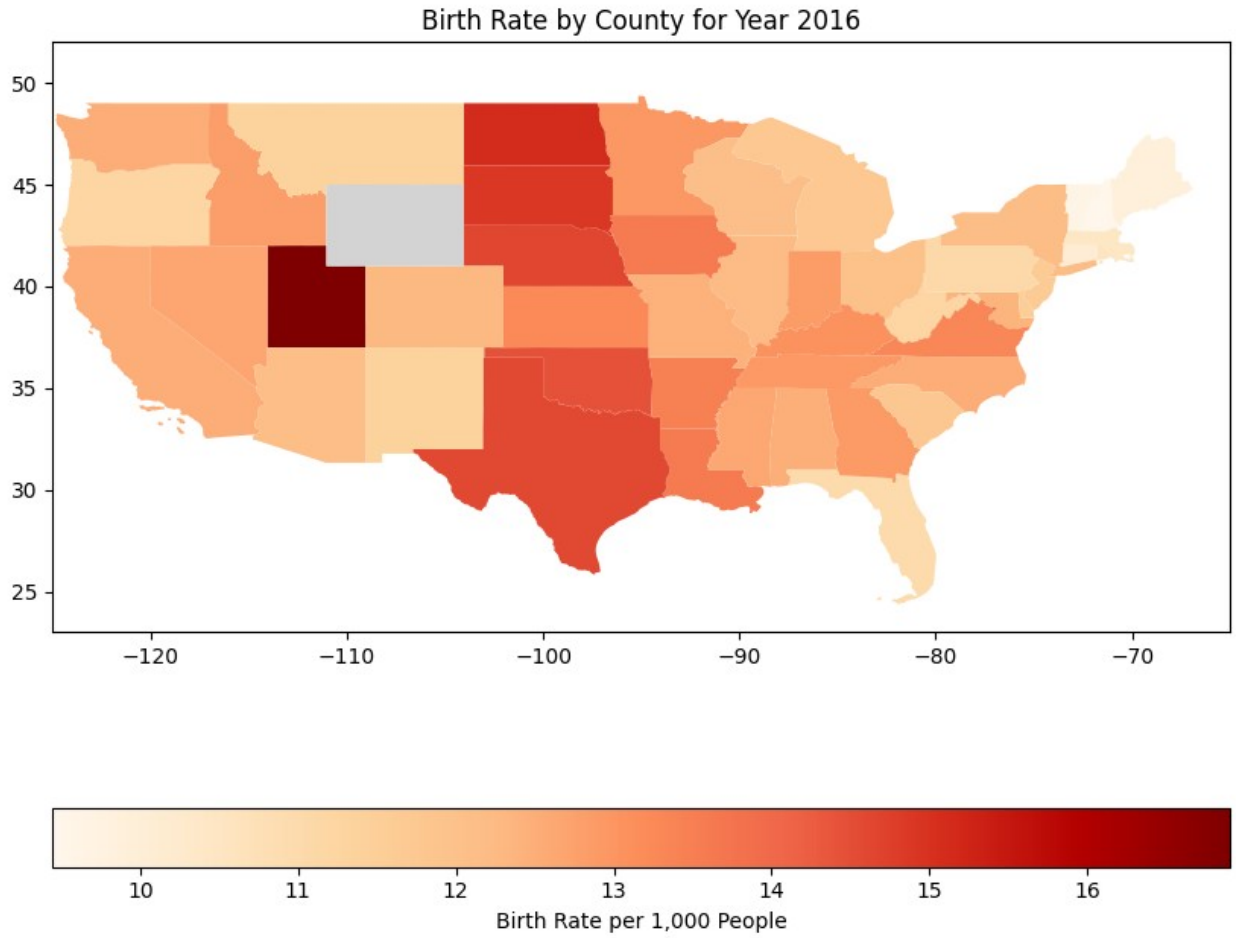
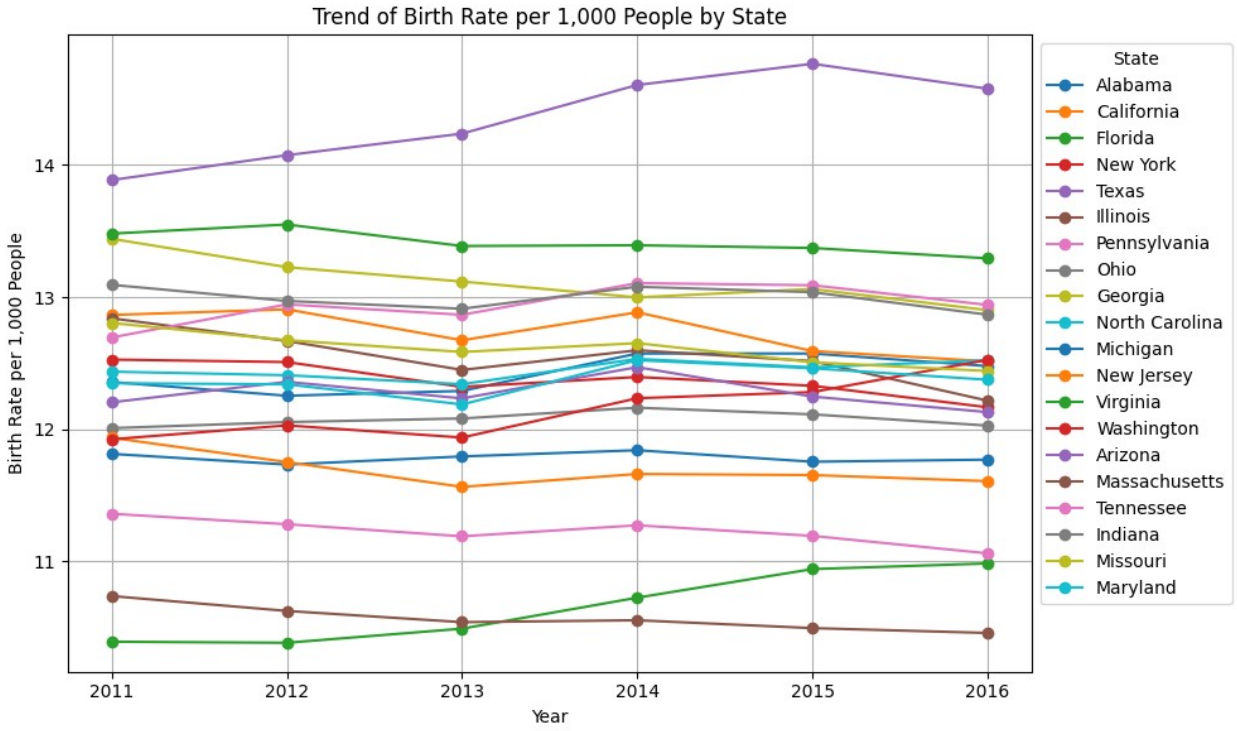
```
import pandas as pd

import geopandas as gpd
import matplotlib.pyplot as plt

birth  = pd.read_csv("birth_rate_file.csv")

df = birth
df['GEOID'] = df['FIPS'].apply(lambda x: str(x).zfill(5)[:2])
specific_year = 2016
df = df[df['year'] == specific_year]
df = df.groupby('GEOID').sum()

# Step 2: Calculate the birth rate per 1,000 people
df['Birth Rate'] = (df['Births'] / df['Population']) * 1000


gdf = gpd.read_file('tl_2016_us_state/tl_2016_us_state.shp')




merged_gdf = gdf.merge(df, on='GEOID', how='left')

# Step 4: Plotting
fig, ax = plt.subplots(1, figsize=(10, 10))
merged_gdf.plot(column='Birth Rate', ax=ax, legend=True, cmap='OrRd',
                missing_kwds={'color': 'lightgrey'},
                legend_kwds={'label': "Birth Rate per 1,000 People",
                             'orientation': "horizontal"})
plt.title('Birth Rate by County for Year {}'.format(specific_year))
# Setting the x and y-axis limits
plt.xlim(-125, -65)
plt.ylim(23, 52)

plt.show()
```




```
major_state_fips = ['01', '06', '12', '36', '48', '17', '42', '39',
'13', '37', '26', '34', '51', '53', '04', '25', '47', '18', '29',
'24']
state_names = ['Alabama', 'California', 'Florida', 'New York',
'Texas', 'Illinois', 'Pennsylvania', 'Ohio', 'Georgia', 'North
Carolina', 'Michigan', 'New Jersey', 'Virginia', 'Washington',
'Arizona', 'Massachusetts', 'Tennessee', 'Indiana', 'Missouri',
'Maryland']
state_dict = dict(zip(major_state_fips, state_names))

df = birth

df['GEOID'] = df['FIPS'].apply(lambda x: str(x).zfill(5)[:2])


filtered_birth = df[df['GEOID'].isin(major_state_fips)]

filtered_birth= filtered_birth.reset_index()
grouped_df = filtered_birth.groupby(['GEOID', 'year']).sum()
```

```
# Calculate the birth rate per 1,000 people
grouped_df['Birth Rate'] = (grouped_df['Births'] /
grouped_df['Population']) * 1000

# Reset index to use columns easily
grouped_df = grouped_df.reset_index()

grouped_df = grouped_df[grouped_df['year'] <= 2016]
# Map FIPS code to state names for the plot
grouped_df['State Name'] = grouped_df['GEOID'].map(state_dict)

# Plotting the trend of birth rates over years for each major state
plt.figure(figsize=(10, 6))
for state in state_names:
    subset = grouped_df[grouped_df['State Name'] == state]
    if not subset.empty:
        plt.plot(subset['year'], subset['Birth Rate'], marker='o',
label=state)

plt.title('Trend of Birth Rate per 1,000 People by State')
plt.xlabel('Year')
plt.ylabel('Birth Rate per 1,000 People')
plt.legend(title='State', loc='upper left', bbox_to_anchor=(1, 1))  #
Adjust legend outside of plot
plt.grid(True)
plt.tight_layout()
plt.show()
```

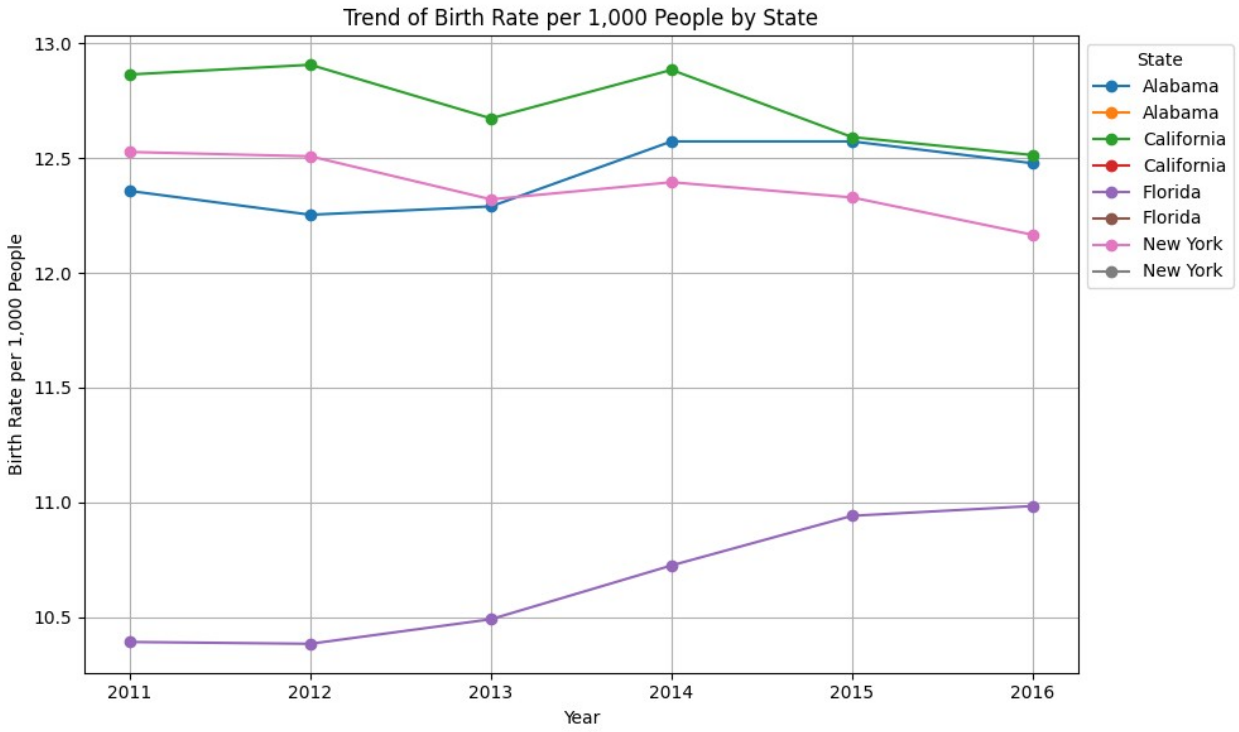


```
df = birth
df['GEOID'] = df['FIPS'].apply(lambda x: str(x).zfill(5)[:2])

df1 = df[df['year'] == 2016]
df2 = df[df['year'] == 2017]
df3 = df[df['year'] == 2018]

len(df1['GEOID'].unique())
len(df2['GEOID'].unique())
len(df3['GEOID'].unique())

sum(df1['GEOID'].unique() == df2['GEOID'].unique())
```

```
50
```

```
csv_data = pd.read_csv('melted_state_data.csv')


csv_data['Birth Rate'] = csv_data['Birth Rate'] * 1000
major_state_fips = ['01', '06', '12', '36']
state_names = ['Alabama', 'California', 'Florida', 'New York']
state_dict = dict(zip(major_state_fips, state_names))

df = birth

df['GEOID'] = df['FIPS'].apply(lambda x: str(x).zfill(5)[:2])


filtered_birth = df[df['GEOID'].isin(major_state_fips)]

filtered_birth= filtered_birth.reset_index()
grouped_df = filtered_birth.groupby(['GEOID', 'year']).sum()

# Calculate the birth rate per 1,000 people
grouped_df['Birth Rate'] = (grouped_df['Births'] /
grouped_df['Population']) * 1000

# Reset index to use columns easily
grouped_df = grouped_df.reset_index()

grouped_df = grouped_df[grouped_df['year'] <= 2016]
# Map FIPS code to state names for the plot
grouped_df['State Name'] = grouped_df['GEOID'].map(state_dict)
csv_data['State Name'] = csv_data['GEOID'].map(state_dict)
# Plotting the trend of birth rates over years for each major state
plt.figure(figsize=(10, 6))
for state in state_names:
    subset = grouped_df[grouped_df['State Name'] == state]
    subsetmodel = csv_data[csv_data['State Name'] == state]
    if not subset.empty:
        plt.plot(subset['year'], subset['Birth Rate'], marker='o',
```

```
label=state)
        plt.plot(subsetmodel['year'], subsetmodel['Birth Rate'],
marker='o', label=state)

plt.title('Trend of Birth Rate per 1,000 People by State')
plt.xlabel('Year')
plt.ylabel('Birth Rate per 1,000 People')
plt.legend(title='State', loc='upper left', bbox_to_anchor=(1, 1))  #
Adjust legend outside of plot
plt.grid(True)
