

Report

1.

- >Pycountry library is used to map the country/region to their corresponding continents.
- > ***pip install pycountry*** before running the program and import it in the current module.
- > Install and import **pycountry_convert** for country code to continent code conversion.
- >Data cleaning has been performed by removing the spelling mistakes from various region/country names that has to be matched with name in pycountry library.
- >Input file is read as a pandas dataframe
- >refer_country contains all the countries that are included in the library.
- >result=[]--List to store the continent code that has been linked from inputfile country/region.
- >note_index=[] --list to store the index of countries whose continent code are not available in pycountry library
- >not_available=[]--list to store the name of countries whose continent code.
- >ref_list=[]--list to store actual pycountry countries
- >now insert result list at index present in note_index as “No continents”
- >convert the data frame to .csv file.

Outpt file:

Estimated Continent	1979	1980	1981	1982	1983	1984	1985	1986	1987	1988	1989	1990	1991	1992	1993
0 Abkhazia No continent															
1 Afghanistan AS															
2 Akrotiri ar No continent															
3 Albania EU															
4 Algeria AF												0.06	0.06	0.06	0.06
5 American OC															
6 Andorra EU															
7 Angola AF	0.026528											0.5	0.8	1	1.2
8 Anguilla NA															
9 Antigua ar NA															
10 Argentina SA	0.020508											0.3	0.3	0.3	0.3
11 Armenia AS												0.06	0.06	0.06	0.06
12 Aruba NA															
13 Australia OC					0.012683							0.1	0.1	0.1	0.1
14 Austria EU			0.010188									0.06	0.06	0.06	0.06
15 Azerbaijan AS												0.06	0.06	0.06	0.06
16 Bahamas NA	0.01											3.6	3.8	3.9	3.9
17 Bahrain AS															
18 Bangladesh AS												0.06	0.06	0.06	0.06
19 Barbados NA	0.027676											0.06	0.06	0.1	0.1
20 Belarus EU												0.06	0.06	0.06	0.06
21 Belgium EU				0.014147								0.06	0.1	0.1	0.1
22 Belize NA												0.5	0.7	1	1.2
23 Benin AF						0.011876						0.2	0.4	0.5	0.7

2.

->matplotlib has been used for plotting the graphs.

->grouped the data frame by continent column and got the maximum and minimum mean by performing .max() and .min() on mean column.

->used subplot to plot max and min mean graph.

->For overlaid bar graph used plot.bar(x,y,color=),where x and y are continent and mean respectively.

->concatenated max, min and actual data frame where the resultant is frame with duplicate rows of max and min.

->Now removed the duplicate copies from the frame where that results in a frame with no max and min mean rows.

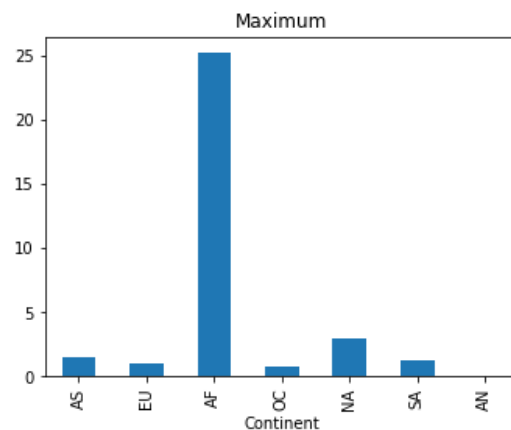
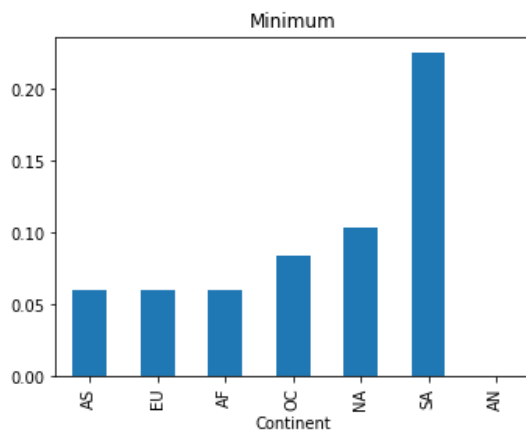
-> randomly selected the value from the resultant frame from previous step and used for plotting.

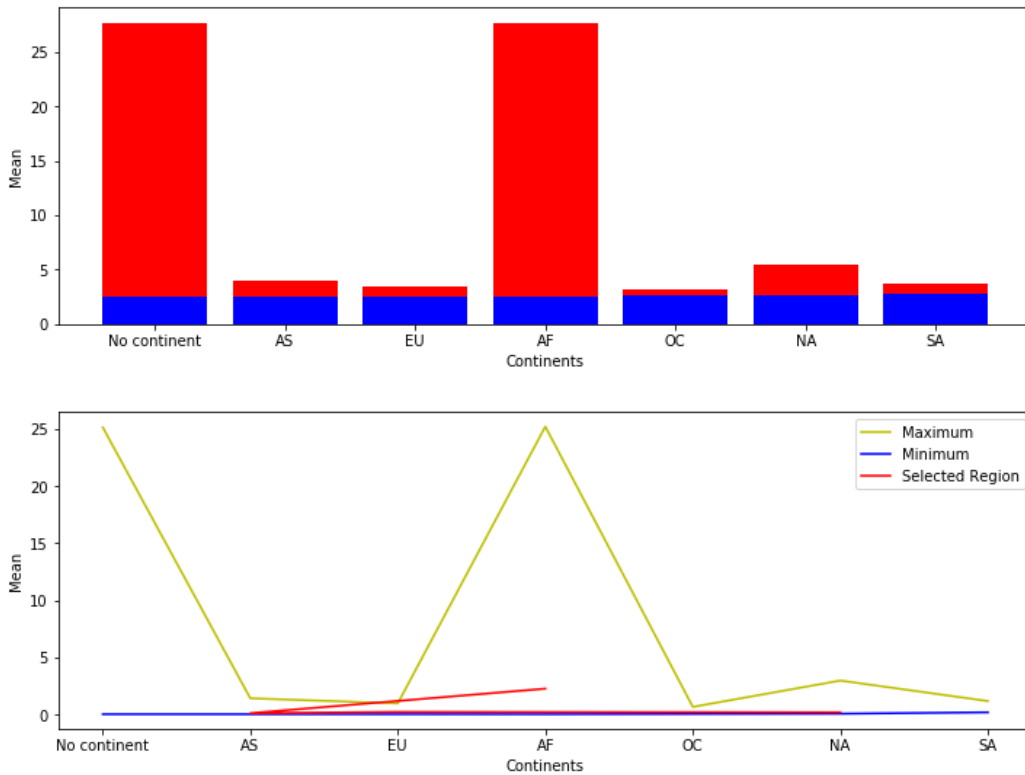
(Here when ever we execute the program new values are choosed randomly which results in difference of line shape in the graph)

Output:

```
Continent
AS      1.450000
EU      1.008333
AF     25.208333
OC      0.700000
NA      3.000000
SA      1.208333
AN         NaN
Name: Avr 2000_2011, dtype: float64
Continent
AS      0.060000
EU      0.060000
AF      0.060000
OC      0.083333
NA      0.103333
SA      0.225000
AN         NaN
Name: Avr 2000_2011, dtype: float64
```

