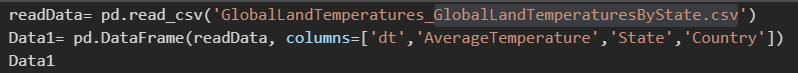
**Code Documentation**

**GlobalLandTemperaturesByState.csv**

**Data Acquisition**

converting the data of GlobalLandTemperatures\_GlobalLandTemperaturesByState.csv

into Pandas dataframe

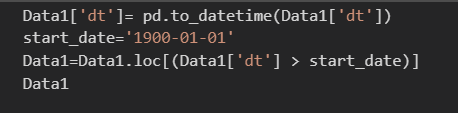


**Data Cleaning**

1. First step in the cleaning part is dropping all the data except the rows that has India, Brazil, & United States as a country



1. Second step is dropping the rows that have a date before 1900



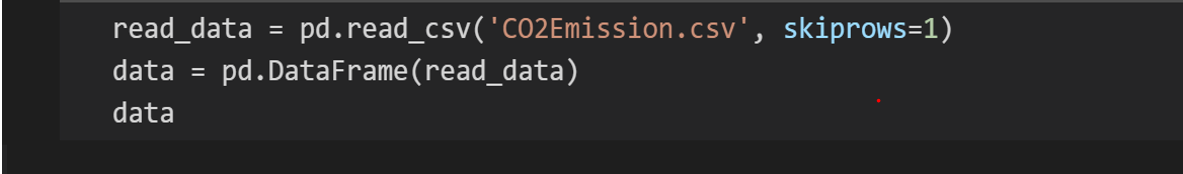
1. Drop any NA value



**Data Acquisition**

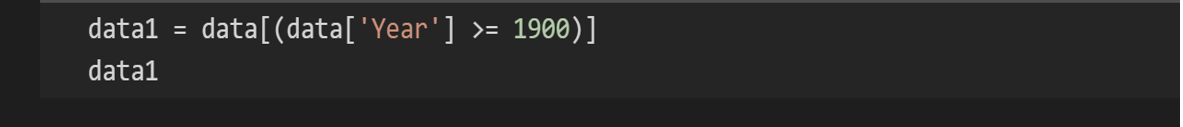
converting the data of CO2Emissions.csv

into Pandas dataframe

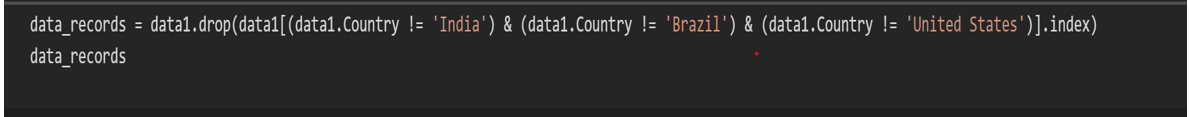


**Data Cleaning**

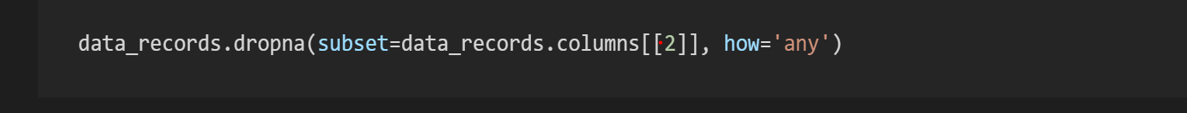
1. First step is dropping the rows that have a date before 1900



1. Second step in the cleaning part is dropping all the data except the rows that has India, Brazil, & United States as a country



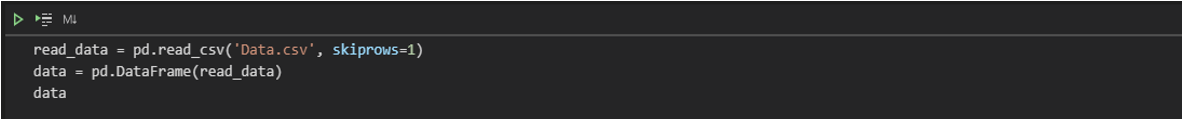
1. Drop any NA value



**Data Acquisition**

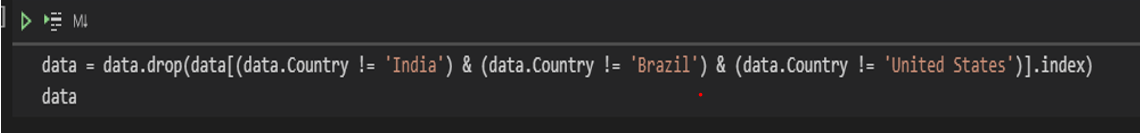
converting the data of SocioEconomic.csv

into Pandas dataframe

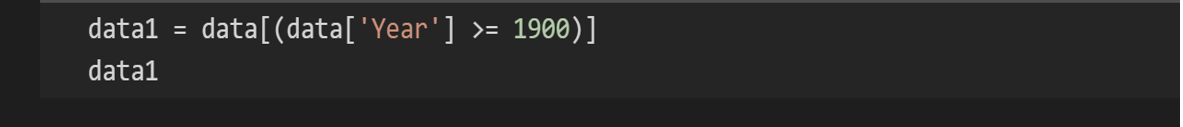


**Data Cleaning**

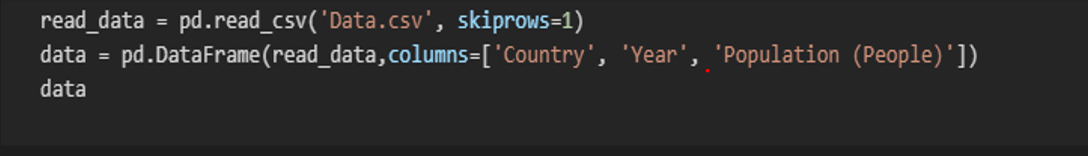
1. First step in the cleaning part is dropping all the data except the rows that has India, Brazil, & United States as a country



1. Second step is dropping the rows that have a date before 1900

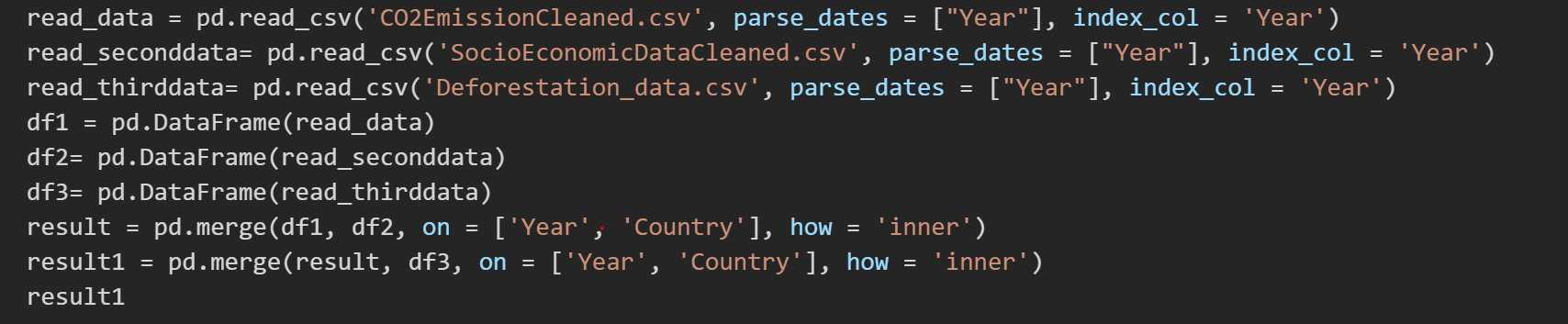


1. Third step this dataset contains columns having data about GDP and Energy used by people in tons. We would be eliminating these columns.



**Data Wrangling**

1. For data wrangling we would be reading three cleaned data namely CO2EmissionsCleaned, , SocioEconomicDataCleaned and Deforestation datasets using pandas and merge three datasets using inner join.



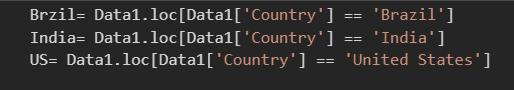
1. Save the merged data frame into a csv file



**Data Visualization**

**Average land temperature data visualization:**

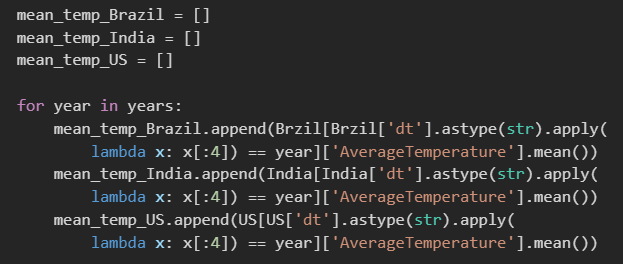
1. First, we get the data of the three countries separated



1. Second, we extract the year from the date

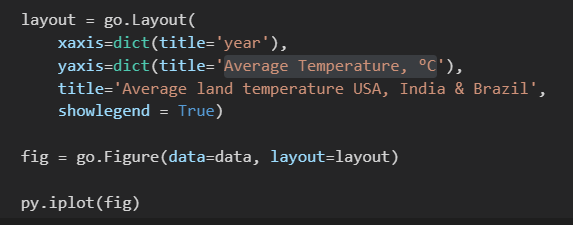


1. Create three empty arrays, and loop on the years and get the average temperature of each year for each country and append it to the arrays that we created

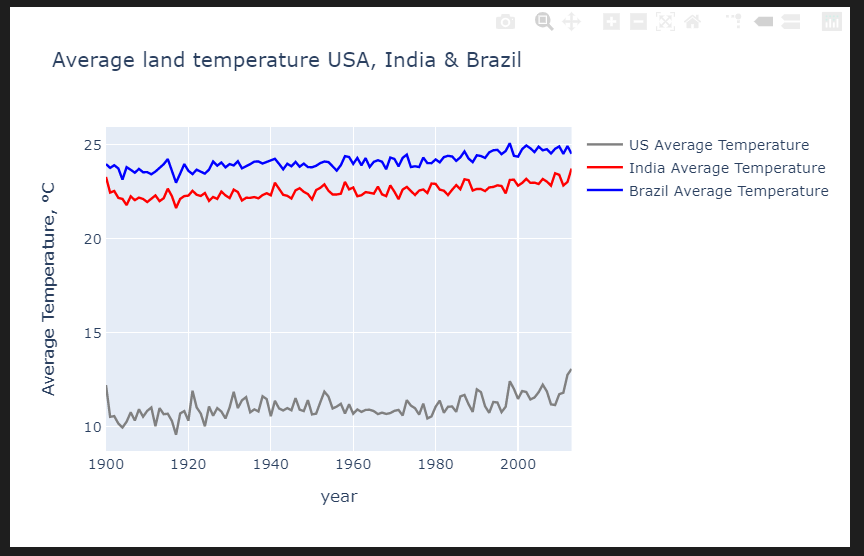


1. Using *plotly.graph\_objs as go* to add animation to the graph we draw each line for the three countries

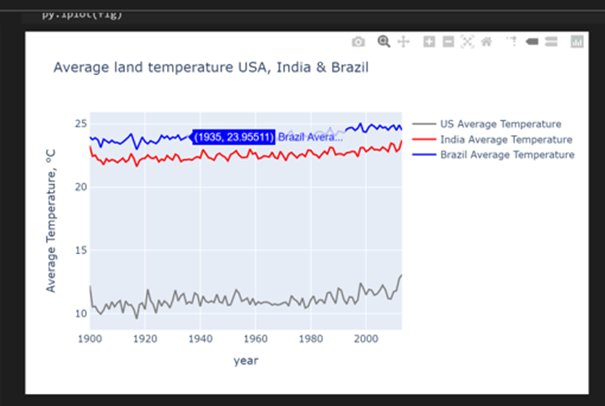


1. We customized the layout of the graph using go.layout, then we named the xaxis as a year and yaxix as Average Temperature, °C. Then using go.Figure, we were able to visualize the data
2. 

The Output:



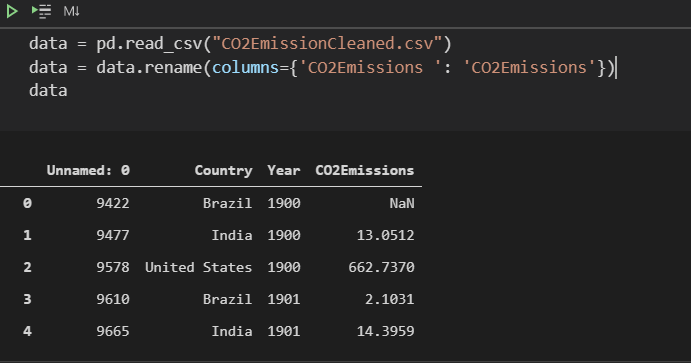
In case the pointer points at any point on the line, the country with the data and average temperature will be shown like the following



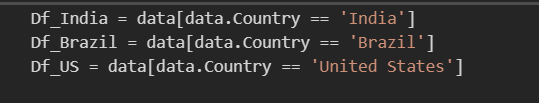
**Co2 emission dataset visualization:**

1. Create the data frame for CO2 emission dataset visualization by reading the cleaned and processed CO2EmissionCleaned.csv, CSV file through read\_csv() function.

Use rename () method to delete the extra space in the column name.



1. Create the three different data sets from the main data set, based on the Country column.



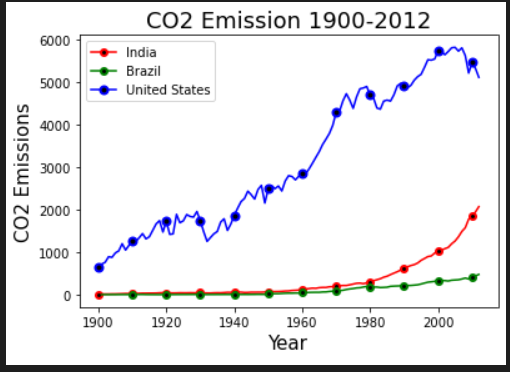
1. To plot the graph between Year and C02 emission for India, Brazil, and the United States. Decided to keep X-axis as Year from 1900 to 2012 and Y-axis as C02 emission.
2. For this, we save the Year and CO2 emission for India, Brazil, and the United States.

in x, y, y1, and y2 respectively

1. Plot the graph by using plt.plot() method of matplotlib library for India, Brazil, and the United States. (markevery = 10 to show the marker for every 10 years)



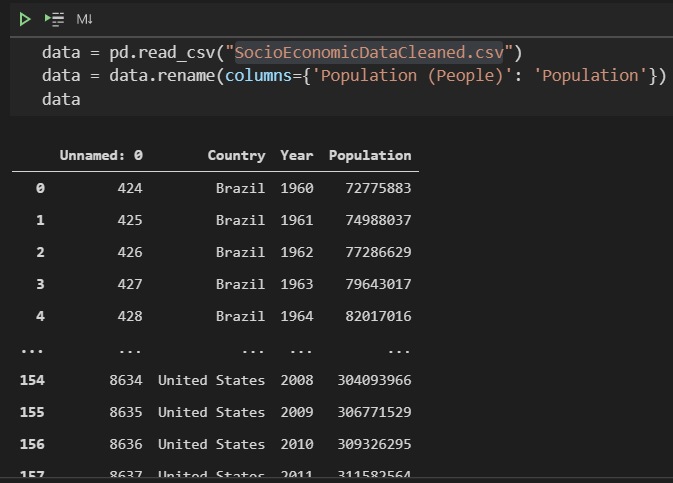
**Output:**



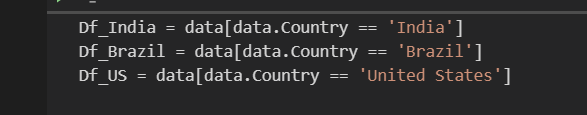
**Population growth data set visualization:**

1. Create the data frame for the Population growth dataset visualization by reading the cleaned and processed SocioEconomicDataCleaned.csv, CSV file through read\_csv () function.

Use rename () method to delete the extra space in the column name.



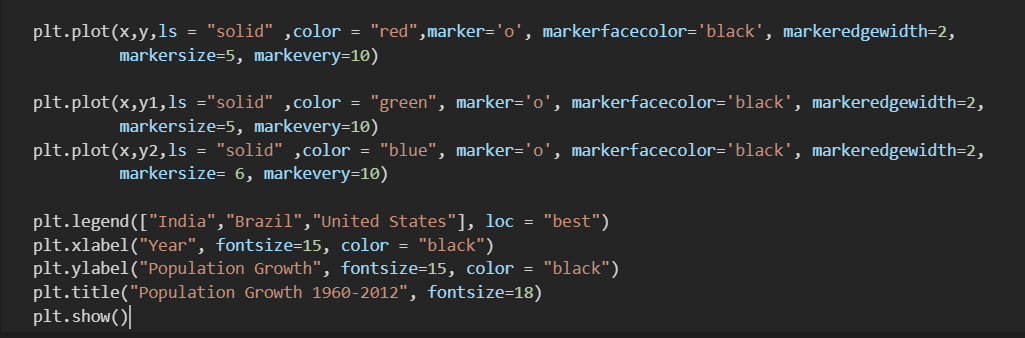
1. Create the three different data sets from the main data set, based on the Country column.



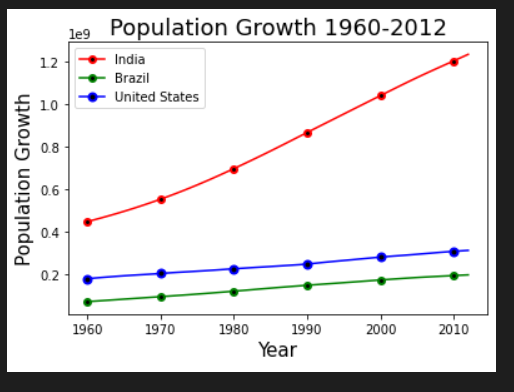
1. To plot the graph between Year and population growth for India, Brazil, and the United States. Decided to keep X-axis as Year from 19600 to 2012 and Y-axis as population growth.
2. For this, we save the Year and population growth for India, Brazil, and the United States

in x, y, y1, and y2 respectively

1. Plot the graph by using plt.plot() method of matplotlib library. (used markevery=10 to put marker for every 10 years)

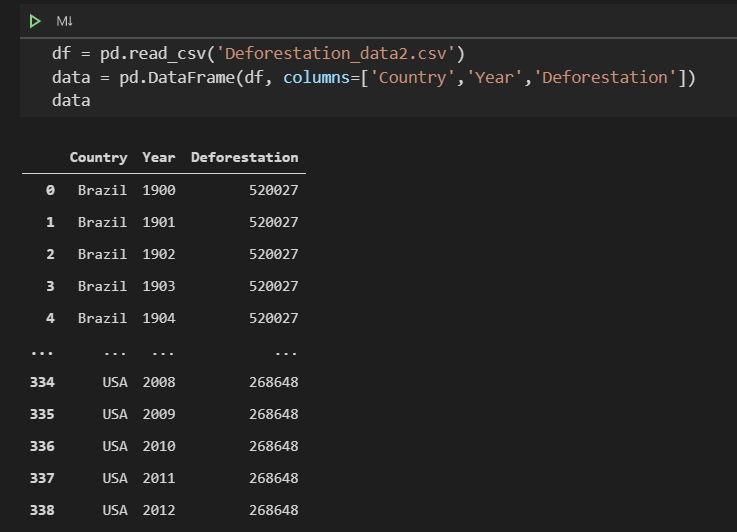


Output:

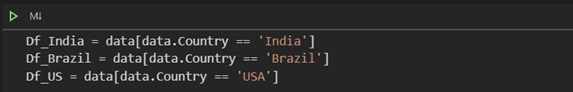


**Deforestation rate dataset visualization**

1. Read the csv file ‘Deforestation\_data2.csv’containing the deforestation data for 3 countries by calling read\_csv function.
2. Created the data frame named as ‘data’ for the dataset visualization.



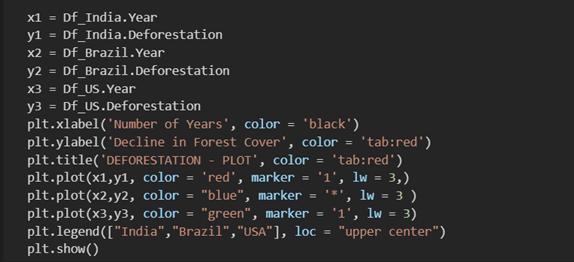
1. Created three different data sets named Df\_India, Df\_Brazil, Df\_US from the main data set, based on the Country column.



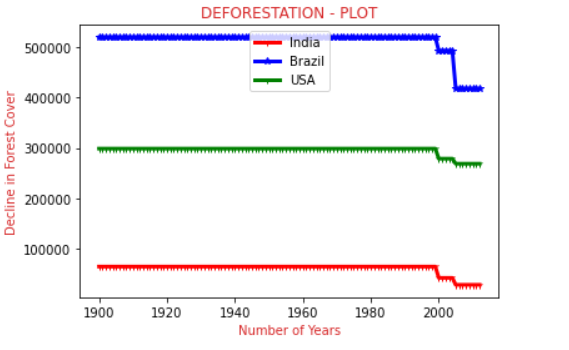
1. To plot the graph between year and deforestation for India, Brazil, and the United States, defined the X-axis as year from 1900 to 2012 and Y-axis as decline in forest cover.
2. For this, we saved the year and deforestation for India, Brazil, and the United States.

in x1, y1, x2, y2, x3 and y3 respectively

1. Finally, plotted the graph by using plt.plot() method of matplotlib library for India, Brazil, and the United States.
2. To differentiate three different graphs, we used the line style for graph1 as red, marker = 1; graph2 as green, marker = \* and graph3 as blue, marker = 1 and kept the location as upper center

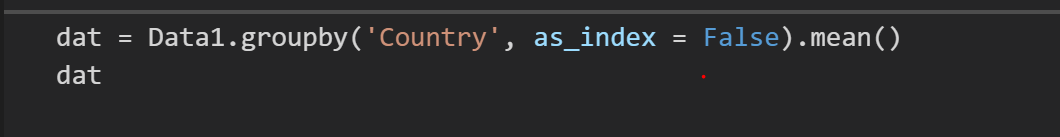


**Output:**

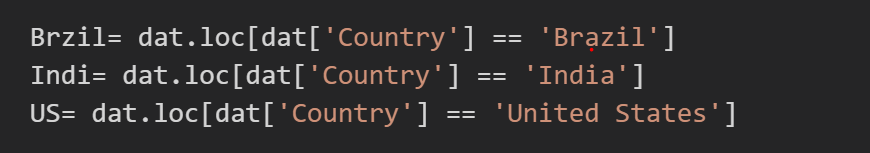


**Wrangled dataset visualization:**

1. **First, we would be calculating the mean for each of the three columns (CO2 emission, population growth, deforestation) grouped by country**

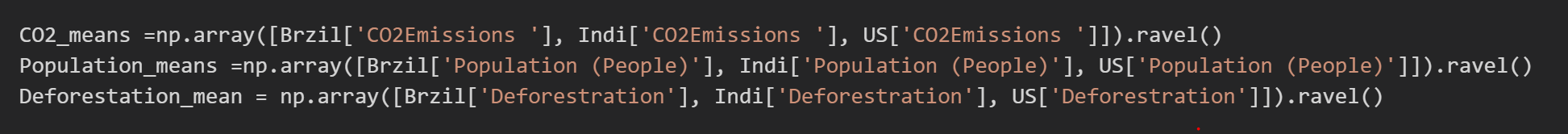


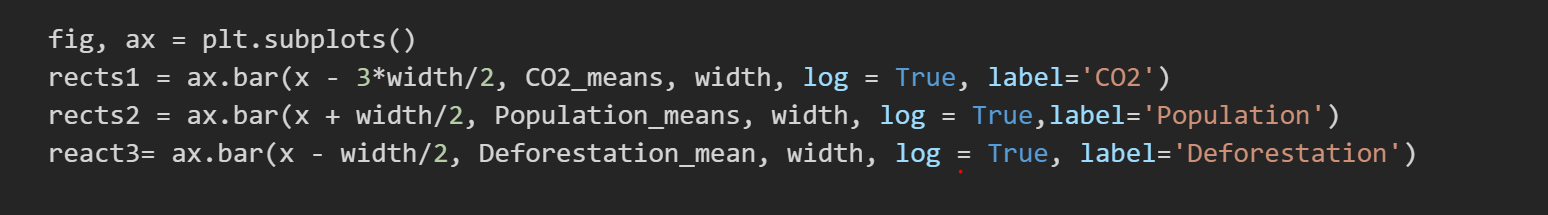
1. Second, we get the mean data of the three countries separated

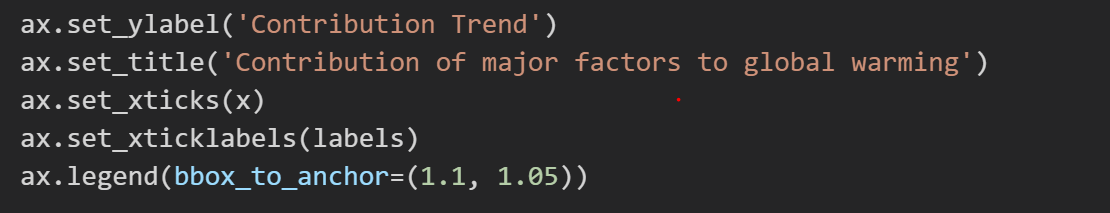


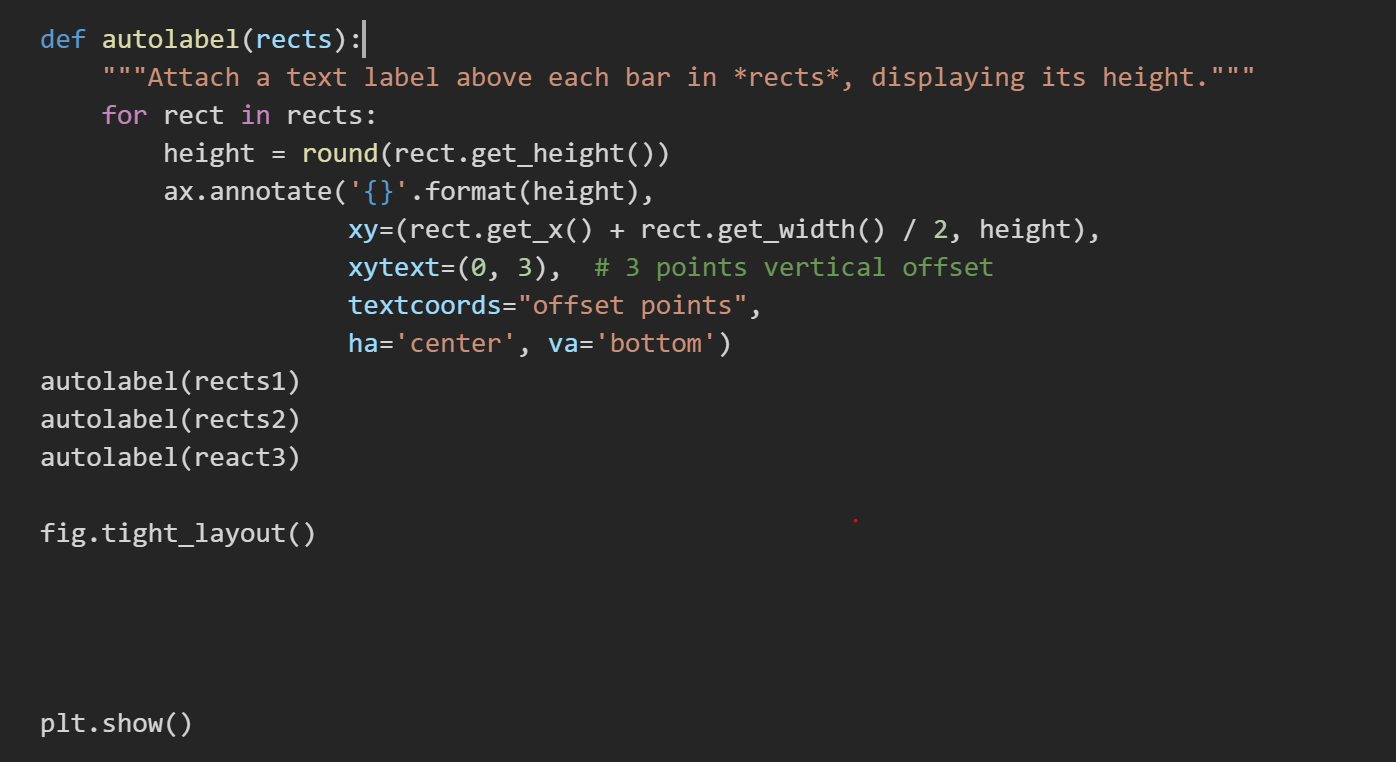
1. Second, we extract the year from the date



1. Thirdly, create three arrays, with the mean values of each of the factors. .ravel() is used to create a contiguous flattened array.
2. Using plotly.bar we plot a graph for the three countries. We can specify the bar width and also the scale we want to plot the graph at. We are using logarithmic scale for this plot.



1. We customized the design of the graph using yalbel, x\_tics and then we named the yaxis as Contribution Trend and also set legend for the graph.
2. For each plotted bar graph for each of the factors of three countries, we can attach the its value using autolabel function and finally plot the graph using matplotlib’s show method.



**Output:**

