Birthrate Analysis

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

In [ ]:
```
births=pd.read_csv('births.csv')
```

In [ ]:
```
births.head()
```

In [ ]:
```
births.tail()
```

replacing na values with 0

In [ ]:
```
births.isna().sum()
```

In [ ]:
```
births['day'].fillna(0, inplace=True)
```

changing data type

In [ ]:
```
births['day'] = births['day'].astype(int)
births.info()
```

calculating decade for each year

In [ ]:
```
births['decade']=10*(births['year']//10)
```

gender by decade

creating pivot table

In [ ]:
```
birth_gender=births.pivot_table('births', index='decade', columns='gender',
birth_gender
```

plotting graph

In [ ]:
```
rcParams['figure.figsize'] = (4,4)
birth_gender.plot()
plt.ylabel("Total births per year")
plt.show()
```

removing outliers

In [ ]:
```
quartiles = np.percentile(births['births'], [25, 50, 75])
mean = quartiles[1]
sigma = 0.74 * (quartiles[2] - quartiles[0]) #0.74 is interquartile range o
```

use query() to filter out rows with births outside this value

In [ ]:
```
births = births.query('(births > @mean - 5 * @sigma) & (births < @mean + 5
births.index = pd.to_datetime(10000 * births.year + 100 * births.month + bi
                              format='%Y%m%d')
# @ is used to specify variable defined outside query
```

Average births by day of week

adding day of week column

In [ ]:
```
births['day of week'] = births.index.dayofweek
```

creating pivot table

In [ ]:
```
births_day = births.pivot_table('births', index='day of week',columns='deca
births_day.index = ['Mon', 'Tues', 'Wed', 'Thurs', 'Fri', 'Sat', 'Sun']
births_day
```

visualisation

In [ ]:
```
births_day.plot()
plt.ylabel("Average Births by Day")
plt.show()
```

we can see that births were less common on weekends as compared to weekdays

average births by day of year

In [ ]:
```
births_month = births.pivot_table('births', [births.index.month, births.ind
print(births_month.head())

births_month.index = [pd.Timestamp(2000, month, day)
                      for (month, day) in births_month.index]
print(births_month.head())
```

In [ ]:
```
fig, ax = plt.subplots(figsize=(12, 4))
births_month.plot(ax=ax)
plt.show()
```

In [ ]: