DCS 640 Data Presentation & Visualization (DSC640-T302 2231-1)

Bellevue University

2.2 Exercises: Line Charts and Step Charts

Author: Jake Meyer

Date: 12/12/2022

Assignment Instructions:

Submit 1 line chart and 1 step chart with Python

In [1]:

```
'''
Import the necessary libraries to complete Exercise 2.2.
'''
import numpy as np
import pandas as pd
import seaborn as sns
import scipy.stats
import matplotlib
import matplotlib.pyplot as plt
```

In [2]:

```
'''
Check the versions of the packages.
'''
print('numpy version:', np.__version__)
print('pandas version:', pd.__version__)
print('seaborn version:', sns.__version__)
print('matplotlib version:', matplotlib.__version__)
```

```
numpy version: 1.20.3
pandas version: 1.3.4
```

```
seaborn version: 0.11.2
matplotlib version: 3.4.3
```

Dataset Understanding

In [11]:

```
'''
Import the datasets.
Note: A copy of the CSV file was placed into the same directory as this notebook.
Utilize pd.read_excel() to read the file as a pandas data frame.
'''
df1 = pd.read_excel('world-population.xlsm')
```

In [12]:

```
'''
Use head() function to display the first 5 rows of data of df1.
'''
df1.head()
```

Out[12]:

	Year	Population
0	1960	3028654024
1	1961	3068356747
2	1962	3121963107
3	1963	3187471383
4	1964	3253112403

In [13]:

```
'''
Understand the shape of the df1.
'''
print('There are {} rows and {} columns in the df1.'.format(df1.shape[0], df1.shape[1]))
```

```
There are 50 rows and 2 columns in the df1.
```

In [14]:

```
'''
Convert Year to Date Time for df1.
Comment this section out of the code. Originally tried plotting with Date Type, however had success leaving it as numeric
'''
# df1['Year'] = pd.to_datetime(df1['Year'])
```

Out[14]:

```
'\nConvert Year to Date Time for df1.\n'
```

In [15]:

```
'''
Find the type of data within each df1 column initially.
'''
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50 entries, 0 to 49
Data columns (total 2 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   Year        50 non-null     int64
 1   Population  50 non-null     int64
dtypes: int64(2)
memory usage: 928.0 bytes
```

In [16]:

```
'''
Understand if there are any missing values in df1.
'''
df1.isna().sum().sort_values(ascending = False)
```

Out[16]:

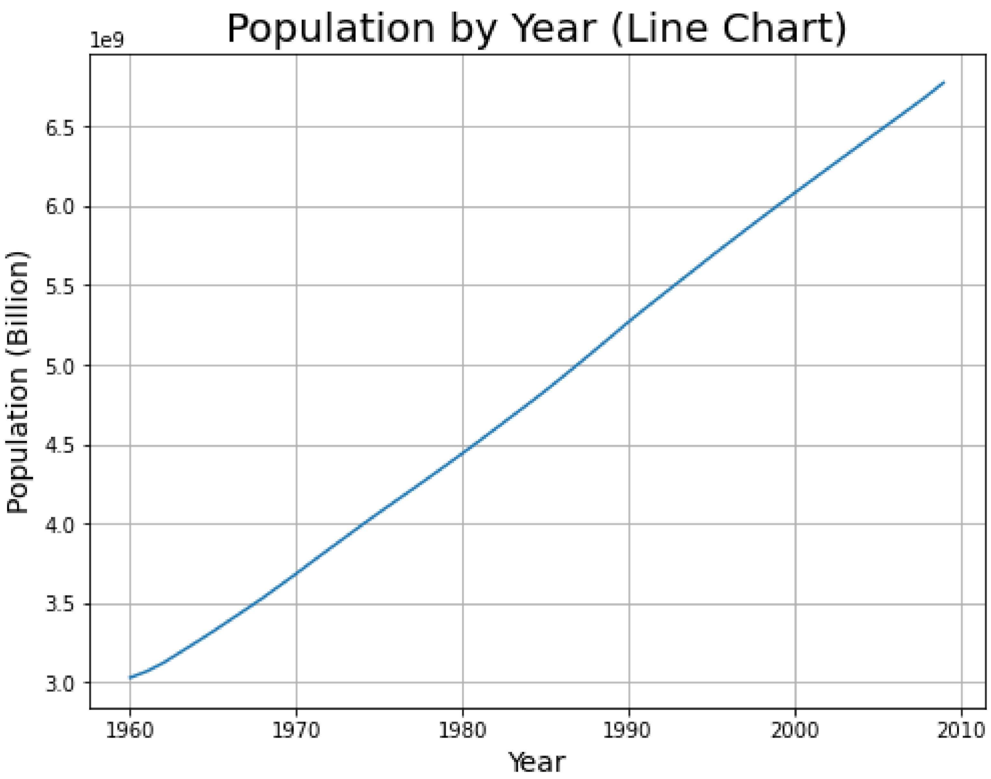
```
Year          0
Population    0
dtype: int64
```

Chart Creation from the Dataset.

Line Chart

In [19]:

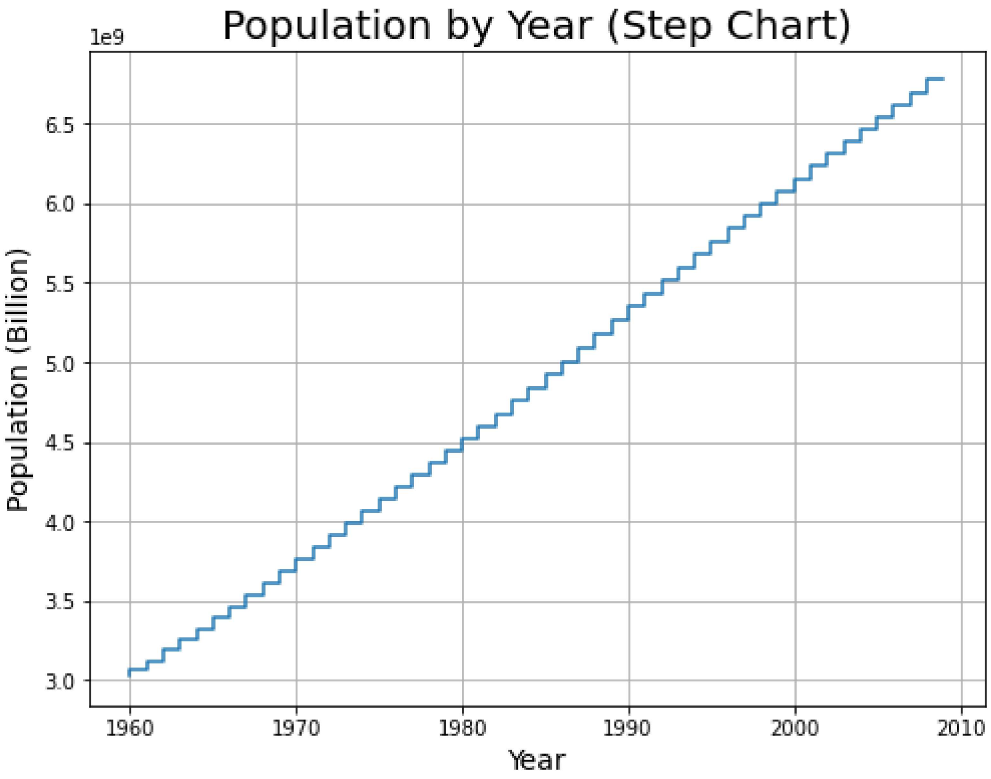
```
'''
Create a bar chart using one of the datasets from the previous section.
Use Seaborn to construct the barplot with barplot()
'''
fig = plt.figure(figsize = (8,6))
sns.lineplot(x = 'Year', y = 'Population', data = df1)
plt.title('Population by Year (Line Chart)', fontsize = 20)
plt.xlabel('Year', fontsize = 14)
plt.ylabel('Population (Billion)', fontsize = 14)
plt.grid(True)
plt.show()
```



Step Chart

In [20]:

```
'''
Create a bar chart using one of the datasets from the previous section.
Use Seaborn to construct the barplot with barplot()
'''
fig = plt.figure(figsize = (8,6))
sns.lineplot(x = 'Year', y = 'Population', data = df1, drawstyle ='steps-pre')
plt.title('Population by Year (Step Chart)', fontsize = 20)
plt.xlabel('Year', fontsize = 14)
plt.ylabel('Population (Billion)', fontsize = 14)
plt.grid(True)
plt.show()
```


Population by Year (Step Chart)
1e9
6.5
6.0
5.5
5.0
4.5
4.0
3.5
3.0
Population (Billion)
1960
1970
1980
1990
2000
2010
Year