Importing Libraries

In [11]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.pyplot as plt
import pandas as pnd
import numpy as nmp
import os
os.listdir()
from pandas.plotting import scatter_matrix
from pandas import DataFrame, read_csv
```

Add one column as “continent” in the dataset and label each country/regionin the dataset to an appropriate continent such as “Europe”, “Asia”,“Africa”, “North America”, “South America”, “Australia”, or “Antarctica”. Explain how do you validate the correctness of your labelling. Output the updated dataset as a new CSV file.

defining countries to their continents

In [49]:
```
df = ('../Data/hivdata.xlsx')
data = pd.read_excel ('../data/hivdata.xlsx')
data.head()
data.rename(columns={"Estimated HIV Prevalence% - (Ages 15-49)":"country"},inplace=True
data.head()
Africa = ('Algeria','Angola','Benin','Botswana','Burkina','Burundi','Cameroon','Cape Ve
Asia = ('Afghanistan','Bahrain','Bangladesh','Bhutan','Brunei','Burma (Myanmar)','Cambo
Europe = ('Albania','Andorra','Armenia','Austria','Azerbaijan','Belarus','Belgium','Bos
North_America = ('Antigua and Barbuda','Bahamas','Barbados','Belize','Canada','Costa Ri
South_America = ('Argentina','Bolivia','Brazil','Chile','Colombia','Ecuador','Guyana','
Australia = ('Australia','Fiji','Kiribati','Marshall Islands','Micronesia','Nauru','New

def  getConti(country):

    if country in Africa:
        return "Africa"
    elif country in Asia:
        return "Asia"
    elif country in Europe:
        return "Europe"
    elif country in North_America:
        return "North America"
    elif country in South_America:
        return "South America"
    elif country in Australia:
        return "Australia"
    else:
        return "Other"
data['continents']=country.apply(getConti)

data
```

Out[49]:

	country	1979	1980	1981	1982	1983	1984	1985	1986	1987	...	2003	2004	2005	20(
0	Abkhazia	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	Na
1	Afghanistan	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	Na
2	Akrotiri and Dhekelia	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	Na
3	Albania	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	Na
4	Algeria	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	0.06	0.1	0.1	(
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
270	Bonaire	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	Na
271	Sark	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	Na
272	Chinese Taipei	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	Na
273	Saint Eustatius	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	Na
274	Saba	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	Na

275 rows × 35 columns

In [18]:
```
s=data
s.to_csv("redefined_data.csv")
```

In [57]:
```
data[["country","continents"]].head(20)
```

Out[57]:

	country	continents
0	Abkhazia	Other
1	Afghanistan	Asia
2	Akrotiri and Dhekelia	Other
3	Albania	Europe
4	Algeria	Africa
5	American Samoa	Other
6	Andorra	Europe
7	Angola	Africa
8	Anguilla	Other
9	Antigua and Barbuda	North America
10	Argentina	South America
11	Armenia	Europe
12	Aruba	Other
13	Australia	Australia

	country	continents
14	Austria	Europe
15	Azerbaijan	Europe
16	Bahamas	North America
17	Bahrain	Asia
18	Bangladesh	Asia
19	Barbados	North America

In [50]:
```
df1=data.drop(columns=data.iloc[:,1:-13])
df1
```

Out[50]:

	country	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	contin
0	Abkhazia	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	O
1	Afghanistan	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0.06	0.06	0.06	
2	Akrotiri and Dhekelia	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	O
3	Albania	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Eur
4	Algeria	0.06	0.06	0.06	0.06	0.1	0.1	0.1	0.1	0.1	NaN	NaN	NaN	A
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
270	Bonaire	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	O
271	Sark	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	O
272	Chinese Taipei	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	O
273	Saint Eustatius	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	O
274	Saba	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	O

275 rows × 14 columns

In [51]:
```
df1['mean'] = df1.mean(axis=1)
```

In [52]:
```
df1
```

Out[52]:

	country	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	contin
0	Abkhazia	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	O
1	Afghanistan	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0.06	0.06	0.06	
2	Akrotiri and Dhekelia	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	O
3	Albania	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Eur

	country	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	contin
4	Algeria	0.06	0.06	0.06	0.06	0.1	0.1	0.1	0.1	0.1	NaN	NaN	NaN	A
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
270	Bonaire	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	O
271	Sark	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	O
272	Chinese Taipei	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	O
273	Saint Eustatius	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	O
274	Saba	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	O

275 rows × 15 columns

In [73]:
```
df2=pd.DataFrame(df1.groupby(["continents","country"])["mean"].max())
df2.head(70)
```

Out[73]:

		mean
continents	**country**	
Africa	**Algeria**	0.082222
	Angola	1.958333
	Benin	1.275000
	Botswana	25.208333
	Burundi	3.566667
...	**...**	...
Asia	**Kuwait**	NaN
	Lebanon	0.141667
	Malaysia	0.433333
	Maldives	0.060000
	Mongolia	0.060000

70 rows × 1 columns

Write a Python program to find the country/region in each continent thathas the highest average HIV estimated prevalence of people ages from 15 to 49 of from year 2000 to 2011.

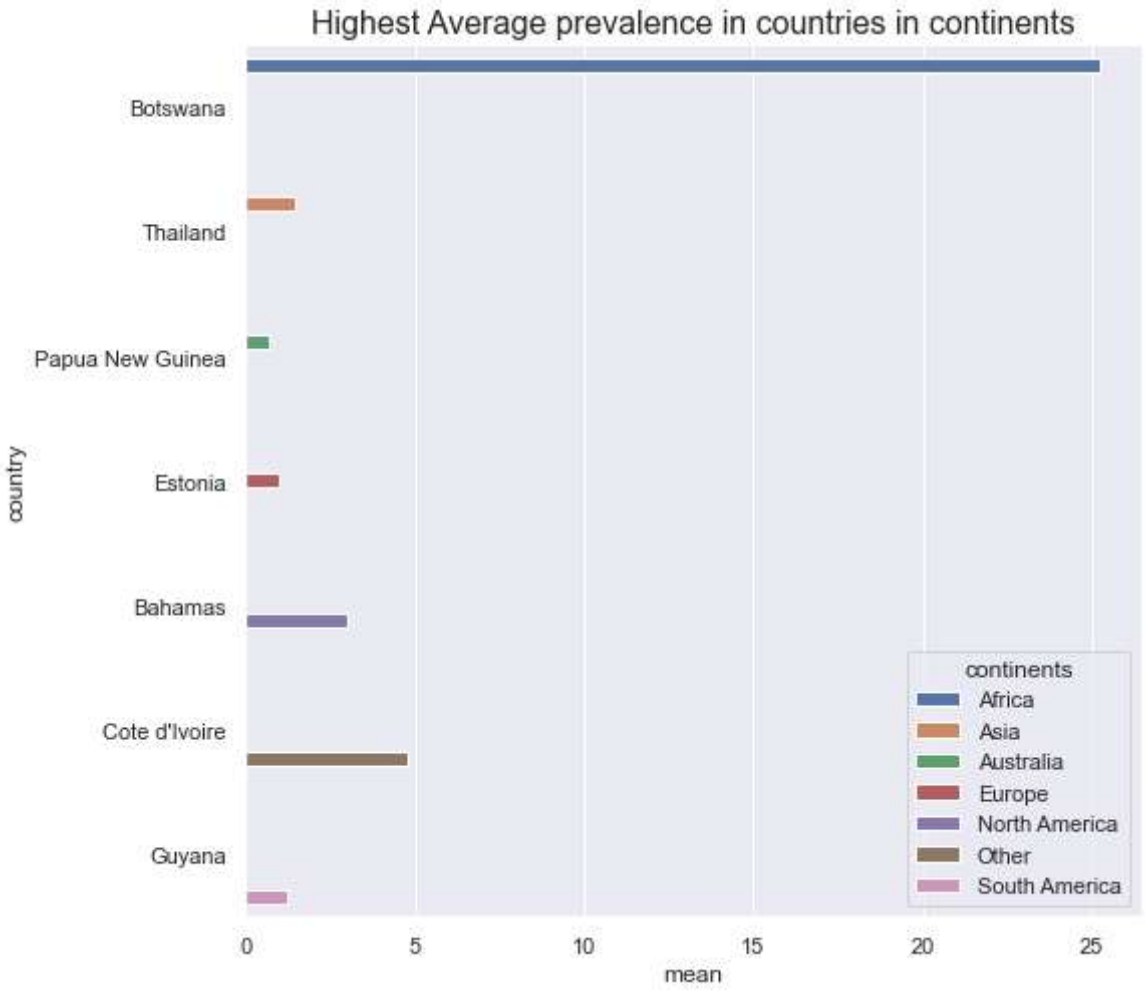
In [80]:
```
Highestaverage = df2.groupby('continents')['mean'].idxmax()
Highestaverage=df2.loc[Highestaverage]
Highestaverage=Highestaverage.reset_index()
Highestaverage
```

Out[80]:

	continents	country	mean
0	Africa	Botswana	25.208333
1	Asia	Thailand	1.450000
2	Australia	Papua New Guinea	0.700000
3	Europe	Estonia	1.008333
4	North America	Bahamas	3.000000
5	Other	Cote d'Ivoire	4.758333
6	South America	Guyana	1.208333

Create a bar chart to show the highest average HIV estimated prevalence of people ages from 15 to 49 of from year 2000 to 2011 in each continent

In [81]:
```
import matplotlib.pyplot as plt
import seaborn as sns
sns.set(style="ticks")
sns.set(rc={'figure.figsize':(8,8)})
plt.title("Highest Average prevalence in countries in continents",size=16)
sns.barplot(x="mean",y="country",hue="continents",data=Highestaverage)
plt.show()
```



Find the country/region in each continent that has the lowest average HIV estimated prevalence of people ages from 15 to 49 of from year 2000 to 2011.

In [88]:
```
minimum=pd.DataFrame(df2.groupby(["continents","country"])["mean"].min())
minimum
```

Out[88]:

		mean
continents	**country**	
Africa	**Algeria**	0.082222
	Angola	1.958333
	Benin	1.275000
	Botswana	25.208333
	Burundi	3.566667
...	**...**	...
South America	**Paraguay**	0.300000
	Peru	0.441667
	Suriname	1.033333
	Uruguay	0.491667
	Venezuela	0.500000

275 rows × 1 columns

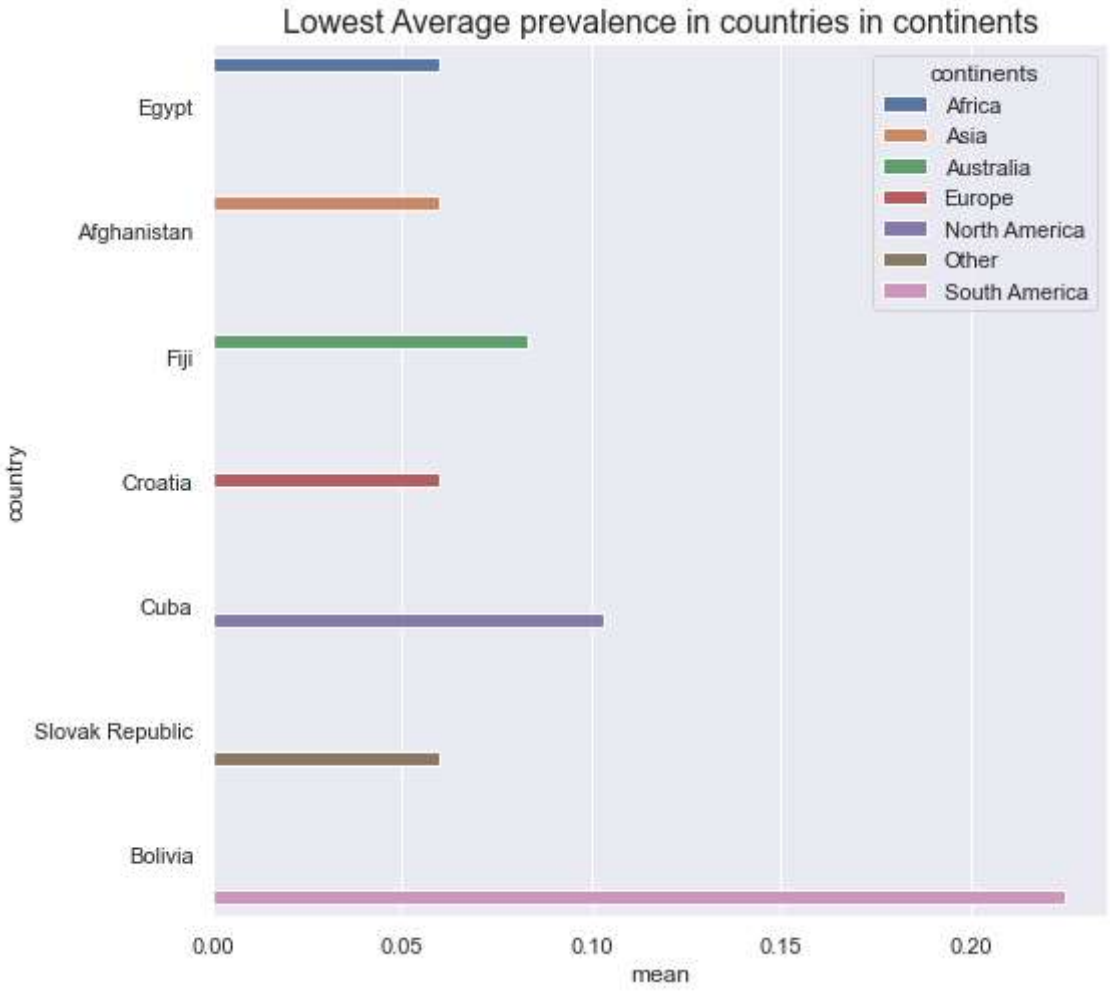
Create a bar chart to show the lowest average HIV estimated prevalence of people ages from 15 to 49 of from year 2000 to 2011 in each continent

In [90]:
```
lowestAverage=df2.groupby('continents')['mean'].idxmin()
lowestAverage=minimum.loc[lowestAverage]
lowestAverage=lowestAverage.reset_index()
lowestAverage
```

Out[90]:

	continents	country	mean
0	Africa	Egypt	0.060000
1	Asia	Afghanistan	0.060000
2	Australia	Fiji	0.083333
3	Europe	Croatia	0.060000
4	North America	Cuba	0.103333
5	Other	Slovak Republic	0.060000
6	South America	Bolivia	0.225000

In [91]:
```
import matplotlib.pyplot as plt
import seaborn as sns
sns.set(style="ticks")
sns.set(rc={'figure.figsize':(8,8)})
plt.title("Lowest Average prevalence in countries in continents",size=16)
sns.barplot(x="mean", y="country",hue="continents",data=lowestAverage)
plt.show()
```

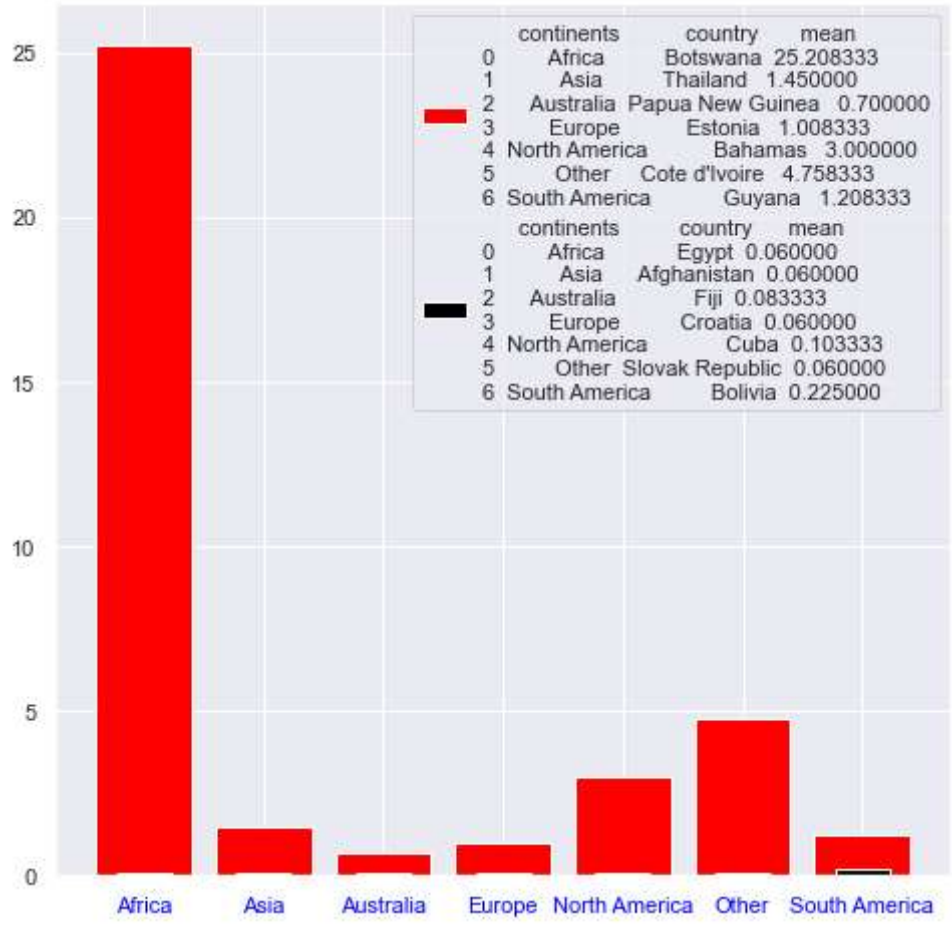


Create an overlaid bar chart to show the highest and lowest average HIV estimated prevalence of people ages from 15 to 49 of from year 2000 to 2011 in each continen

In [95]:
```
con=Highestaverage["continents"]
continents=list(con)
```

In [98]:
```
plt.bar(continents, Highestaverage["mean"] ,label=Highestaverage[["continents","country
plt.bar(continents,lowestAverage["mean"],width=0.45,label=lowestAverage[["continents","
plt.xticks(color='blue')
plt.legend()
```

Out[98]:
```
<matplotlib.legend.Legend at 0x2a1c4998fd0>
```



Select acountry/region that is different from the average highest or lowest HIV estimated prevalence of people ages from 15 to 49 from year 2000 to 2011 from each continent, then create an overlaid line chart for the selected country/region, the average highest and lowest HIV estimated prevalence of people ages from 15 to 49 from year 2000 to 2011 for each continent

In [121…
```
data["mean"]=data.iloc[:, 1:].mean(axis=1)
```

In [122…
```
k=pd.DataFrame()
k["year"]=data.iloc[:,1:].mean(axis=0)
k
```

Out[122…

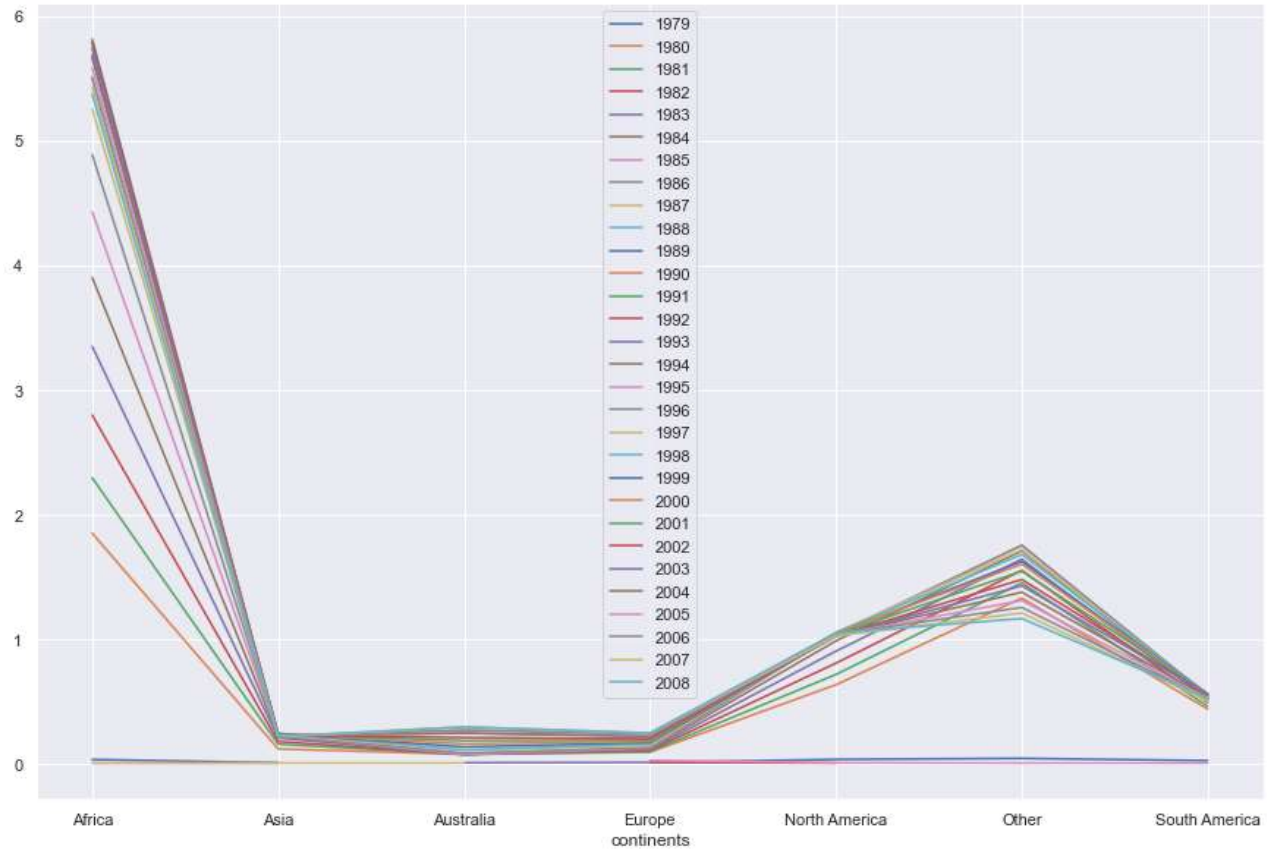
	year
1979	0.034125
1980	0.013259
1981	0.011890
1982	0.013604
1983	0.013706
1984	0.011703

	year
1985	0.014788
1986	0.010441
1987	0.010250
1988	NaN
1989	NaN
1990	0.795685
1991	0.955959
1992	1.132534
1993	1.323151
1994	1.508630
1995	1.680137
1996	1.820822
1997	1.934658
1998	2.017123
1999	2.066712
2000	2.095068
2001	2.105616
2002	2.094932
2003	2.081918
2004	2.055137
2005	2.028973
2006	2.000479
2007	1.977123
2008	1.962260
2009	1.929247
2010	1.926207
2011	1.907483
mean	1.673950

Create an overlaid line chart for all continents to show their changes of the average HIV estimated prevalence from 1979 to 2011

In [108…
```
yearlyaverage.plot(x="continents", y=[i for i in range(1979,2009)], figsize=(15,10), gr
```

Out[108… <AxesSubplot:xlabel='continents'>

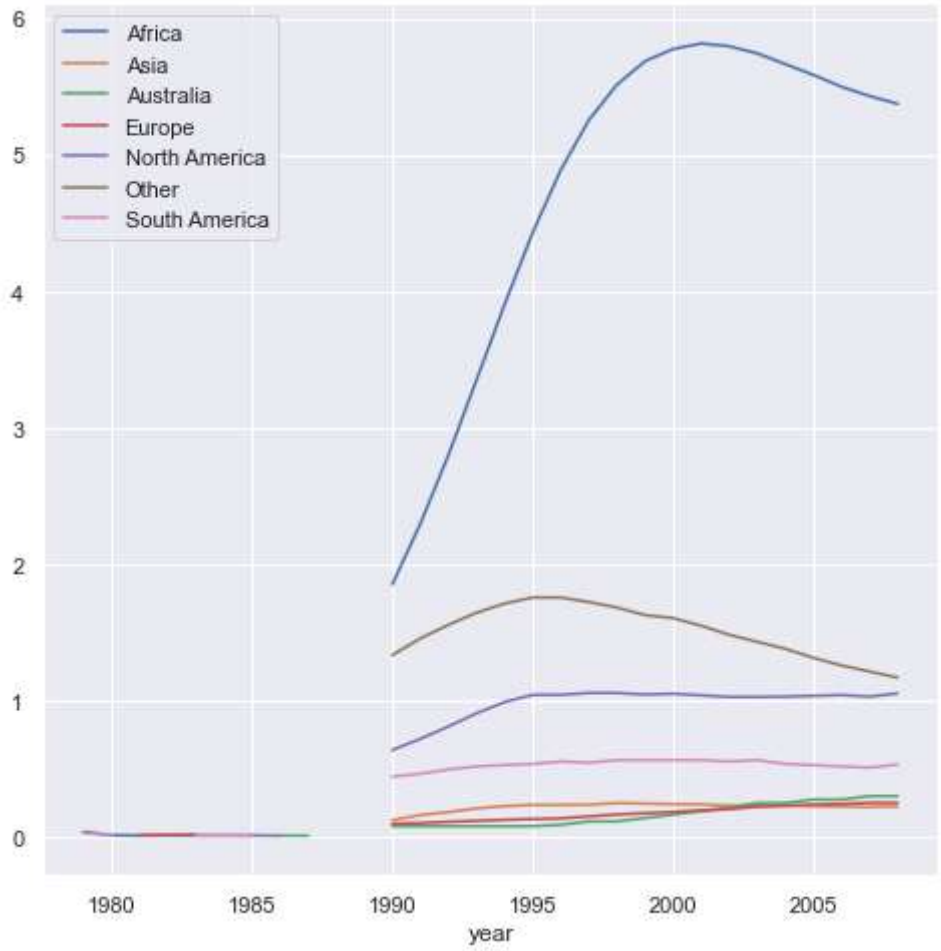


Based on the calculation, create a line chart for each continent to show the changes of the average HIV estimated prevalence from 1979 to 2011

In [109…

```
s=yearlyaverage.T
s.rename(columns=s.iloc[0], inplace = True)
s=s.reset_index()
s.drop(0,inplace=True)
s.rename(columns={"index":"year"},inplace=True)
s.plot(x="year")
```

Out[109… <AxesSubplot:xlabel='year'>



Create two scatter plots to show the data (i.e. each country/region) in year 1990 and year 2010, respectively. The vertical axis in the scatter plot is the HIV estimated prevalence, and the horizontal axis is the corresponding year average HIV estimated prevalence in each continent, which you calculated above. Using different color to show data from different continent

In [111…
```
yearlyaverage
```

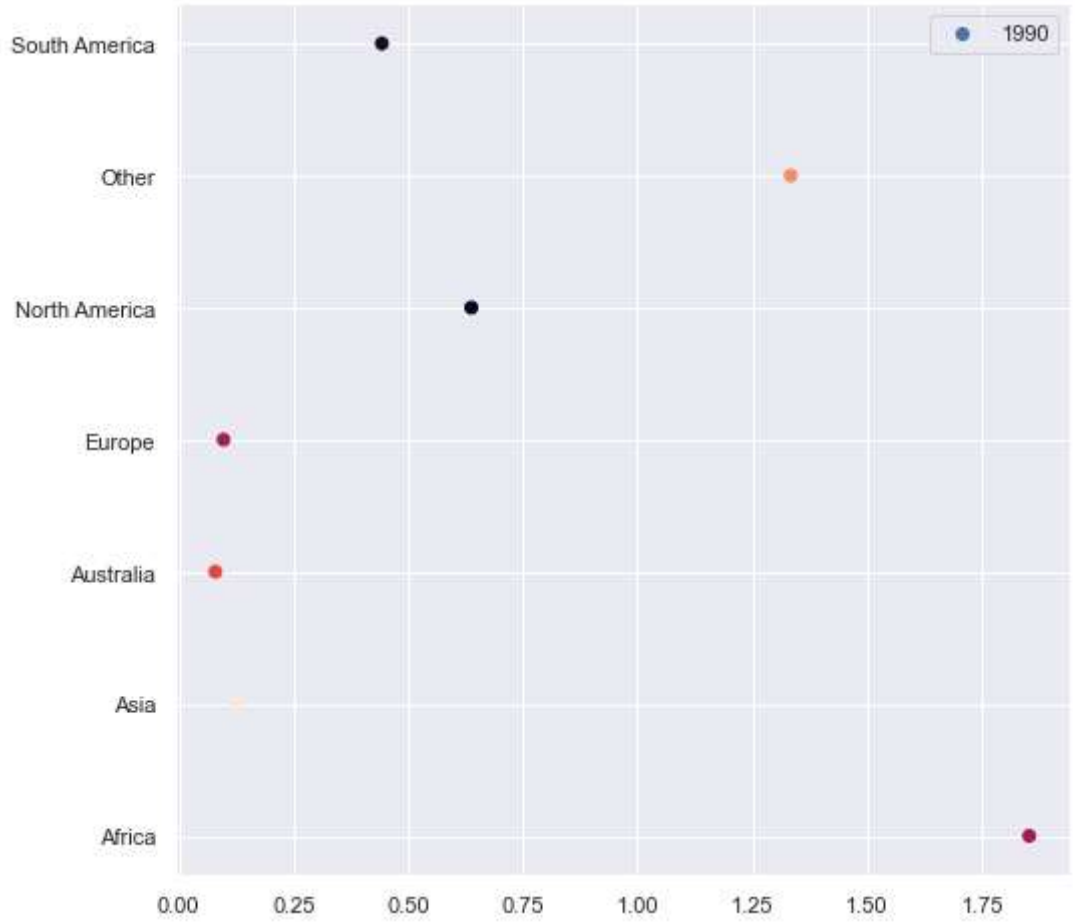
Out[111…

	continents	1979	1980	1981	1982	1983	1984	1985	1986	1987
0	Africa	0.038769	0.013923	0.011185	0.011773	0.011911	0.011477	0.010948	NaN	0.010400
1	Asia	0.012168	NaN	NaN	NaN	NaN	NaN	0.010000	NaN	0.010175
2	Australia	NaN	NaN	NaN	NaN	0.012683	NaN	NaN	0.011372	0.010175
3	Europe	0.014247	NaN	0.012948	0.014927	0.015850	NaN	0.032011	NaN	NaN
4	North America	0.039628	NaN	NaN	0.010653	NaN	NaN	0.012270	0.009510	NaN
5	Other	0.047726	NaN	NaN	NaN	NaN	NaN	0.010000	NaN	NaN
6	South America	0.029865	0.011931	NaN	NaN	0.009743	0.012153	0.009689	NaN	NaN

7 rows × 31 columns

In [114…
```
colr=np.random.RandomState(0)
colors=colr.rand(7)
plt.scatter(yearlyaverage[1990],"continents",data=yearlyaverage,c=colors)
plt.legend(["1990"])
```

Out[114… <matplotlib.legend.Legend at 0x2a1c4af7e80>



In [115…
```
v=data.iloc[:,[0,12,32]]
v.head()
```

Out[115…

	country	1990	2010
0	Abkhazia	NaN	NaN
1	Afghanistan	NaN	0.06
2	Akrotiri and Dhekelia	NaN	NaN
3	Albania	NaN	NaN
4	Algeria	0.06	NaN

In [ ]:
```
v.plot.scatter(x='2010', y='newvariable', title= "2010 Actual vs Average");

plot.show(block=True);
```

In [ ]:
```
```