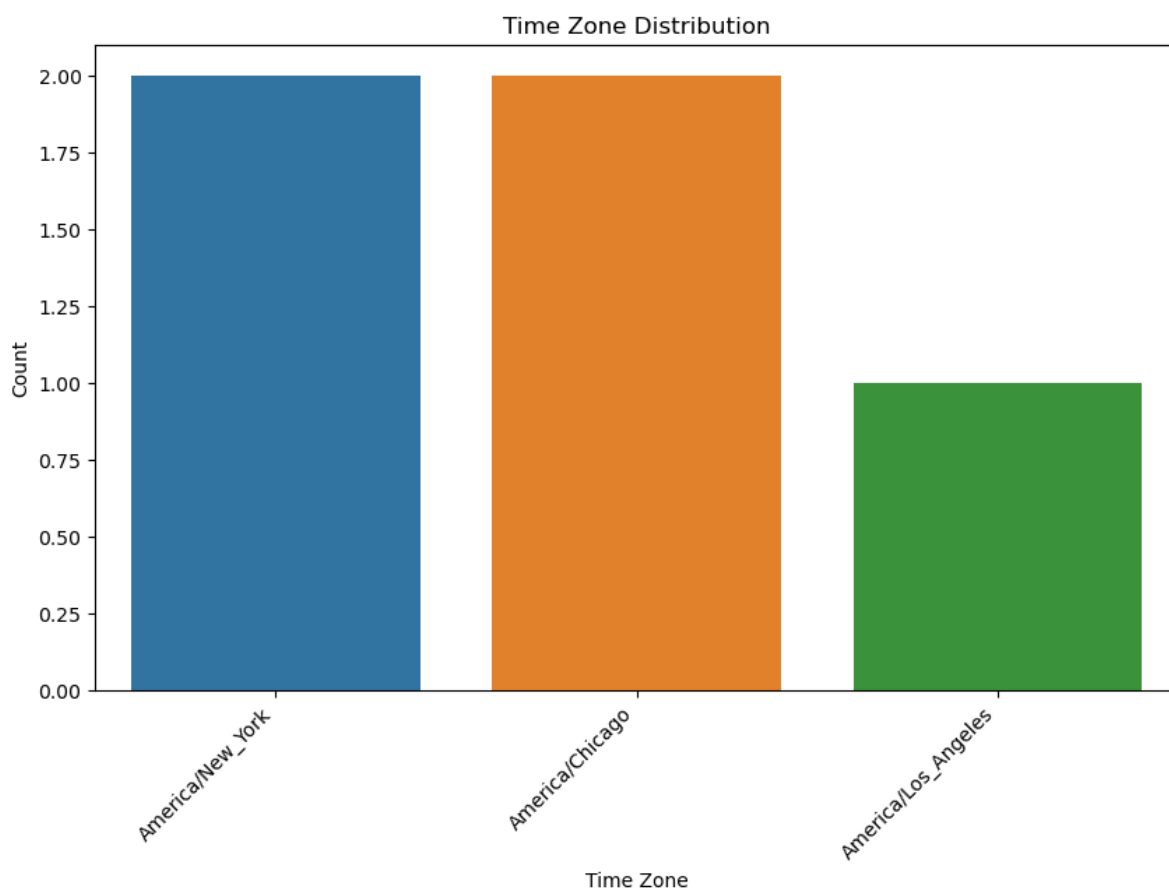
```
In [10]: # import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Sample data for illustration
data = {
    'tz': ['America/New_York', 'America/Chicago', 'America/New_York', 'America/Los_
    'count': [10, 15, 20, 25, 30]
}

# Create a pandas DataFrame from the sample data
df = pd.DataFrame(data)
# Display the first few rows of the DataFrame
print("Sample Data:")
print(df)
# Analyze time zone distribution
plt.figure(figsize=(10, 6))
sns.countplot(data=df, x='tz')
plt.title('Time Zone Distribution')
plt.xlabel('Time Zone')
plt.ylabel('Count')
plt.xticks(rotation=45, ha='right')
plt.show()
```

Sample Data:

	tz	count
0	America/New_York	10
1	America/Chicago	15
2	America/New_York	20
3	America/Los_Angeles	25
4	America/Chicago	30



```
In [6]: import pandas as pd
import numpy as np

# Create a movies dataset
```

```
movies_data = {
    'MovieID': range(1, 11),
    'Title': [f'Movie {i}' for i in range(1, 11)],
    'Genres': ['Genre A', 'Genre B', 'Genre C', 'Genre A', 'Genre B',
               'Genre C', 'Genre A', 'Genre B', 'Genre C', 'Genre A']
}

movies = pd.DataFrame(movies_data)

# Create a ratings dataset
np.random.seed(42)
ratings_data = {
    'UserID': np.random.randint(1, 101, size=1000),
    'MovieID': np.random.randint(1, 11, size=1000),
    'Rating': np.random.randint(1, 6, size=1000),
    'Timestamp': pd.to_datetime(np.random.randint(0, 2_000_000_000, size=1000), unit='s')
}

ratings = pd.DataFrame(ratings_data)

# Display sample data
print("Movies Data:")
print(movies)

print("\nRatings Data:")
print(ratings)

# Calculate rating disagreement
average_rating = ratings.groupby('MovieID')['Rating'].mean()
rating_disagreement = ratings.groupby('MovieID')['Rating'].std()

# Combine information into a DataFrame
movie_stats = pd.DataFrame({'AverageRating': average_rating, 'RatingDisagreement': rating_disagreement})

# Display movie stats
print("\nMovie Statistics:")
print(movie_stats.head())
```

Movies Data:

	MovieID	Title	Genres
0	1	Movie 1	Genre A
1	2	Movie 2	Genre B
2	3	Movie 3	Genre C
3	4	Movie 4	Genre A
4	5	Movie 5	Genre B
5	6	Movie 6	Genre C
6	7	Movie 7	Genre A
7	8	Movie 8	Genre B
8	9	Movie 9	Genre C
9	10	Movie 10	Genre A

Ratings Data:

	UserID	MovieID	Rating	Timestamp
0	52	2	5	1981-04-09 08:24:02
1	93	8	5	2006-06-12 06:00:42
2	15	8	2	2009-05-10 06:19:25
3	72	1	1	2018-05-30 17:58:09
4	61	3	2	1979-12-26 05:10:44
..
995	10	2	5	1992-07-23 11:11:25
996	67	5	1	2019-07-15 15:04:37
997	18	1	5	2030-08-21 23:53:00
998	100	7	1	1998-06-21 15:58:35
999	86	8	4	2024-02-15 12:22:04

[1000 rows x 4 columns]

Movie Statistics:

	AverageRating	RatingDisagreement
MovieID		
1	2.981132	1.479904
2	3.283019	1.336340
3	3.275510	1.405296
4	3.054348	1.303960
5	2.990099	1.493285

```
In [7]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Generate a sample baby names dataset
np.random.seed(42)
years = np.random.choice(range(2000, 2023), 1000)
names = np.random.choice(['Emma', 'Liam', 'Olivia', 'Noah', 'Ava', 'Sophia'], 1000)
counts = np.random.randint(10, 1000, 1000)

baby_names = pd.DataFrame({'Year': years, 'Name': names, 'Count': counts})

# Display the first few rows of the sample dataset
print("Sample Dataset:")
print(baby_names.head())

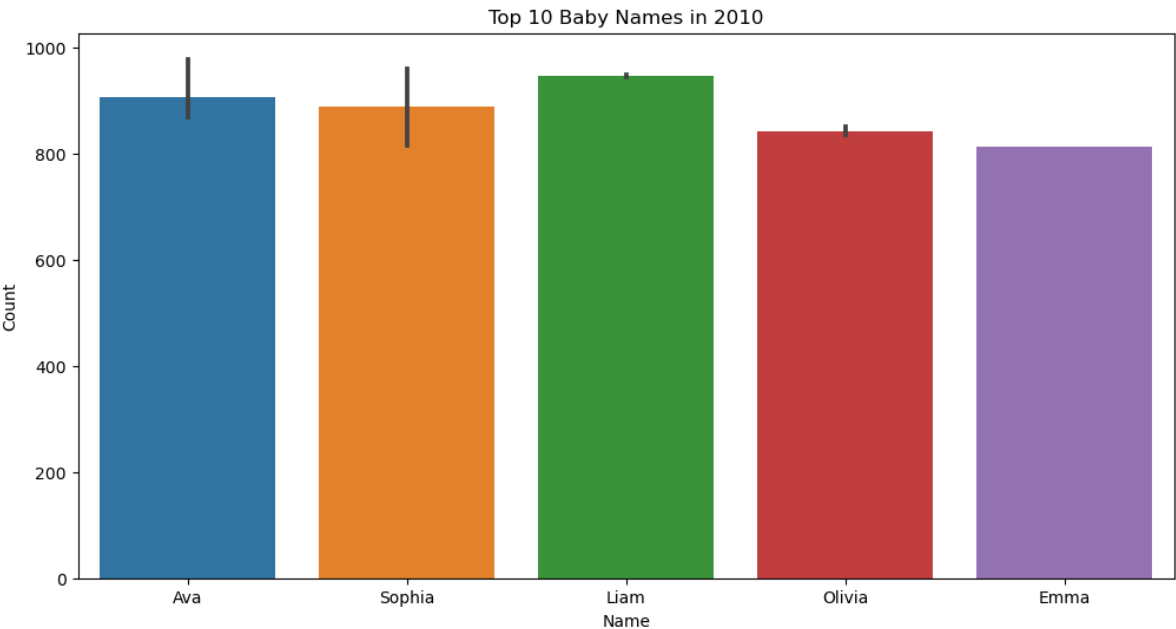
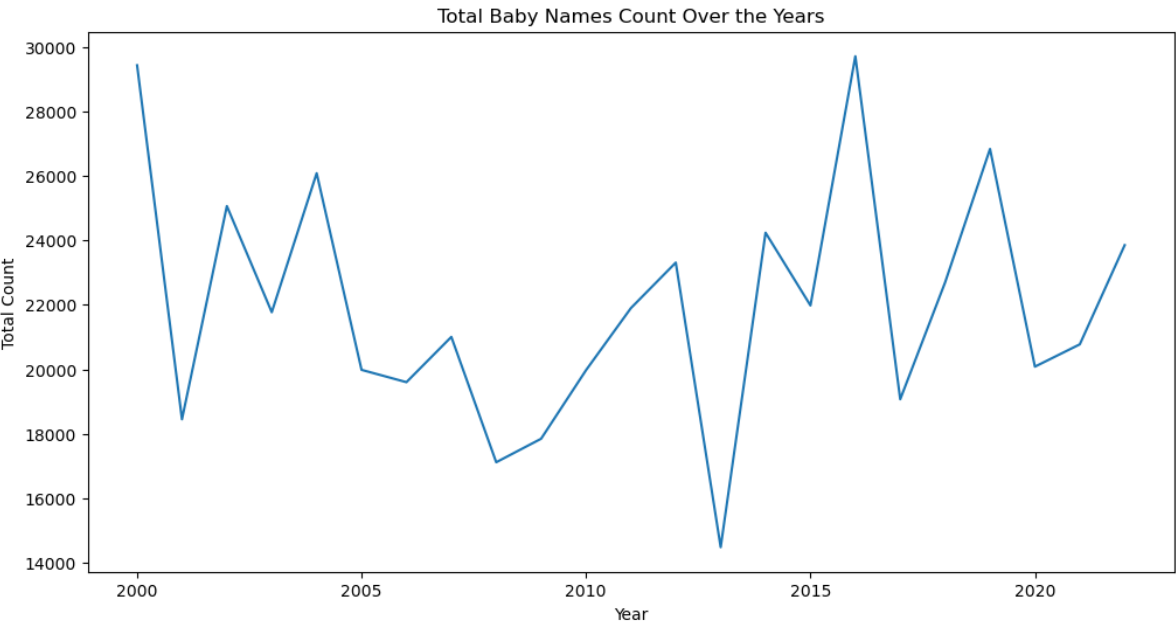
# Analyze naming trends over the years
plt.figure(figsize=(12, 6))
sns.lineplot(data=baby_names, x='Year', y='Count', estimator='sum', ci=None)
plt.title('Total Baby Names Count Over the Years')
plt.xlabel('Year')
plt.ylabel('Total Count')
plt.show()

# Analyze popular names for a specific year
```

```
selected_year = 2010
popular_names = baby_names[baby_names['Year'] == selected_year].nlargest(10, 'Count')
plt.figure(figsize=(12, 6))
sns.barplot(data=popular_names, x='Name', y='Count')
plt.title(f'Top 10 Baby Names in {selected_year}')
plt.xlabel('Name')
plt.ylabel('Count')
plt.show()
```

Sample Dataset:

	Year	Name	Count
0	2006	Noah	900
1	2019	Olivia	866
2	2014	Olivia	37
3	2010	Emma	75
4	2007	Sophia	172



In []: