Report

The project starts with importing required libraries like numpy, pandas, matplotlib, sklearn etc. **dataFrame()** and **randomYearsFromDF()**, which are used to read data from a CSV file and select random rows from the resulting dataframe.

The **dataFrame()** function reads data from a CSV file named "LifeExpectancy.csv" using the Pandas library's **read\_csv()** function, with the header parameter set to 0. This indicates that the first row of the CSV file contains the column names. The function then returns the resulting Pandas dataframe.

The **randomYearsFromDF()** function calls **dataFrame()** to retrieve the dataframe, and then selects a random row from the dataframe where the 'Year' column matches a random integer between 2000 and 2015 (inclusive). The function then returns the resulting row as a Pandas dataframe. Function named **split\_test\_data(type)** that splits data into train and test datasets for three input dataframes: **a**, **b**, and **c**. The function randomly selects rows from the **randomYearsFromDF()** function, which is assumed to return dataframes.

The **train\_test\_split()** function from the Scikit-learn library is used to split each of these dataframes into train and test datasets. The resulting train and test datasets are then returned based on the value of the **type** parameter, which is either "train" or "test". If **type** is "train", the function returns the train datasets for **a**, **b**, and **c**. If **type** is "test", the function returns the test datasets for **a**, **b**, and **c**.

The function two\_a is get test values from split\_test\_data, with zip method returns ‘Size of the Data Frame’ for each element in the lists\_of\_data\_sets.

The **two\_b()** function **two\_a()**. It then calculates and prints the mean, median, and standard deviation of the "Life expectancy" column for each of the three dataframes returned by **two\_a()**. The **zip()** function is used to iterate over the three dataframes and assign them the labels 'A', 'B', and 'C'.

The **mean()**, **median()**, and **stdev()** functions from the **statistics** library are used to calculate the mean, median, and standard deviation of the "Life expectancy" column for each of the dataframes.

Function named **two()** that calls another function named **two\_b()**. It then iterates over the three dataframes returned by **two\_b()**, retrieves the "Life expectancy", "Country", and "Year" columns for each dataframe, sorts the values in descending order based on the "Year" column, and returns the top three values for each dataframe.

The function **three()** it creates three linear regression models using the **linear\_model.LinearRegression()** function from scikit-learn library. The models are fitted on different independent variables, namely "GDP", "Total expenditure", and "Alcohol". The code returns the three fitted models if **type** is equal to "model", and it returns the two dataframes **a** and **b** if **type** is equal to "data".

The **four()** first calls the **three()** function with argument **'model'** to retrieve the **LinearRegression** objects for GDP, total expenditure, and alcohol. Then, it prints the coefficients and intercepts for each regression.

Next, it calls the **three()** function with argument **'data'** to retrieve the training data sets for GDP and total expenditure. It then creates a scatter plot of the GDP data points and a red line representing the regression line calculated by the linear regression model. The plot also displays the equation of the regression line and the coefficients and intercepts printed earlier. Finally, it displays the plot using **matplotlib**.

The **five()** function is calculating the mean absolute error and standard deviation for the predictions made by the linear regression models created in the **three()** function on the test data. First it retrieves the trained models and the test data from **three('model')** and **three('data')**, respectively. Then, it uses the trained models to predict the output values for the test data. After that, it calculates the mean absolute error for each of the three features (GDP, Total expenditure, and Alcohol) using the **mean\_absolute\_error** function from the **sklearn.metrics** module. Finally, it calculates the average error and standard deviation of the errors across the three features and prints them out.

Output is:

Text

Description automatically generated

Chart

Description automatically generated