### Author: Souvik Nath

### Dated: 18.12.2019

**Objective**: The objective of this document is to walk the user through the ‘Seaborn python program file’ on the different functionalities being used to attain the end outcome of creating a heatmap

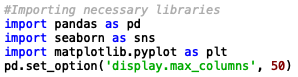
**SECTION 1: Importing python libraries**

This is the first section of the code, where all the necessary libraries are being imported. Python programmers might find this resemblance with almost all python programs being written.

*NOTE: Line number 4 is a pandas functionality that allows the programmer to set the number of visible columns for any* ***Pandas DataFrame*** *object being printed onto the IPython console. This is also available for setting the number of visible rows if a programmer wishes to print the* ***Pandas******DataFrame*** *object via this command:*

pandas.set\_option(‘display.max\_columns, 100)

**CODE:**

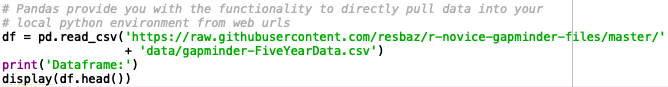


**SECTION 2: Importing flat files as Pandas Objects**

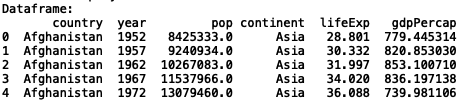
This section of the code is where we pull the .csv file from the provided web url.

**CODE:**

1. **pd.read\_csv / pandas.read\_csv:** This command provides the programmer with the functionality to pull flat files from both local directory or web urls
2. **display:** The display command prettify the output of printing a Pandas DataFrame object
3. **df.head:** This only restricts the console from printing the whole dataframe and prevents memory overhead, to the first five rows as the default number of displayed rows. The programmer can choose to tweak the number of rows he/she wants printed under the df.head() command



**OUTPUT:**



**SECTION 3: Creating a pivot table from the original frame**

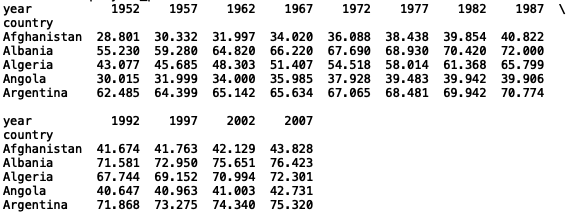
This section of the code is where we create the pivot table with available **Pandas** functionalities.

**CODE:**

1. **pd.pivot\_table / pandas.pivot\_table:** This command allows the programmer to create a pivot table from the original **Pandas DataFrame** object
   1. **df:** This is where the original **Pandas DataFrame** object is passed from where the pivot table has to be created
   2. **values:** This is where the programmer selects the field that is to be used to fill the new **Pandas DataFrame**’s values
   3. **index:** This argument allows the programmer to select the field that is to be used as the index for the new **Pandas DataFrame**
   4. **columns:** This argument allows the programmer to select the field, whose values are to be used as the columns for the new **Pandas DataFrame**

****

**OUTPUT:**

****

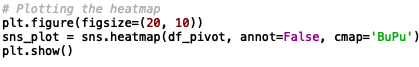
**SECTION 4: Plotting the heatmap from the created pivot table**

This section of the code is where we plot a heatmap of the pivot table created using **Seaborn,** with the **year** along x-axes, **country** along y-axes and **lifeExp** filled within cells.

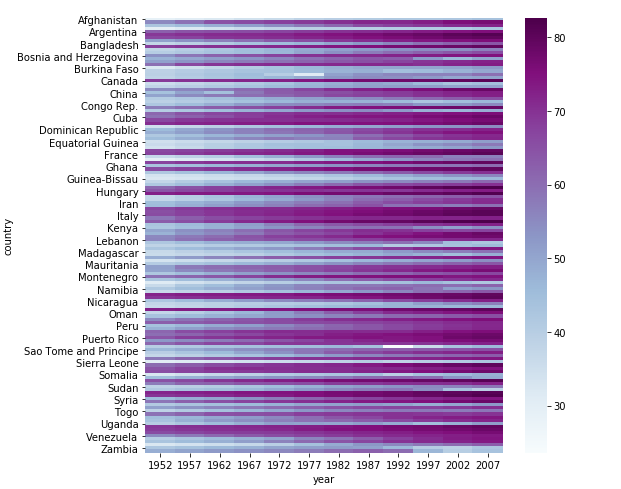
**Seaborn:** Seaborn is a popular python library used for creating intuitive statistical graphics or visualizations in python. It is built on top of the **matplotlib** library and is closely integrated with the **Pandas** data structures.

**CODE:**

1. **plt.figure:** figure() is used to create a figure object. plt.figure is used to tweak the size of the visualization or when the programmer wants to add multiple axes to the plot
2. **sns.heatmap / seaborn.heatmap:** It is used to create a heatmap
   1. **df\_pivot:** It is where the programmer defines the **Pandas DataFrame** object to be used to create the heatmap
   2. **annot:** This argument only takes in boolean **True/False.** If it is set to **True,** all the data values would be written in each cell
   3. **cmap:** This argument allows the programmer to map the data values to a color space under a list of matplotlib color options
3. **plt.show:** This command just displays all the figure/figures within one or more interactive windows. This is majorly used when matplotlib is being called from within a script

****

**OUTPUT:**

****

**SECTION 4: Exporting the heatmap figure into a file of .png format**

This section of the code is where we export the heatmap as a .png file

**CODE:**

1. **sns\_plot.get\_figure:** This command returns the figure instance
2. **fig.savefig:** This command saves the figure into a .png file