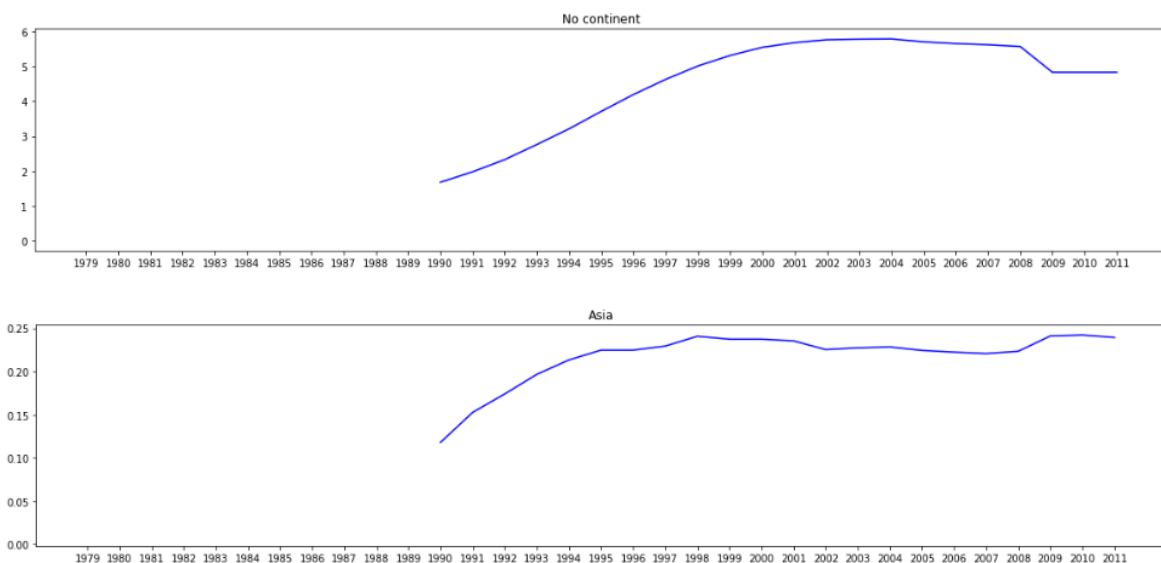
```
-----> plt.subplot(2,1,1)
```

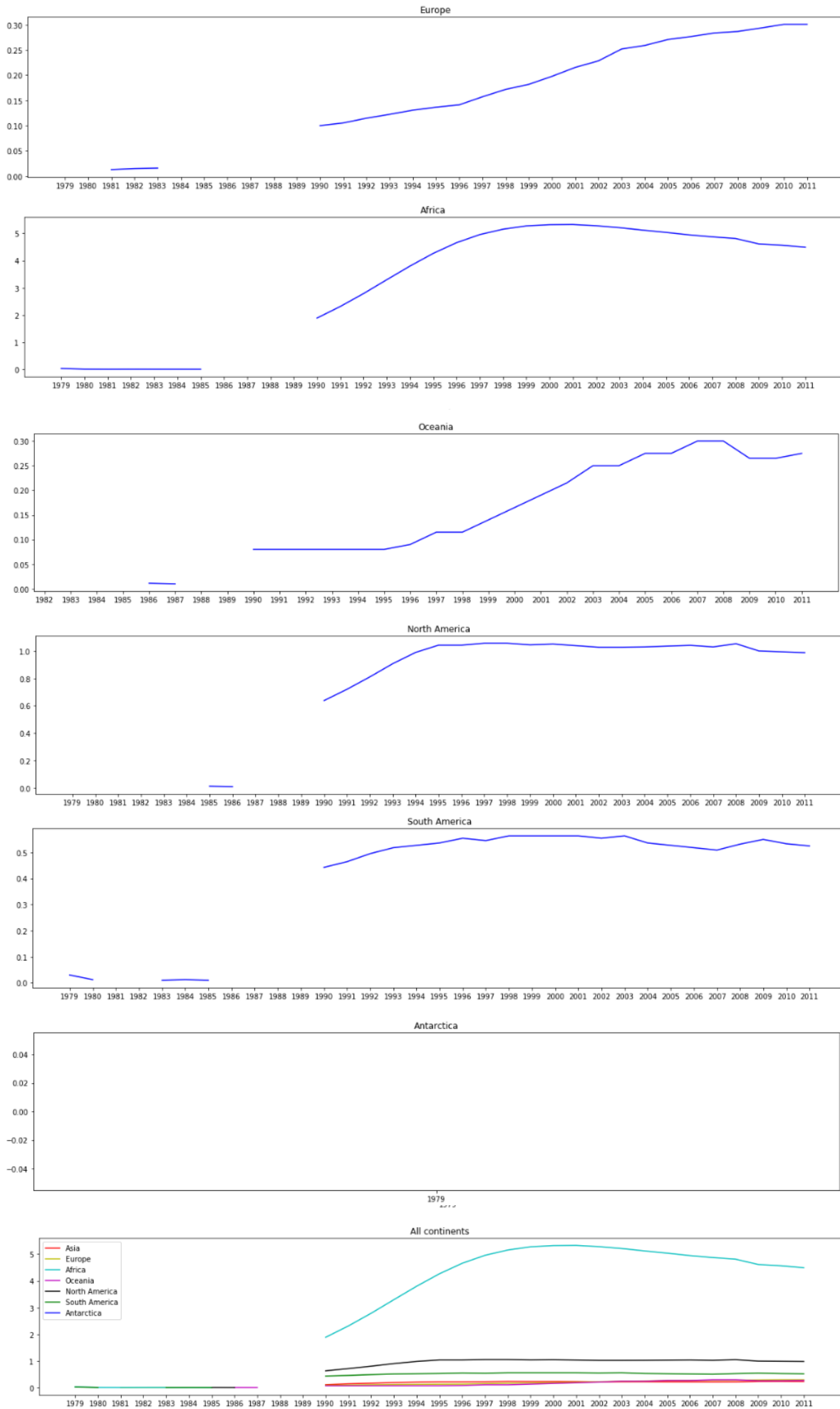




- 3.
- >Here a frame with continent and all years columns has been considered.
 - >groupby is done on the continents.
 - >Transpose is done for the frame to access the year values in x-axis.
 - >Used `.plot(x,y,color=')` where x and y are years (which is the index of the frame) and continents respectively.
 - >For the overlaid graph all the `.plots` need to be taken in single `plt.figure()`

Output:





4.

->here groupby is done on 'continent', 'Estimated HIV Prevalence% - (Ages 15-49)'

->Used `plt.scatter(x,y,data=c,s=,alpha=)` to scatter the points, where x is continents and y are mean values.

->Used `colors=rng.rand(274)` to display 274 different colors for each row and s denotes the size of the scatter point.

Output:

