Animation with PowerBI

**As you proceed with the assignment, follow the written instructions. Screenshots are provided ONLY as a reference.**

**Make sure you submit all screenshots with a clearly visible menu bar including the date and timestamp.**

# **Objective**

The objective of this exercise is to develop skills on how to visualize the data set using PowerBI tool.

**Step 1: Download the CSV data file provided in eLearning.**

**Step 2: Installation of PowerBI**

Click on the link below to download and install PowerBI Desktop.

<https://powerbi.microsoft.com/en-us/downloads/>

**Graphical user interface, Word

Description automatically generated**

The browser will redirect you to the Microsoft Store.

Graphical user interface, text, application

Description automatically generated

Click on *Install*.

**Step 3: Write the following python program in Jupyter Notebook**

The purpose of this program is to format the data of the csv file and make it appropriate to use for animation with PowerBI. This step allows you to experience how quickly you can make file format changes using a Jupyter Notebook. The equivalent work requires several hours in excel. We will make the following format changes to the csv data file you downloaded.

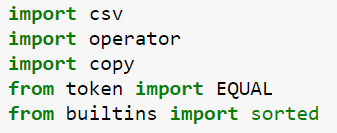
* It will merge the columns *Place* and *State* and make it one column with the format *State-Place*, turning these two columns into a location. (Lafayette is in more than one state)
* It will transpose *Years* and *Population* from their current row format in the file, and make them columns. The updated csv file will have *Years* and *Population* as columns.
* A new column *Student First Name - Rank* will be created which will rank the *State-City* based on population.

**Note: You need to make following changes in the code**

* *‘Student First name’* in column ‘*Student First Name - Rank’* should be replaced with your first name.
* While opening the file, you need to provide the location where you have downloaded the CSV file in Step-2.
* While writing the file, you need to provide a location where you want to save the newly created CSV file.

Let’s perform these adjustments a step at a time.

**Step 1**: At first, we need to import the libraries that we will need to reference for the third party functions in our code. Add the code below to the window and remember to select *Run*.

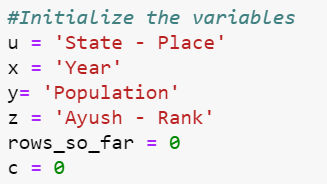


**Step 2:** We will now read in the CSV file that you downloaded.

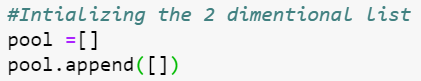
**Note:** The delimiter used in our CSV file is a “*,”* (comma). CSV files can use several types of delimiters, they may use “ “ (tab), or “;” (semi-colon), or maybe even a “^” (caret). Ours uses a “,” *delimiter* that you see below.



**Step 3:** We will now define a few variables that will determine the output file we generate from the downloaded CSV. The first 4 variables *u*, *x*, *y*, and *z* represent the organized columns for our output file. *rows\_so\_far* and *c* are just counter variables that we will use in our code loops later. **Rename column *z* with *YourFirstName-Rank***.

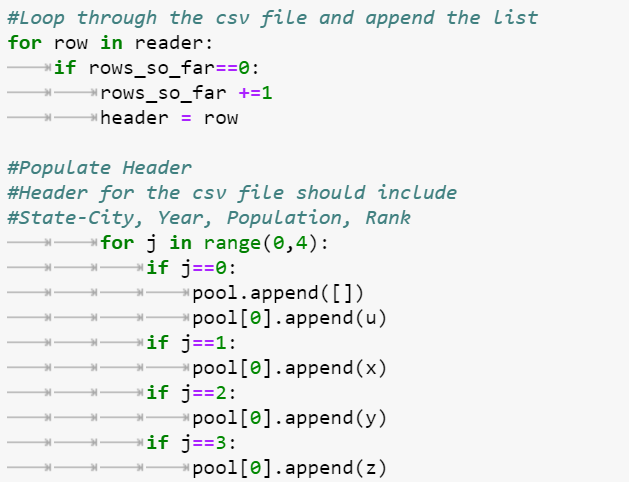


**Step 4:** We are defining a 2D array here and allowing it to be appended. Think of this as a swimming pool of data, you will throw a ball (in our case - a data row), our dog will then fetch the row and put it in his toy basket (final list). Add the code to a window and remember to select *Run*.



**Note: The code lines in Step 5 and Step 6 need to be in the same window. If they are not, the loop will not work.**

**Step 5:** There are 3 sub-loops running underneath this loop. Pay special attention to get the indention correct. Initially, we have zero rows in our pool, so let's put in the first row (Header) with the labels.



**Step 6:** Now after populating the header, we will get into the main data records that we need. Now we will append the population of a city for each year. We create a new variable *a* which is the length of our *pool* array. This is helpful in verification and not overplotting. In nested loops, there are a lot of conditional verification steps using *if-else* that need to be carried out. *.deepcopy()* is just a simple copy function we import from the *copy* library. At the end of the loop, our *rows\_so\_far* variable is increased by 1. *AGAIN, BE VERY CAREFUL WITH THE INDENTATION.*

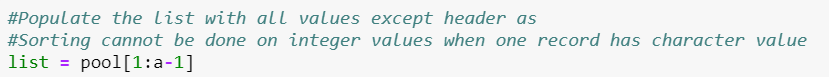


Make sure that the code entered for Step 5 and Step 6 is in the same code window, remember to select *Run*.

**Step 7**: The *len* length of the *pool* is verified after populating the sheet.



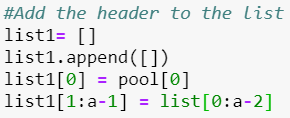
**Step 8:** We have to keep in mind that we do not populate the list including the header, explaining *a-1* in the code below. Also, sorting cannot be performed on integer values when one record (header) has a character value, so we convert it to a *list*.



**Step 9:** Here the list is sorted by year and population. The *lambda* key is a simple one parameter anonymous callable function. In the case of sorting, it can only perform and return one thing. *reverse* is a built in python function which when *True*, sorts the list in DESCENDING (reverse ascending) order.

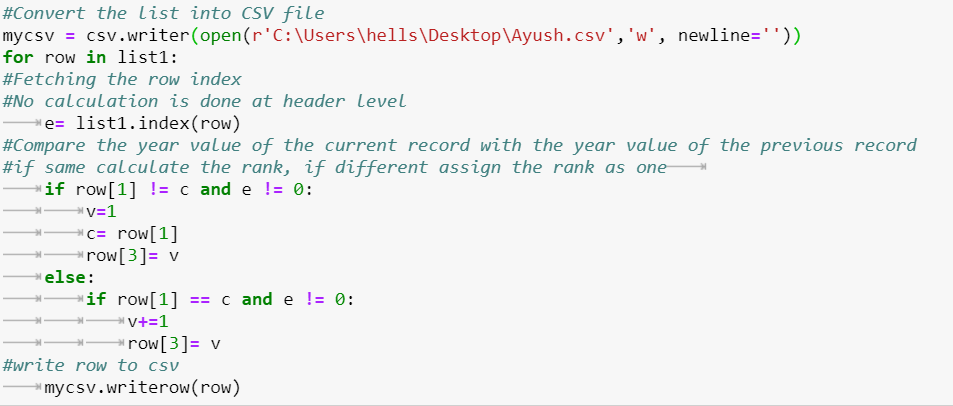


**Step 10:** Add the header to the list using proxy variables.



**Step 11:** *csv.writer* is an output file function that creates a new CSV file from our work performed in the script. We will use conditional verification using *if-else* commands to make sure that our puzzle pieces (rows), fall into the right places. *w* is a FILE MODE that creates a text file for writing. If we were reading a file, you would specifiy *rb* (READ & Binary). Enter the following code in a window.

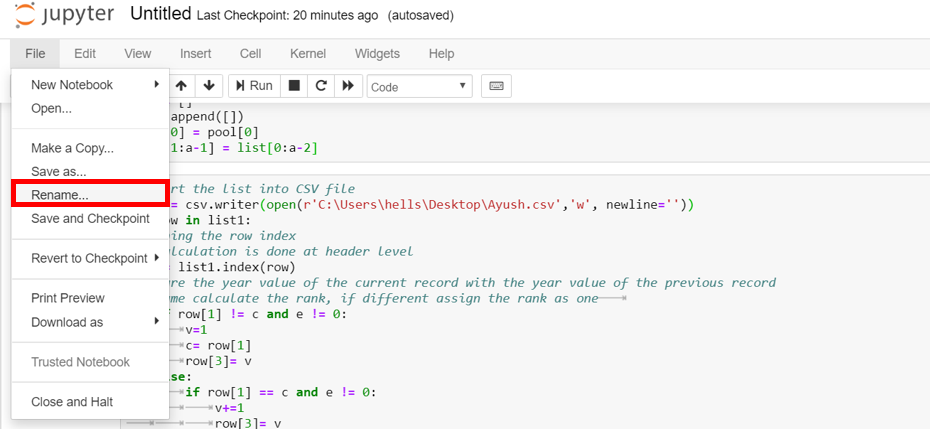
**Name your csv file with your first name *“FirstName.csv”***



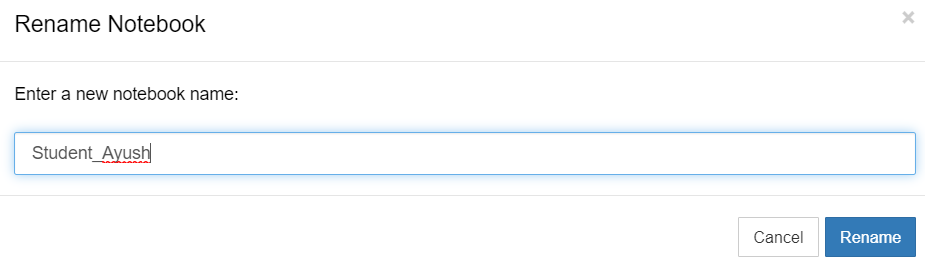
**Note**: *Make sure that the lines of code are indented* as shown in the screenshot above. Not having them indented might lead to errors or move lines of code out of the loop. *Indentation should use the tab space and not a spacebar*

**Step 3: Save the Program**

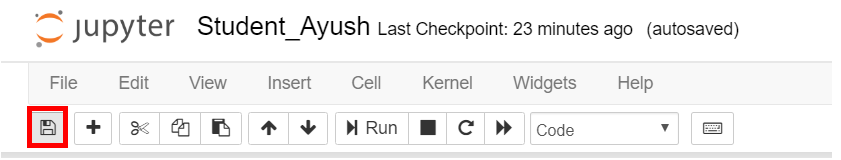
Go to *File>Rename*.

****

Rename it as Student\_FirstName and click on the *Rename* button.

****

Finally, click on the disk save icon as shown in the screenshot below.



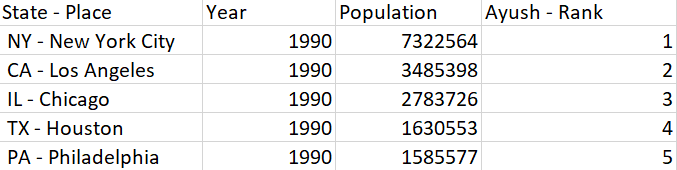
Note: Files are auto-saved in Jupyter Notebook. However, we performed the rename step to confirm saving.

**Step 4: Check Output**

The output file should be available at the path you provided in your python program.



The output of the program should be a csv file formatted as the screenshot below. Congratulations you are a file formatting python whiz!



**Question 1:** Open Microsoft Excel and paste the screenshot of your file.

**Step 5: Open PowerBI by clicking on *Launch*.**

Graphical user interface, text, application

Description automatically generated

Close the sign-in option.

**Graphical user interface

Description automatically generated**

**Step 6: Load the refined data**

In the *Home* ribbon click on *Get Data > Text/CSV*.

**Graphical user interface, application

Description automatically generated**

**Question 2:** Paste the screen shot of the load data screen.

Click the *Load* button.

*Note: The student.csv in the screenshot below is same as FirstName.csv file created earlier*

Table

Description automatically generated

**Step 7: Creating a data slicer (filter) of Rank**

Select *slicer* from the *Visualizations* pane as shown in the following screenshot.

Graphical user interface, text, application, email

Description automatically generated

Select the *Rank* column you created as part of the previous code.

A picture containing shape

Description automatically generated

Position the rank slider on the bottom right of the canvas.

**Question 3:** Paste the screen shot of the rank slider.

**Step 8: Creating a new measure**

Click on *More options* and select *New measure* as shown below.

**A picture containing table

Description automatically generated Graphical user interface, application

Description automatically generated**

Configure the measure as: ***workforce = (SUM(student[Population]))\*.60***

Replace *student* with the name of your csv file.

Background pattern

Description automatically generated with low confidence

Click on the check mark after completion.

****

**Question 4:** Paste the screen shot of the newly created measure including the formula.

**Step 9: Preparing the Scatter Chart**

Place the *Scatter chart* from the *Visualizations* pane on the canvas. Maintain the following fields in the*details*section of the visualization.

1. Drag **Student-Rank** from fields to *X axis*.
2. Drag **Population** from fields to *Y axis*.
3. Drag **Work Force** from fields to *Size section*.
4. Drag **State-Place** from fields to *legend section*.
5. Drag **Year** from fields to *play axis*.

Expand the chart to fill the screen. Make sure the slider is visible as well.

**Question 5:** Paste a screen shot of the chart.

*Change the slider to show the Top 5 ranks.*

Turn on the *category* labels under format.

**Graphical user interface, text, application

Description automatically generated**

**Question 6:** Paste a screen shot of the chart.

**Step 10:** Click on the play button to view how the population of cities change as we scan through time.

*Save the file as StudentName.pbix*

**Step 11:** **Download trial version of Camtasia software and record an animated video**

Click on the link below to download Camtasia.

<https://www.techsmith.com/video-editor.html>

Graphical user interface, application

Description automatically generated

Click on *Download Free Trial*. Once the download has completed, open the Camtasia application.

A screenshot of a computer

Description automatically generated with medium confidence

Click on *New Recording*.

Make sure that you go into the drop-down menu of the first tile. The first tile shows the preview of what is to be recorded. Adjust the recording so that it only records the PowerBI animated data visualization.

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

Open the PowerBI file created in Step 10. After you have recorded the animated data visualization, the Camtasia Editor will open.

A picture containing text, monitor, electronics, indoor

Description automatically generated

After signing in to *Start Trial*, right click on the video in the timeline and click on *Add clip speed*.

A screenshot of a computer

Description automatically generated

Extend the right side of the clip speed box until it shows the clip speed to be *0.50x*.

Graphical user interface, text, application

Description automatically generated

**NOTE:** The length of your video may vary.

After you are satisfied with the end result, click on *Export* on the top right corner of the workspace. Then click on *Local File*.

A screenshot of a phone

Description automatically generated with medium confidence

On the next dialog box that appears, click on *Produce with Watermark*.

Graphical user interface, website

Description automatically generated

Make sure to uncheck *Produce with controller* on the *Smart Player Options*.

Graphical user interface, text, application, email

Description automatically generated

Your finished video recording in mp4 format should be at the file path you have set at the last step of the *Export* process.

**Step 13: Attach and submit assignments in eLearning**

1. Attach the **assignment document** in Microsoft Word.
2. Attach the python **.ipynb file** created in step 3.
3. Attach the **.pbix file** created in Step 10.
4. Attach the **Camtasia video** created in Step 18 *as an* ***mp4*** *file type*.