plt.tight_layout()
plt.show()
```




```
# Replace this line with csv_data =
pd.read_csv('your_actual_file_path.csv')
csv_data = pd.read_csv('state_means.csv')

# Define the correct range of years based on the number of timestamp
columns
years = range(2011, 2011 + len(csv_data.columns) - 1)

# Rename the columns with actual years for clarity before melting
csv_data.columns = ['State Name'] + [str(year) for year in years]

# Melting the DataFrame
melted_data = pd.melt(csv_data, id_vars=['State Name'],
var_name='year', value_name='Birth Rate')
```

```python
# Display the melted DataFrame
print(melted_data)

# Optionally, save the melted DataFrame for further analysis
melted_data.to_csv('melted_state_data.csv', index=False)
```

```
    State Name  year  Birth Rate
0      Alabama  2011    0.103659
1   California  2011    0.103852
2      Florida  2011    0.103850
3     New York  2011    0.103695
4      Alabama  2012    0.101655
5   California  2012    0.101573
6      Florida  2012    0.101927
7     New York  2012    0.101485
8      Alabama  2013    0.103980
9   California  2013    0.103871
10     Florida  2013    0.103877
11    New York  2013    0.103844
12     Alabama  2014    0.108208
13  California  2014    0.108264
14     Florida  2014    0.108072
15    New York  2014    0.107942
16     Alabama  2015    0.112800
17  California  2015    0.112871
18     Florida  2015    0.113020
19    New York  2015    0.112733
20     Alabama  2016    0.108010
21  California  2016    0.108213
22     Florida  2016    0.108118
23    New York  2016    0.107905
```