Assignment4

October 12, 2020

1 Assignment 4

Before working on this assignment please read these instructions fully. In the submission area, you will notice that you can click the link to **Preview the Grading** for each step of the assignment. This is the criteria that will be used for peer grading. Please familiarize yourself with the criteria before beginning the assignment.

This assignment requires that you to find **at least** two datasets on the web which are related, and that you visualize these datasets to answer a question with the broad topic of **economic activity or measures** (see below) for the region of **Ann Arbor, Michigan, United States**, or **United States** more broadly.

You can merge these datasets with data from different regions if you like! For instance, you might want to compare **Ann Arbor, Michigan, United States** to Ann Arbor, USA. In that case at least one source file must be about **Ann Arbor, Michigan, United States**.

You are welcome to choose datasets at your discretion, but keep in mind **they will be shared with your peers**, so choose appropriate datasets. Sensitive, confidential, illicit, and proprietary materials are not good choices for datasets for this assignment. You are welcome to upload datasets of your own as well, and link to them using a third party repository such as github, bitbucket, pastebin, etc. Please be aware of the Coursera terms of service with respect to intellectual property.

Also, you are welcome to preserve data in its original language, but for the purposes of grading you should provide english translations. You are welcome to provide multiple visuals in different languages if you would like!

As this assignment is for the whole course, you must incorporate principles discussed in the first week, such as having as high data-ink ratio (Tufte) and aligning with Cairo's principles of truth, beauty, function, and insight.

Here are the assignment instructions:

- State the region and the domain category that your data sets are about (e.g., **Ann Arbor, Michigan, United States** and **economic activity or measures**).
- You must state a question about the domain category and region that you identified as being interesting.
- You must provide at least two links to available datasets. These could be links to files such as CSV or Excel files, or links to websites which might have data in tabular form, such as Wikipedia pages.
- You must upload an image which addresses the research question you stated. In addition to addressing the question, this visual should follow Cairo's principles of truthfulness, functionality, beauty, and insightfulness.

- You must contribute a short (1-2 paragraph) written justification of how your visualization addresses your stated research question.

What do we mean by **economic activity or measures**? For this category you might look at the inputs or outputs to the given economy, or major changes in the economy compared to other regions.

1.1 Tips

- Wikipedia is an excellent source of data, and I strongly encourage you to explore it for new data sources.
- Many governments run open data initiatives at the city, region, and country levels, and these are wonderful resources for localized data sources.
- Several international agencies, such as the United Nations, the World Bank, the Global Open Data Index are other great places to look for data.
- This assignment requires you to convert and clean datafiles. Check out the discussion forums for tips on how to do this from various sources, and share your successes with your fellow students!

1.2 Example

Looking for an example? Here's what our course assistant put together for the **Ann Arbor, MI, USA** area using **sports and athletics** as the topic. Example Solution File

```
In [1]: import pandas as pd
        import matplotlib.pyplot as plt
        url_births='https://raw.githubusercontent.com/hamzaelanssari/dataset_birth_
        df_births=pd.read_csv(url_births)
        url_deaths='https://raw.githubusercontent.com/hamzaelanssari/dataset_birth_
        df_deaths=pd.read_csv(url_deaths)

In [12]: '''
         World Arab
         df_ARB_births=df_births[df_births['Country Code']=='ARB']
         df_ARB_deaths=df_deaths[df_deaths['Country Code']=='ARB']
         # Caribbean Countries
         df_CSS_births=df_births[df_births['Country Code']=='CSS']
         df_CSS_deaths=df_deaths[df_deaths['Country Code']=='CSS']
         # Central Europe and the Baltics
         df_CEB_births=df_births[df_births['Country Code']=='CEB']
         df_CEB_deaths=df_deaths[df_deaths['Country Code']=='CEB']
         #East Asia & Pacific
         df_EAS_births=df_births[df_births['Country Code']=='EAS']
         df_EAS_deaths=df_deaths[df_deaths['Country Code']=='EAS']
         #European Union
         df_EUU_births=df_births[df_births['Country Code']=='EUU']
         df_EUU_deaths=df_deaths[df_deaths['Country Code']=='EUU']
         #Latin America & Caribbean
         df_LCN_births=df_births[df_births['Country Code']=='LCN']
```

```
        df_LCN_deaths=df_deaths[df_deaths['Country Code']=='LCN']
        #North America
        df_NAC_births=df_births[df_births['Country Code']=='NAC']
        df_NAC_deaths=df_deaths[df_deaths['Country Code']=='NAC']
        '''
```

Out[12]: "\nWorld Arab\ndf_ARB_births=df_births[df_births['Country Code']=='ARB']\r

```
In [13]: df_births.rename(columns={'Value': 'Value_Births'},inplace=True)
         df_deaths.rename(columns={'Value': 'Value_Deaths'},inplace=True)

In [14]: #Check empty Birth_Data
         df_births.isnull().sum()
         #Other method df_birth.isnull().values.any()
         #Check empty Deaths_Data
         df_deaths.isnull().sum()
```

Out[14]:
```
Country Name    0
Country Code    0
Year            0
Value_Deaths    0
dtype: int64
```

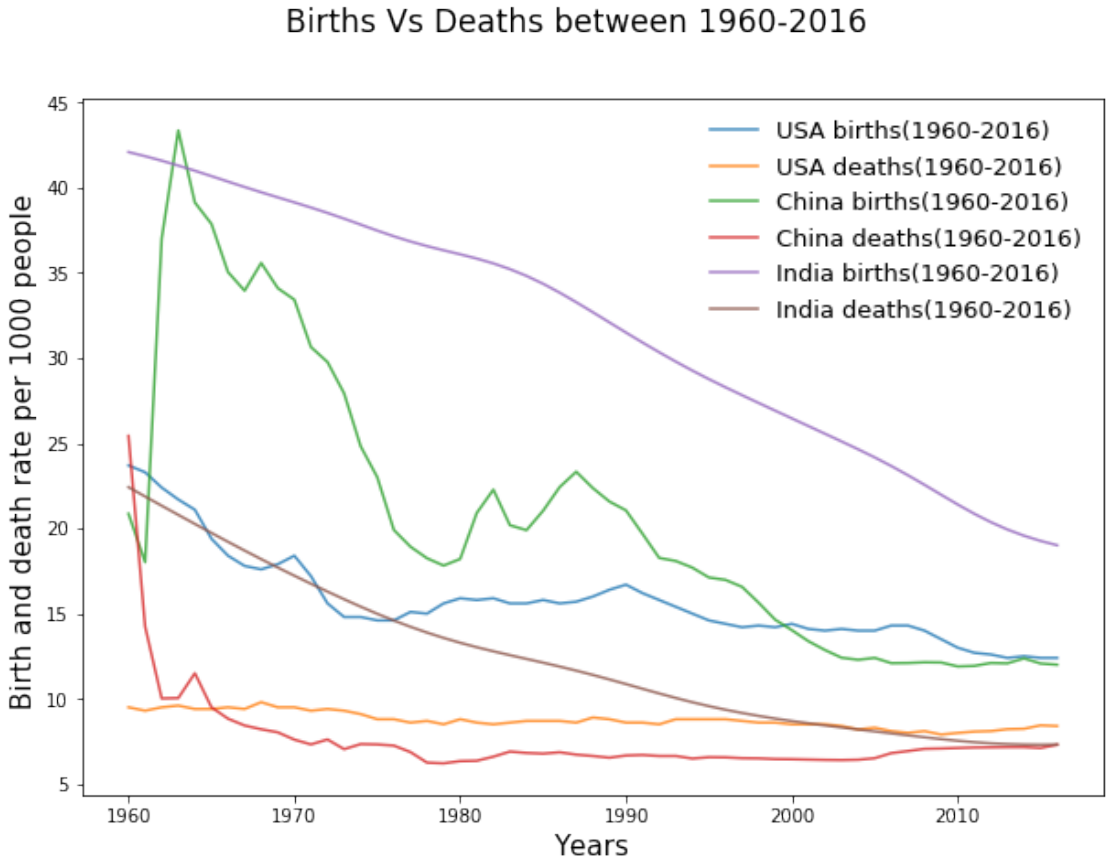
```
In [17]: #USA
         df_USA_births=df_births[df_births['Country Code']=='USA']
         df_USA_deaths=df_deaths[df_deaths['Country Code']=='USA']
         #CHINA
         df_CHN_births=df_births[df_births['Country Code']=='CHN']
         df_CHN_deaths=df_deaths[df_deaths['Country Code']=='CHN']
         #INDIA
         df_IND_births=df_births[df_births['Country Code']=='IND']
         df_IND_deaths=df_deaths[df_deaths['Country Code']=='IND']

In [5]:

In [48]: # merge data of births with data of deaths
         # USA
         df_USA=pd.merge(df_USA_births,df_USA_deaths,on=['Year','Country Code','Cou
         df_USA.set_index('Year',inplace=True)
         # CHINA
         df_CHN=pd.merge(df_CHN_births,df_CHN_deaths,on=['Year','Country Code','Cou
         df_CHN.set_index('Year',inplace=True)
         # INDIA
         df_IND=pd.merge(df_IND_births,df_IND_deaths,on=['Year','Country Code','Cou
         df_IND.set_index('Year',inplace=True)

         # Set Axis
         axis=df_USA.index.tolist()
         df_CHN
```

Out[48]:

Year	Country Name	Country Code	Value_Births	Value_Deaths
1960	China	CHN	20.86	25.43
1961	China	CHN	18.02	14.24
1962	China	CHN	37.01	10.02
1963	China	CHN	43.37	10.04
1964	China	CHN	39.14	11.50
1965	China	CHN	37.88	9.50
1966	China	CHN	35.05	8.83
1967	China	CHN	33.96	8.43
1968	China	CHN	35.59	8.21
1969	China	CHN	34.11	8.03
1970	China	CHN	33.43	7.60
1971	China	CHN	30.65	7.32
1972	China	CHN	29.77	7.61
1973	China	CHN	27.93	7.04
1974	China	CHN	24.82	7.34
1975	China	CHN	23.01	7.32
1976	China	CHN	19.91	7.25
1977	China	CHN	18.93	6.87
1978	China	CHN	18.25	6.25
1979	China	CHN	17.82	6.21
1980	China	CHN	18.21	6.34
1981	China	CHN	20.91	6.36
1982	China	CHN	22.28	6.60
1983	China	CHN	20.19	6.90
1984	China	CHN	19.90	6.82
1985	China	CHN	21.04	6.78
1986	China	CHN	22.43	6.86
1987	China	CHN	23.33	6.72
1988	China	CHN	22.37	6.64
1989	China	CHN	21.58	6.54
1990	China	CHN	21.06	6.67
1991	China	CHN	19.68	6.70
1992	China	CHN	18.27	6.64
1993	China	CHN	18.09	6.64
1994	China	CHN	17.70	6.49
1995	China	CHN	17.12	6.57
1996	China	CHN	16.98	6.56
1997	China	CHN	16.57	6.51
1998	China	CHN	15.64	6.50
1999	China	CHN	14.64	6.46
2000	China	CHN	14.03	6.45
2001	China	CHN	13.38	6.43
2002	China	CHN	12.86	6.41
2003	China	CHN	12.41	6.40
2004	China	CHN	12.29	6.42
2005	China	CHN	12.40	6.51

```
2006        China          CHN          12.09          6.81
2007        China          CHN          12.10          6.93
2008        China          CHN          12.14          7.06
2009        China          CHN          12.13          7.08
2010        China          CHN          11.90          7.11
2011        China          CHN          11.93          7.14
2012        China          CHN          12.10          7.15
2013        China          CHN          12.08          7.16
2014        China          CHN          12.37          7.16
2015        China          CHN          12.07          7.11
2016        China          CHN          12.00          7.30
```

In [47]:
```
fig, ax = plt.subplots(1, figsize=(10, 7))
#colors = ['green', 'red']
#ax.axis(ymin=0,ymax=100)
# USA
ax.plot(axis,df_USA['Value_Births'].tolist(),alpha = 0.8, label = 'USA bir
ax.plot(axis,df_USA['Value_Deaths'].tolist(),alpha = 0.8, label = 'USA dea
# CHINA
ax.plot(axis,df_CHN['Value_Births'].tolist(),alpha = 0.8, label = 'China b
ax.plot(axis,df_CHN['Value_Deaths'].tolist(),alpha = 0.8, label = 'China d
# INDIA
ax.plot(axis,df_IND['Value_Births'].tolist(),alpha = 0.8, label = 'India b
ax.plot(axis,df_IND['Value_Deaths'].tolist(),alpha = 0.8, label = 'India d
ax.legend(loc ='best', frameon=False,fontsize=13)
ax.set_xlabel('Years',fontsize=15)
ax.set_ylabel('Birth and death rate per 1000 people ',fontsize=15)
fig.suptitle('Births Vs Deaths between 1960-2016',fontsize=17)
plt.show()
```


Births Vs Deaths between 1960-2016
Birth and death rate per 1000 people
Years
USA births(1960-2016)
USA deaths(1960-2016)
China births(1960-2016)
China deaths(1960-2016)
India births(1960-2016)
India deaths(1960-2016)
45
40
35
30
25
20
15
10
5
1960
1970
1980
1990
2000
2010