DSC640 WEEKS 3 & 4

EXERCISE 2.2

Datasets - You may also download them directly from this link:
https://content.bellevue.edu/cst/dsc/640/datasets/ex2-2.zip
(https://content.bellevue.edu/cst/dsc/640/datasets/ex2-2.zip)

Exercise Goal:

You need to submit 3 line charts and 3 step charts using Tableau or PowerBI, Python and R using the data below (or your own datasets). You can also submit using D3, though not required. You can choose which library to use in Python or R, documentation is provided to help you decide and as you start to play around in the libraries, you will decide which you prefer.

CINDY HERRERA

Decemeber 19, 2019

1. Data collection: importing data and analyzing

In [1]:
```
import os
import datetime
import pprint
import pandas as pd
import numpy as np
from pandas import ExcelWriter
from pandas import ExcelFile
import matplotlib as mpl
import matplotlib.pyplot as plt
%matplotlib inline
```

In [2]:
```
os.getcwd()
```

Out[2]:
```
'/Users/Cindy/Desktop/00 data640/ex2-2'
```

Load data into dataframe

In [3]:
```
wp = pd.read_excel('/Users/Cindy/Desktop/00 data640/ex2-2/world-population.xlsm')
```

Check the dimension of table and view data

In [4]:
```
print("The dimension of the table is: ", wp.shape)

print(wp)
```

```
The dimension of the table is:  (50, 2)
    Year  Population
0   1960  3028654024
1   1961  3068356747
2   1962  3121963107
3   1963  3187471383
4   1964  3253112403
5   1965  3320396924
6   1966  3390712300
7   1967  3460521851
8   1968  3531547287
9   1969  3606994959
10  1970  3682870688
11  1971  3761750672
12  1972  3839147707
13  1973  3915742695
14  1974  3992806090
15  1975  4068032705
16  1976  4141383058
17  1977  4214499013
18  1978  4288485981
19  1979  4363754326
20  1980  4439638086
21  1981  4516734312
22  1982  4595890494
23  1983  4675178812
24  1984  4753877875
25  1985  4834206631
26  1986  4918126890
27  1987  5004006066
28  1988  5090899475
29  1989  5178059174
30  1990  5266783430
31  1991  5351836347
32  1992  5433823608
33  1993  5516863641
34  1994  5598658151
35  1995  5681689325
```

```
36  1996  5762235749
37  1997  5842585301
38  1998  5921799957
39  1999  6001269553
40  2000  6078274622
41  2001  6155652495
42  2002  6232413711
43  2003  6309266583
44  2004  6385778679
45  2005  6462054420
46  2006  6538196688
47  2007  6614396907
48  2008  6692030277
49  2009  6775235741
```

In [5]:
```
wp.head()
```

Out[5]:

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

2. Data formatting

After reviewing the data set, years are formatted correctly but we have to format the 'Population' column to display separators in the numbers.

In [6]:
```
type(wp.Population)
```

Out[6]:
```
pandas.core.series.Series
```

In [7]:
```
wp['Population'] = wp.apply(lambda x: "{:,}".format(x['Population']),
axis=1)
print(wp.head())
```

```
   Year     Population
0  1960  3,028,654,024
1  1961  3,068,356,747
2  1962  3,121,963,107
3  1963  3,187,471,383
4  1964  3,253,112,403
```

I am looking for the min and max values for our axis when we plot the data

In [8]:
```
wp.min()
```

Out[8]:
```
Year                    1960
Population     3,028,654,024
dtype: object
```

In [9]:
```
wp.max()
```

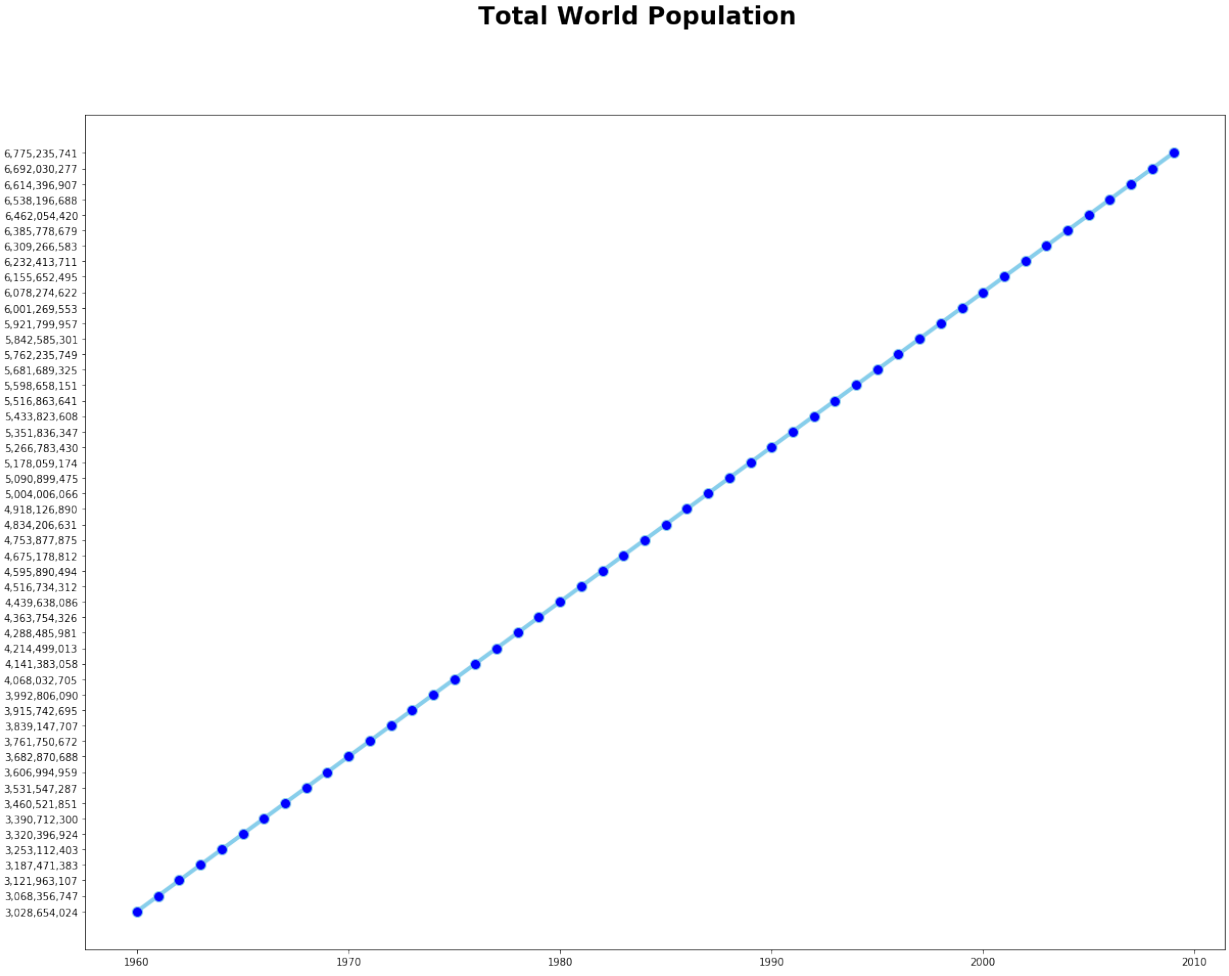
Out[9]:
```
Year                    2009
Population     6,775,235,741
dtype: object
```

3. Data Model and Analyzing

Now we can go into graphing the data

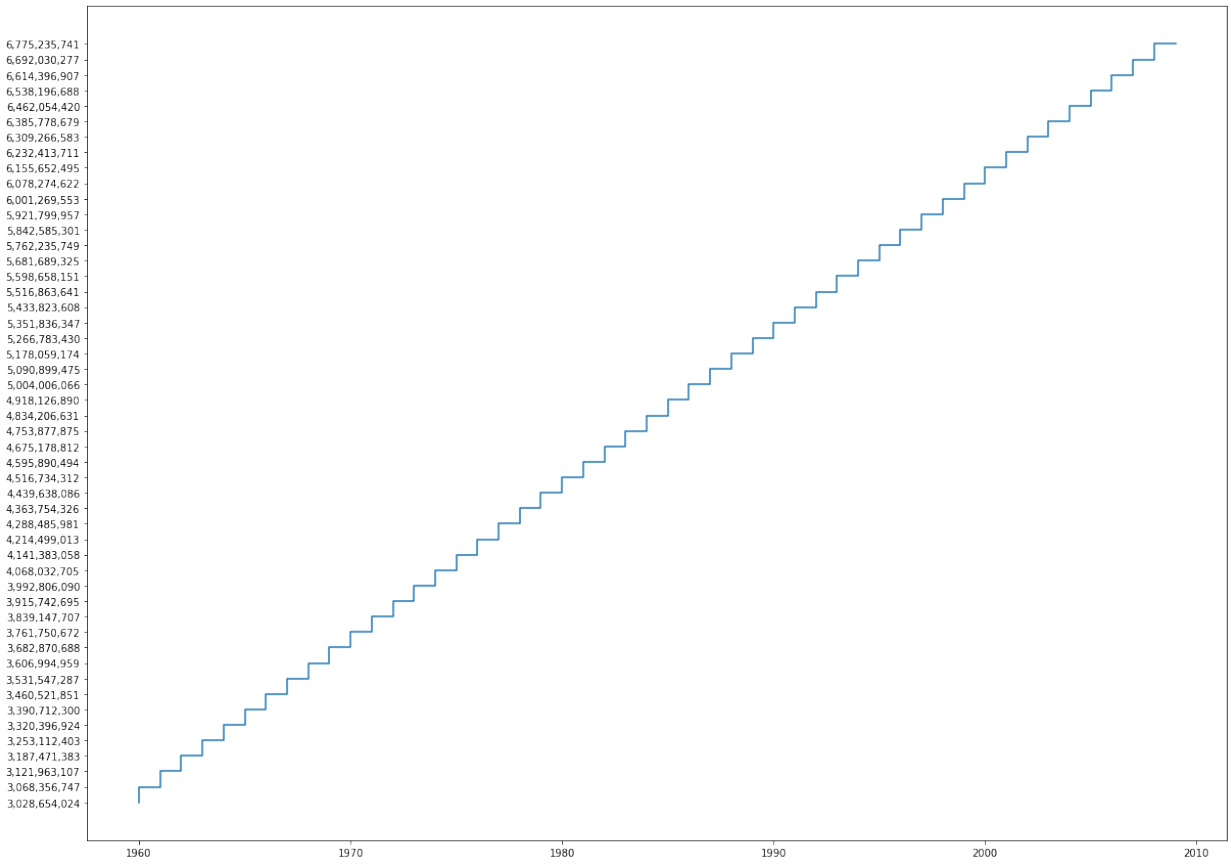
Line Chart

In [12]:
```
fig = plt.figure(figsize=(20,15))
fig.suptitle('Total World Population', fontsize=24, fontweight='bold')


wp['Year'] = pd.to_datetime(wp['Year'], format='%Y')
plt.plot( 'Year', 'Population', data=wp, marker='o', markerfacecolor='
blue', markersize=10,
         color='skyblue', linewidth=4)
plt.show()
```



Step Chart

In [13]:
```
fig = plt.figure(figsize=(20,15))
fig.suptitle('Total World Population', fontsize=24, fontweight='bold')

X = wp['Year']
Y = wp['Population']
plt.step(X, Y)
plt.show()
```



In [ ]:
```
```