

Hello, welcome to another lesson. This lesson is in continuation of the series on using Python and its vast Libraries [Numpy, Pandas, Matplotlib etc] for Data Analysis Tasks.

Most importantly, it's about sharing my growth process on this career path, and also to encourage you on what you're doing. Be encouraged and keep doing it, even if it means doing it poorly till you gain mastery of it

First, you need to import the necessary Libraries required for task execution, then remember to add "%matplotlib inline" for display of plots.

This dataset is another excel file downloaded from the website of "WHO". It's readily available for the public access

Import the file and read the first 5 lines of the DataFrame

```
In [1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
```

```
In [2]: dataset = pd.read_excel("TB Outcomes Completed.xlsx", sheet_name = 0,
index_col = 0)
dataset.head()
```

Out[2]:

	iso2	iso3	iso_numeric	g_whoregion	year	rep_meth	new_sp_coh	new_sp_cur	new
country									
Afghanistan	AF	AFG	4	EMR	1994	100	10863.51162	7887.186906	9
Afghanistan	AF	AFG	4	EMR	1995	100	10863.51162	7887.186906	9
Afghanistan	AF	AFG	4	EMR	1996	100	10863.51162	7887.186906	9
Afghanistan	AF	AFG	4	EMR	1997	100	2001.00000	786.000000	1
Afghanistan	AF	AFG	4	EMR	1998	100	2913.00000	772.000000	1

5 rows × 71 columns

Next, you should familiarize yourself with the nature of the Dataset. This would give you an idea of what the dataset looks like

```
In [3]: dataset.shape
```

```
Out[3]: (5118, 71)
```

```
In [4]: dataset.dtypes
```

```
Out[4]: iso2          object
iso3          object
iso_numeric    int64
g_whoregion    object
year          int64
...
xdr_coh        float64
xdr_succ        float64
xdr_fail        float64
xdr_died        float64
xdr_lost        float64
Length: 71, dtype: object
```

```
In [5]: type(dataset)
```

```
Out[5]: pandas.core.frame.DataFrame
```

```
In [6]: dataset.shape
```

```
Out[6]: (5118, 71)
```

```
In [7]: dataset.size
```

```
Out[7]: 363378
```

```
In [8]: dataset.index
```

```
Out[8]: Index(['Afghanistan', 'Afghanistan', 'Afghanistan', 'Afghanistan',
               'Afghanistan', 'Afghanistan', 'Afghanistan', 'Afghanistan',
               'Afghanistan', 'Afghanistan',
               ...
               'Zimbabwe', 'Zimbabwe', 'Zimbabwe', 'Zimbabwe', 'Zimbabwe', 'Zi
mbabwe',
               'Zimbabwe', 'Zimbabwe', 'Zimbabwe', 'Zimbabwe'],
              dtype='object', name='country', length=5118)
```

```
In [9]: dataset.columns
```

```
Out[9]: Index(['iso2', 'iso3', 'iso_numeric', 'g_whoregion', 'year', 'rep_meth',
              'new_sp_coh', 'new_sp_cur', 'new_sp_cmplt', 'new_sp_died',
              'new_sp_fail', 'new_sp_def', 'c_new_sp_tsr', 'new_snep_coh',
              'new_snep_cmplt', 'new_snep_died', 'new_snep_fail', 'new_snep_def',
              'c_new_snep_tsr', 'ret_coh', 'ret_cur', 'ret_cmplt', 'ret_died',
              'ret_fail', 'ret_def', 'hiv_new_sp_coh', 'hiv_new_sp_cur',
              'hiv_new_sp_cmplt', 'hiv_new_sp_died', 'hiv_new_sp_fail',
              'hiv_new_sp_def', 'hiv_new_snep_coh', 'hiv_new_snep_cmplt',
              'hiv_new_snep_died', 'hiv_new_snep_fail', 'hiv_new_snep_def',
              'hiv_ret_coh', 'hiv_ret_cur', 'hiv_ret_cmplt', 'hiv_ret_died',
              'hiv_ret_fail', 'hiv_ret_def', 'rel_with_new_flg', 'newrel_coh',
              'newrel_succ', 'newrel_fail', 'newrel_died', 'newrel_lost', 'c_newrel_tsr',
              'ret_nrel_coh', 'ret_nrel_succ', 'ret_nrel_fail', 'ret_nrel_died',
              'ret_nrel_lost', 'c_ret_tsr', 'tbhiv_coh', 'tbhiv_succ', 'tbhiv_fail',
              'tbhiv_died', 'tbhiv_lost', 'c_tbhiv_tsr', 'mdr_coh', 'mdr_succ',
              'mdr_fail', 'mdr_died', 'mdr_lost', 'xdr_coh', 'xdr_succ', 'xdr_fail',
              'xdr_died', 'xdr_lost'],
              dtype='object')
```

```
In [10]: dataset.describe()
```

```
Out[10]:
```

	iso_numeric	year	rep_meth	new_sp_coh	new_sp_cur	new_sp_cmplt	new
count	5118.000000	5118.000000	5118.000000	5118.000000	5118.000000	5118.000000	511
mean	432.063111	2005.565651	100.528136	10863.51162	7887.186906	964.721656	43
std	253.529561	6.926333	0.825792	35233.88547	28455.963769	2521.550115	123
min	4.000000	1994.000000	100.000000	0.000000	0.000000	0.000000	
25%	212.000000	2000.000000	100.000000	689.000000	395.500000	81.000000	3
50%	430.000000	2006.000000	100.000000	10751.500000	7887.186906	964.721656	43
75%	643.000000	2012.000000	101.000000	10863.51162	7887.186906	964.721656	43
max	894.000000	2017.000000	102.000000	642321.000000	544731.000000	64938.000000	2700

8 rows × 68 columns

Next, check the dataset for duplicated values and treat them accordingly if any

```
In [11]: dataset[dataset.duplicated()].sum()
```

```
Out[11]: iso2          0.0
         iso3          0.0
         iso_numeric   0.0
         g_whoregion    0.0
         year          0.0
         ...
         xdr_coh        0.0
         xdr_succ       0.0
         xdr_fail       0.0
         xdr_died       0.0
         xdr_lost       0.0
         Length: 71, dtype: float64
```

The Dataset contains no duplicate values as it's indicated in the results above. That's a plus, so proceed to other data Preparation steps

The Dataset contains no null values because I'm working with the version of the file which I've cleaned and manipulated beforehand. When you get your file, it will require some data cleaning and formatting before it'll be ready for analysis and exploration

```
In [12]: dataset.isna().sum()
```

```
Out[12]: iso2          0
         iso3          0
         iso_numeric   0
         g_whoregion    0
         year          0
         ..
         xdr_coh        0
         xdr_succ       0
         xdr_fail       0
         xdr_died       0
         xdr_lost       0
         Length: 71, dtype: int64
```

```
In [13]: dataset.info()
```

```

<class 'pandas.core.frame.DataFrame'>
Index: 5118 entries, Afghanistan to Zimbabwe
Data columns (total 71 columns):
iso2                5118 non-null object
iso3                5118 non-null object
iso_numeric         5118 non-null int64
g_whoregion         5118 non-null object
year                5118 non-null int64
rep_meth            5118 non-null int64
new_sp_coh          5118 non-null float64
new_sp_cur          5118 non-null float64
new_sp_cmplt        5118 non-null float64
new_sp_died         5118 non-null float64
new_sp_fail         5118 non-null float64
new_sp_def          5118 non-null float64
c_new_sp_tsr        5118 non-null float64
new_snep_coh        5118 non-null float64
new_snep_cmplt      5118 non-null float64
new_snep_died       5118 non-null float64
new_snep_fail       5118 non-null float64
new_snep_def        5118 non-null float64
c_new_snep_tsr      5118 non-null float64
ret_coh             5118 non-null float64
ret_cur             5118 non-null float64
ret_cmplt           5118 non-null float64
ret_died            5118 non-null float64
ret_fail            5118 non-null float64
ret_def             5118 non-null float64
hiv_new_sp_coh      5118 non-null float64
hiv_new_sp_cur      5118 non-null float64
hiv_new_sp_cmplt    5118 non-null float64
hiv_new_sp_died     5118 non-null float64
hiv_new_sp_fail     5118 non-null float64
hiv_new_sp_def      5118 non-null float64
hiv_new_snep_coh    5118 non-null float64
hiv_new_snep_cmplt  5118 non-null float64
hiv_new_snep_died   5118 non-null float64
hiv_new_snep_fail   5118 non-null float64
hiv_new_snep_def    5118 non-null float64
hiv_ret_coh         5118 non-null float64
hiv_ret_cur         5118 non-null float64
hiv_ret_cmplt       5118 non-null float64
hiv_ret_died        5118 non-null float64
hiv_ret_fail        5118 non-null float64
hiv_ret_def         5118 non-null float64
rel_with_new_flg    5118 non-null float64
newrel_coh          5118 non-null float64
newrel_succ         5118 non-null float64
newrel_fail         5118 non-null float64
newrel_died         5118 non-null float64
newrel_lost         5118 non-null float64
c_new_tsr           5118 non-null float64
ret_nrel_coh        5118 non-null float64
ret_nrel_succ       5118 non-null float64
ret_nrel_fail       5118 non-null float64
ret_nrel_died       5118 non-null float64
ret_nrel_lost       5118 non-null float64

```

```

c_ret_tsr          5118 non-null float64
tbhiv_coh          5118 non-null float64
tbhiv_succ         5118 non-null float64
tbhiv_fail         5118 non-null float64
tbhiv_died         5118 non-null float64
tbhiv_lost         5118 non-null float64
c_tbhiv_tsr        5118 non-null float64
mdr_coh            5118 non-null float64
mdr_succ           5118 non-null float64
mdr_fail           5118 non-null float64
mdr_died           5118 non-null float64
mdr_lost           5118 non-null float64
xdr_coh            5118 non-null float64
xdr_succ           5118 non-null float64
xdr_fail           5118 non-null float64
xdr_died           5118 non-null float64
xdr_lost           5118 non-null float64
dtypes: float64(65), int64(3), object(3)
memory usage: 2.7+ MB

```

Checking the various types of data types contained in the dataset and summing it together. It's just part of understanding the data in question

```
In [14]: dataset.dtypes.value_counts()
```

```

Out[14]: float64    65
         int64      3
         object     3
         dtype: int64

```

The dataset is now ready for analysis and exploration to derive insights.

The insights and visualization you derive from any Dataset depends on your goal. For the sake of practice and learning, most of what will be done here will be on random basis just to help you build your skills

First, you'll determine how many countries are represented in the dataset

```
In [15]: countries_present = dataset.index.unique()
         countries_present1 = len(countries_present)
         countries_present1
```

```
Out[15]: 218
```

```
In [16]: print("There are",countries_present1,"countries represented on this da
         taset")
```

```
There are 218 countries represented on this dataset
```

Next, let's determine the number of regions represented in this dataset

```
In [17]: regions_present = dataset["g_whoregion"].unique()  
regions_present1 = len(regions_present)  
regions_present1
```

Out[17]: 6

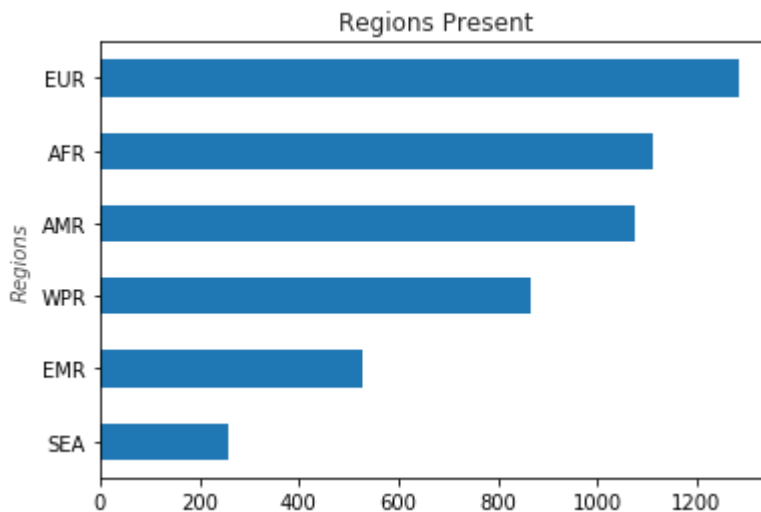
```
In [18]: print("There are", regions_present1,"regions represented on this dataset")
```

There are 6 regions represented on this dataset

A visual plot of the regions in the dataset is good to plot at this stage. It's represented by a horizontal bar chart . Even if a pie chart is utilized, it'll still work perfectly.

```
In [19]: regs = dataset["g_whoregion"].groupby(dataset['g_whoregion'])  
regs1 = regs.size()  
regs11 = regs1.sort_values()  
regs11.plot.barh()  
plt.title("Regions Present", alpha = 0.85)  
plt.ylabel("Regions", alpha = 0.7, fontstyle = 'italic')
```

Out[19]: Text(0, 0.5, 'Regions')



Next is to determine the unique number of years that was covered in the dataset

```
In [20]: years_covered = dataset.year.unique()  
years_covered1 = len(years_covered)  
years_covered1
```

Out[20]: 24


```
In [21]: print('The number of years covered in this dataset is',years_covered1,
"years. It spans between", years_covered.min(),"to",years_covered.max
())
```

The number of years covered in this dataset is 24 years. It spans between 1994 to 2017

Next, Let's extract the number of countries that's present in each region

```
In [22]: curled = dataset["rep_meth"].groupby([dataset["g_whoregion"],dataset.i
ndex])
curled1 = curled.size()
curled1
```

```
Out[22]: g_whoregion  country
AFR                Algeria      24
                Angola      24
                Benin      24
                Botswana      24
                Burkina Faso    24
                ..
WPR                Tonga      24
                Tuvalu      24
                Vanuatu      24
                Viet Nam      24
                Wallis and Futuna Islands 24
Name: rep_meth, Length: 218, dtype: int64
```

```
In [23]: curled11 = curled1.unstack(0)
curled11
```

Out[23]:

g_whoregion	AFR	AMR	EMR	EUR	SEA	WPR
country						
Afghanistan	NaN	NaN	24.0	NaN	NaN	NaN
Albania	NaN	NaN	NaN	24.0	NaN	NaN
Algeria	24.0	NaN	NaN	NaN	NaN	NaN
American Samoa	NaN	NaN	NaN	NaN	NaN	24.0
Andorra	NaN	NaN	NaN	24.0	NaN	NaN
...
Wallis and Futuna Islands	NaN	NaN	NaN	NaN	NaN	24.0
West Bank and Gaza Strip	NaN	NaN	24.0	NaN	NaN	NaN
Yemen	NaN	NaN	24.0	NaN	NaN	NaN
Zambia	24.0	NaN	NaN	NaN	NaN	NaN
Zimbabwe	24.0	NaN	NaN	NaN	NaN	NaN

218 rows × 6 columns

A dataframe containing the regions and the countries has been extracted. The next few steps will be to extract each region and its corresponding countries.

Now let's extract the information which reveals the number of countries in the AFR Region

```
In [24]: AFR_countries = curled11.AFR[curled11.AFR > 0]
AFR_countries1 = AFR_countries.count()
AFR_countries1
```

Out[24]: 47

```
In [25]: print("There are", AFR_countries1,"countries in \"AFR\" region")
```

There are 47 countries in "AFR" region

Next, Let's extract the information which reveals the number of countries in the AMR region

```
In [26]: AMR_countries = curled11.AMR[curled11.AMR > 0]
AMR_countries1 = AMR_countries.count()
AMR_countries1
```

Out[26]: 47

```
In [27]: print("There are",AMR_countries1,"countries in the \"AMR\" region")
```

There are 47 countries in the "AMR" region

Next, I'm going to extract the information that reveals the countries in the EMR Region

```
In [28]: EMR_countries = curled11.EMR[curled11.EMR > 0]
EMR_countries1 = EMR_countries.count()
EMR_countries1
```

Out[28]: 22

```
In [29]: print("There are",EMR_countries1,"countries in the \"EMR\" Region")
```

There are 22 countries in the "EMR" Region

Next, Let's extract the information which reveals the countries in the EUR Region

```
In [30]: EUR_countries = curled11.EUR[curled11.EUR > 0]
EUR_countries1 = EUR_countries.count()
EUR_countries1
```

Out[30]: 55

```
In [31]: print("There are",EUR_countries1,"countries in the \"EUR\" Region")
```

There are 55 countries in the "EUR" Region

Next, Let's extract the information that reveals the number of countries in the SEA Region

```
In [32]: SEA_countries = curled11.SEA[curled11.SEA > 0]
SEA_countries1 = SEA_countries.count()
SEA_countries1
```

Out[32]: 11

```
In [33]: print("There are",SEA_countries1,"countries in the \"SEA\" Region")
```

There are 11 countries in the "SEA" Region

Next, Let's extract the information that reveals the countries in the WPR Region

```
In [34]: WPR_countries = curled11.WPR[curled11.WPR > 0]
WPR_countries1 = WPR_countries.count()
WPR_countries1
```

Out[34]: 36

```
In [35]: print("There are",WPR_countries1,"countries in the \"WPR\" Region")
```

There are 36 countries in the "WPR" Region

```
In [36]: dataset.tail()
```

Out[36]:

	iso2	iso3	iso_numeric	g_whoregion	year	rep_meth	new_sp_coh	new_sp_cur	new_
country									
Zimbabwe	ZW	ZWE	716	AFR	2013	100	10863.51162	7887.186906	96
Zimbabwe	ZW	ZWE	716	AFR	2014	100	10863.51162	7887.186906	96
Zimbabwe	ZW	ZWE	716	AFR	2015	100	10863.51162	7887.186906	96
Zimbabwe	ZW	ZWE	716	AFR	2016	100	10863.51162	7887.186906	96
Zimbabwe	ZW	ZWE	716	AFR	2017	100	10863.51162	7887.186906	96

5 rows × 71 columns

Next, I'm going to determine the specific regions and the number of countries in that region that were attended to in each of the years contained in the dataset.

I strongly feel plotting visualization for this part of the analysis is not so much appropriate. If you decide that doing so will make more meaning, feel free to unleash your imagination

```
In [37]: print('The number of years covered in this dataset is',years_covered1,
"years. It spans between", years_covered.min(),"to",years_covered.max
())
```

The number of years covered in this dataset is 24 years. It spans between 1994 to 2017

```
In [38]: extract094 = dataset[dataset.year == 1994]
extract094
```

Out[38]:

	iso2	iso3	iso_numeric	g_whoregion	year	rep_meth	new_sp_coh	new_sp_cur	new
country									
Afghanistan	AF	AFG	4	EMR	1994	100	10863.51162	7887.186906	9
Albania	AL	ALB	8	EUR	1994	100	10863.51162	7887.186906	9
Algeria	DZ	DZA	12	AFR	1994	100	7253.00000	7887.186906	56
American Samoa	AS	ASM	16	WPR	1994	102	10863.51162	7887.186906	9
Andorra	AD	AND	20	EUR	1994	102	10863.51162	7887.186906	9
...
Wallis and Futuna Islands	WF	WLF	876	WPR	1994	100	5.00000	0.000000	
West Bank and Gaza Strip	PS	PSE	275	EMR	1994	102	10863.51162	7887.186906	9
Yemen	YE	YEM	887	EMR	1994	100	3351.00000	1035.000000	2
Zambia	ZM	ZMB	894	AFR	1994	100	10863.51162	7887.186906	9
Zimbabwe	ZW	ZWE	716	AFR	1994	100	6724.00000	1701.000000	17

211 rows × 10 columns

```
In [39]: extract094.year.unique()
```

Out[39]: array([1994], dtype=int64)

```
In [40]: count094 = extract094.rep_meth.groupby(extract094.index)
count94 = count094.count()
count94.head(20)
```

```
Out[40]: country
Afghanistan      1
Albania          1
Algeria          1
American Samoa  1
Andorra          1
Angola           1
Anguilla         1
Antigua and Barbuda 1
Argentina        1
Armenia          1
Aruba            1
Australia        1
Austria          1
Azerbaijan       1
Bahamas          1
Bahrain          1
Bangladesh       1
Barbados         1
Belarus          1
Belgium          1
Name: rep_meth, dtype: int64
```

```
In [41]: regions094 = extract094.rep_meth.groupby(extract094["g_whoregion"])
regions94 = regions094.size()
regions94
```

```
Out[41]: g_whoregion
AFR      46
AMR      44
EMR      22
EUR      53
SEA      10
WPR      36
Name: rep_meth, dtype: int64
```

```
In [42]: print("In the year",extract094.year.unique(),",",len(count94.index),"C
ountries were attended to and",len(regions94.index),"Regions were atte
nded to respectively")
```

In the year [1994] , 211 Countries were attended to and 6 Regions were attended to respectively

```
In [43]: extract095 = dataset[dataset.year == 1995]
extract095
```

Out[43]:

	iso2	iso3	iso_numeric	g_whoregion	year	rep_meth	new_sp_coh	new_sp_cur	new
country									
Afghanistan	AF	AFG	4	EMR	1995	100	10863.51162	7887.186906	9
Albania	AL	ALB	8	EUR	1995	100	10863.51162	7887.186906	9
Algeria	DZ	DZA	12	AFR	1995	100	10863.51162	7887.186906	9
American Samoa	AS	ASM	16	WPR	1995	100	4.00000	4.000000	
Andorra	AD	AND	20	EUR	1995	102	10863.51162	7887.186906	9
...
Wallis and Futuna Islands	WF	WLF	876	WPR	1995	100	10863.51162	7887.186906	9
West Bank and Gaza Strip	PS	PSE	275	EMR	1995	100	13.00000	13.000000	9
Yemen	YE	YEM	887	EMR	1995	100	3681.00000	1598.000000	3
Zambia	ZM	ZMB	894	AFR	1995	100	5957.00000	2815.000000	13
Zimbabwe	ZW	ZWE	716	AFR	1995	100	9702.00000	3137.000000	20

211 rows × 10 columns

```
In [44]: extract095.year.unique()
```

Out[44]: array([1995], dtype=int64)

```
In [45]: count095 = extract095.rep_meth.groupby(extract095.index)
count95 = count095.count()
count95
```

Out[45]:

country	
Afghanistan	1
Albania	1
Algeria	1
American Samoa	1
Andorra	1
...	..
Wallis and Futuna Islands	1
West Bank and Gaza Strip	1
Yemen	1
Zambia	1
Zimbabwe	1

Name: rep_meth, Length: 211, dtype: int64

```
In [46]: regions095 = extract095.rep_meth.groupby(extract095["g_whoregion"])
regions95 = regions095.size()
regions95
```

```
Out[46]: g_whoregion
AFR      46
AMR      44
EMR      22
EUR      53
SEA      10
WPR      36
Name: rep_meth, dtype: int64
```

```
In [47]: print("In the year",extract095.year.unique(),"",len(count95.index),"c
ountries were attended to and",len(regions95.index),"Regions were atte
nded to respectively")
```

In the year [1995] , 211 countries were attended to and 6 Regions were attended to respectively

```
In [48]: extract096 = dataset[dataset.year == 1996]
extract096
```

```
Out[48]:
```

	iso2	iso3	iso_numeric	g_whoregion	year	rep_meth	new_sp_coh	new_sp_cur	nev
country									
Afghanistan	AF	AFG	4	EMR	1996	100	10863.51162	7887.186906	9
Albania	AL	ALB	8	EUR	1996	100	10863.51162	7887.186906	9
Algeria	DZ	DZA	12	AFR	1996	100	6860.00000	5630.000000	2
American Samoa	AS	ASM	16	WPR	1996	100	10863.51162	7887.186906	9
Andorra	AD	AND	20	EUR	1996	102	10863.51162	7887.186906	9
...
Wallis and Futuna Islands	WF	WLF	876	WPR	1996	100	10863.51162	7887.186906	9
West Bank and Gaza Strip	PS	PSE	275	EMR	1996	100	10863.51162	7887.186906	9
Yemen	YE	YEM	887	EMR	1996	100	4221.00000	1936.000000	2
Zambia	ZM	ZMB	894	AFR	1996	100	10863.51162	7887.186906	9
Zimbabwe	ZW	ZWE	716	AFR	1996	100	11965.00000	3836.000000	

211 rows × 71 columns

```
In [49]: extract096.year.unique()
```

```
Out[49]: array([1996], dtype=int64)
```



```
In [50]: count096 = extract096.rep_meth.groupby(extract096.index)
count96 = count096.size()
count96
```

```
Out[50]: country
Afghanistan          1
Albania              1
Algeria              1
American Samoa       1
Andorra              1
..
Wallis and Futuna Islands  1
West Bank and Gaza Strip  1
Yemen                1
Zambia               1
Zimbabwe             1
Name: rep_meth, Length: 211, dtype: int64
```

```
In [51]: regions096 = extract096.rep_meth.groupby(extract096["g_whoregion"])
regions96 = regions096.size()
regions96
```

```
Out[51]: g_whoregion
AFR      46
AMR      44
EMR      22
EUR      53
SEA      10
WPR      36
Name: rep_meth, dtype: int64
```

```
In [52]: print("In the year",extract096.year.unique(),"",len(count96.index),"C
ountries were attended to and",len(regions96.index),"were attended to
respectively")
```

In the year [1996] , 211 Countries were attended to and 6 were attended to respectively

```
In [53]: extract097 = dataset[dataset.year == 1997]
extract097
```

Out[53]:

	iso2	iso3	iso_numeric	g_whoregion	year	rep_meth	new_sp_coh	new_sp_cur	new
country									
Afghanistan	AF	AFG	4	EMR	1997	100	2001.00000	786.000000	1
Albania	AL	ALB	8	EUR	1997	100	10863.51162	7887.186906	9
Algeria	DZ	DZA	12	AFR	1997	100	6860.00000	5630.000000	2
American Samoa	AS	ASM	16	WPR	1997	100	10863.51162	7887.186906	9
Andorra	AD	AND	20	EUR	1997	102	10863.51162	7887.186906	9
...
Wallis and Futuna Islands	WF	WLF	876	WPR	1997	100	10863.51162	7887.186906	9
West Bank and Gaza Strip	PS	PSE	275	EMR	1997	100	10863.51162	7887.186906	9
Yemen	YE	YEM	887	EMR	1997	100	4365.00000	2608.000000	4
Zambia	ZM	ZMB	894	AFR	1997	100	10863.51162	7887.186906	9
Zimbabwe	ZW	ZWE	716	AFR	1997	100	12410.00000	6361.000000	21

211 rows × 10 columns

```
In [54]: extract097.year.unique()
```

Out[54]: array([1997], dtype=int64)

```
In [55]: count097 = extract097.rep_meth.groupby(extract097.index)
count97 = count097.size()
count97
```

Out[55]:

```
country
Afghanistan      1
Albania          1
Algeria          1
American Samoa   1
Andorra          1
...
Wallis and Futuna Islands  1
West Bank and Gaza Strip  1
Yemen            1
Zambia           1
Zimbabwe         1
Name: rep_meth, Length: 211, dtype: int64
```

```
In [56]: regions097 = extract097.rep_meth.groupby(extract097["g_whoregion"])
regions97 = regions097.size()
regions97
```

```
Out[56]: g_whoregion
AFR      46
AMR      44
EMR      22
EUR      53
SEA      10
WPR      36
Name: rep_meth, dtype: int64
```

```
In [57]: print("In the year",extract097.year.unique(),"",len(count97.index),"C
ountries were attended to and",len(regions97.index),"Regions were atte
nded to respectively")
```

In the year [1997] , 211 Countries were attended to and 6 Regions were attended to respectively

```
In [58]: extract098 = dataset[dataset.year == 1998]
extract098
```

```
Out[58]:
```

	iso2	iso3	iso_numeric	g_whoregion	year	rep_meth	new_sp_coh	new_sp_cur	new
country									
Afghanistan	AF	AFG	4	EMR	1998	100	2913.00000	772.000000	1
Albania	AL	ALB	8	EUR	1998	100	10863.51162	7887.186906	9
Algeria	DZ	DZA	12	AFR	1998	100	2490.00000	1741.000000	9
American Samoa	AS	ASM	16	WPR	1998	100	4.00000	2.000000	9
Andorra	AD	AND	20	EUR	1998	100	2.00000	0.000000	
...	
Wallis and Futuna Islands	WF	WLF	876	WPR	1998	100	10863.51162	7887.186906	9
West Bank and Gaza Strip	PS	PSE	275	EMR	1998	100	10863.51162	7887.186906	9
Yemen	YE	YEM	887	EMR	1998	100	4983.00000	2889.000000	4
Zambia	ZM	ZMB	894	AFR	1998	100	10863.51162	7887.186906	9
Zimbabwe	ZW	ZWE	716	AFR	1998	100	12748.00000	6420.000000	24

211 rows × 10 columns

```
In [59]: extract098.year.unique()
```

```
Out[59]: array([1998], dtype=int64)
```

```
In [60]: count098 = extract098.rep_meth.groupby(extract098.index)
count98 = count098.size()
count98
```

```
Out[60]: country
Afghanistan          1
Albania              1
Algeria              1
American Samoa       1
Andorra              1
..
Wallis and Futuna Islands  1
West Bank and Gaza Strip  1
Yemen                1
Zambia               1
Zimbabwe             1
Name: rep_meth, Length: 211, dtype: int64
```

```
In [61]: regions098 = extract098.rep_meth.groupby(extract098["g_whoregion"])
regions98 = regions098.size()
regions98
```

```
Out[61]: g_whoregion
AFR      46
AMR      44
EMR      22
EUR      53
SEA      10
WPR      36
Name: rep_meth, dtype: int64
```

```
In [62]: print("In the year",extract098.year.unique(),"",len(count98.index),"C
ountries were attended to and",len(regions98.index),"were attended to
respectively")
```

In the year [1998] , 211 Countries were attended to and 6 were attended to respectively

```
In [63]: extract099 = dataset[dataset.year == 1999]
extract099
```

Out[63]:

	iso2	iso3	iso_numeric	g_whoregion	year	rep_meth	new_sp_coh	new_sp_cur	new_sp_tot
country									
Afghanistan	AF	AFG	4	EMR	1999	100	2039.00000	1571.000000	1
Albania	AL	ALB	8	EUR	1999	100	10863.51162	7887.186906	9
Algeria	DZ	DZA	12	AFR	1999	100	7622.00000	6621.000000	9
American Samoa	AS	ASM	16	WPR	1999	100	3.00000	3.000000	1
Andorra	AD	AND	20	EUR	1999	100	3.00000	7887.186906	9
...
Wallis and Futuna Islands	WF	WLF	876	WPR	1999	100	10863.51162	7887.186906	9
West Bank and Gaza Strip	PS	PSE	275	EMR	1999	100	10863.51162	7887.186906	9
Yemen	YE	YEM	887	EMR	1999	100	5463.00000	3482.000000	5
Zambia	ZM	ZMB	894	AFR	1999	100	11645.00000	5808.000000	22
Zimbabwe	ZW	ZWE	716	AFR	1999	100	12791.00000	7535.000000	17

211 rows × 10 columns

```
In [64]: extract099.year.unique()
```

Out[64]: array([1999], dtype=int64)

```
In [65]: count099 = extract099.rep_meth.groupby(extract099.index)
count99 = count099.size()
count99
```

Out[65]:

```
country
Afghanistan      1
Albania          1
Algeria          1
American Samoa   1
Andorra          1
...
Wallis and Futuna Islands  1
West Bank and Gaza Strip  1
Yemen            1
Zambia           1
Zimbabwe         1
Name: rep_meth, Length: 211, dtype: int64
```

```
In [66]: regions099 = extract099.rep_meth.groupby(extract099["g_whoregion"])
regions99 = regions099.size()
regions99
```

```
Out[66]: g_whoregion
AFR      46
AMR      44
EMR      22
EUR      53
SEA      10
WPR      36
Name: rep_meth, dtype: int64
```

```
In [67]: print("In the year",extract099.year.unique(),"",len(count99.index),"C
ountries were attended to and",len(regions99.index),"Regions were atte
nded to respectively")
```

In the year [1999] , 211 Countries were attended to and 6 Regions were attended to respectively

```
In [68]: extract000 = dataset[dataset.year == 2000]
extract000
```

```
Out[68]:
```

	iso2	iso3	iso_numeric	g_whoregion	year	rep_meth	new_sp_coh	new_sp_cur	nev
country									
Afghanistan	AF	AFG	4	EMR	2000	100	3136.00000	2396.000000	2
Albania	AL	ALB	8	EUR	2000	100	10863.51162	7887.186906	9
Algeria	DZ	DZA	12	AFR	2000	100	8328.00000	6690.000000	5
American Samoa	AS	ASM	16	WPR	2000	100	2.00000	0.000000	
Andorra	AD	AND	20	EUR	2000	100	2.00000	7887.186906	
...	
Wallis and Futuna Islands	WF	WLF	876	WPR	2000	100	10863.51162	7887.186906	9
West Bank and Gaza Strip	PS	PSE	275	EMR	2000	100	10863.51162	7887.186906	9
Yemen	YE	YEM	887	EMR	2000	100	5565.00000	3273.000000	7
Zambia	ZM	ZMB	894	AFR	2000	100	7014.00000	3348.000000	13
Zimbabwe	ZW	ZWE	716	AFR	2000	100	14392.00000	8820.000000	10

211 rows × 71 columns

```
In [69]: extract000.year.unique()
```

```
Out[69]: array([2000], dtype=int64)
```

```
In [70]: count000 = extract000.rep_meth.groupby(extract000.index)
count00 = count000.size()
count00
```

```
Out[70]: country
Afghanistan          1
Albania              1
Algeria              1
American Samoa       1
Andorra              1
..
Wallis and Futuna Islands  1
West Bank and Gaza Strip  1
Yemen                1
Zambia               1
Zimbabwe             1
Name: rep_meth, Length: 211, dtype: int64
```

```
In [71]: regions000 = extract000.rep_meth.groupby(extract000["g_whoregion"])
regions00 = regions000.size()
regions00
```

```
Out[71]: g_whoregion
AFR      46
AMR      44
EMR      22
EUR      53
SEA      10
WPR      36
Name: rep_meth, dtype: int64
```

```
In [72]: print("In the year",extract000.year.unique(),"",len(count00.index),"C
ountries were attended to and",len(regions00.index),"Regions were atte
nded to respectively")
```

In the year [2000] , 211 Countries were attended to and 6 Regions were attended to respectively

```
In [73]: extract001 = dataset[dataset["year"] == 2001]
extract001
```

Out[73]:

	iso2	iso3	iso_numeric	g_whoregion	year	rep_meth	new_sp_coh	new_sp_cur	ne
country									
Afghanistan	AF	AFG	4	EMR	2001	100	6292.00000	3305.000000	2
Albania	AL	ALB	8	EUR	2001	100	171.00000	76.000000	
Algeria	DZ	DZA	12	AFR	2001	100	8361.00000	4561.000000	2
American Samoa	AS	ASM	16	WPR	2001	100	2.00000	7887.186906	
Andorra	AD	AND	20	EUR	2001	100	1.00000	1.000000	
...	
Wallis and Futuna Islands	WF	WLF	876	WPR	2001	100	1.00000	1.000000	
West Bank and Gaza Strip	PS	PSE	275	EMR	2001	100	10863.51162	7887.186906	
Yemen	YE	YEM	887	EMR	2001	100	4968.00000	3206.000000	
Zambia	ZM	ZMB	894	AFR	2001	100	13024.00000	7246.000000	2
Zimbabwe	ZW	ZWE	716	AFR	2001	100	16569.00000	10521.000000	1

211 rows × 71 columns

```
In [74]: extract001.year.unique()
```

Out[74]: array([2001], dtype=int64)

```
In [75]: count001 = extract001.rep_meth.groupby(extract001.index)
count01 = count001.size()
count01
```

Out[75]:

country	
Afghanistan	1
Albania	1
Algeria	1
American Samoa	1
Andorra	1
...	..
Wallis and Futuna Islands	1
West Bank and Gaza Strip	1
Yemen	1
Zambia	1
Zimbabwe	1

Name: rep_meth, Length: 211, dtype: int64


```
In [76]: regions001 = extract001.rep_meth.groupby(extract001["g_whoregion"])
regions01 = regions001.size()
regions01
```

```
Out[76]: g_whoregion
AFR      46
AMR      44
EMR      22
EUR      53
SEA      10
WPR      36
Name: rep_meth, dtype: int64
```

```
In [77]: print("In the year",extract001.year.unique(),"",len(count01.index),"C
countries were attended to and",len(regions01.index),"were attended to
respectively")
```

In the year [2001] , 211 Countries were attended to and 6 were attended to respectively

```
In [78]: extract002 = dataset[dataset.year == 2002]
extract002
```

```
Out[78]:
```

	iso2	iso3	iso_numeric	g_whoregion	year	rep_meth	new_sp_coh	new_sp_cur	new_sp_tot
country									
Afghanistan	AF	AFG	4	EMR	2002	100	7780.0	4668.000000	2002
Albania	AL	ALB	8	EUR	2002	100	225.0	98.000000	2002
Algeria	DZ	DZA	12	AFR	2002	100	9200.0	6631.000000	10000
American Samoa	AS	ASM	16	WPR	2002	100	1.0	7887.186906	2002
Andorra	AD	AND	20	EUR	2002	100	3.0	3.000000	2002
...
Wallis and Futuna Islands	WF	WLF	876	WPR	2002	100	5.0	5.000000	2002
West Bank and Gaza Strip	PS	PSE	275	EMR	2002	100	10.0	10.000000	2002
Yemen	YE	YEM	887	EMR	2002	100	4204.0	2864.000000	2002
Zambia	ZM	ZMB	894	AFR	2002	100	16351.0	10410.000000	2002
Zimbabwe	ZW	ZWE	716	AFR	2002	100	15941.0	9833.000000	2002

212 rows × 10 columns

```
In [79]: extract002.year.unique()
```

```
Out[79]: array([2002], dtype=int64)
```

```
In [80]: count002 = extract002.rep_meth.groupby(extract002.index)
count02 = count002.size()
count02
```

```
Out[80]: country
Afghanistan          1
Albania              1
Algeria              1
American Samoa       1
Andorra              1
..
Wallis and Futuna Islands  1
West Bank and Gaza Strip  1
Yemen                1
Zambia               1
Zimbabwe             1
Name: rep_meth, Length: 212, dtype: int64
```

```
In [81]: regions002 = extract002.rep_meth.groupby(extract002["g_whoregion"])
regions02 = regions002.size()
regions02
```

```
Out[81]: g_whoregion
AFR      46
AMR      44
EMR      22
EUR      53
SEA      11
WPR      36
Name: rep_meth, dtype: int64
```

```
In [82]: print("In the year",extract002.year.unique(),"",len(count02.index),"C
ountries were attended to and",len(regions02.index),"were attended to
respectively")
```

In the year [2002] , 212 Countries were attended to and 6 were attended to respectively

```
In [83]: extract003 = dataset[dataset.year == 2003]
extract003
```

Out[83]:

	iso2	iso3	iso_numeric	g_whoregion	year	rep_meth	new_sp_coh	new_sp_cur	ne
country									
Afghanistan	AF	AFG	4	EMR	2003	100	6793.00000	5505.000000	
Albania	AL	ALB	8	EUR	2003	100	212.00000	104.000000	
Algeria	DZ	DZA	12	AFR	2003	100	8521.00000	6548.000000	1
American Samoa	AS	ASM	16	WPR	2003	100	2.00000	2.000000	
Andorra	AD	AND	20	EUR	2003	100	6.00000	6.000000	
...	
Wallis and Futuna Islands	WF	WLF	876	WPR	2003	100	10863.51162	7887.186906	
West Bank and Gaza Strip	PS	PSE	275	EMR	2003	100	15.00000	10.000000	
Yemen	YE	YEM	887	EMR	2003	100	3793.00000	2624.000000	
Zambia	ZM	ZMB	894	AFR	2003	100	18934.00000	12603.000000	1
Zimbabwe	ZW	ZWE	716	AFR	2003	100	14488.00000	8892.000000	

212 rows × 71 columns

```
In [84]: extract003.year.unique()
```

Out[84]: array([2003], dtype=int64)

```
In [85]: count003 = extract003.rep_meth.groupby(extract003.index)
count03 = count003.size()
count03
```

Out[85]:

country	
Afghanistan	1
Albania	1
Algeria	1
American Samoa	1
Andorra	1
...	..
Wallis and Futuna Islands	1
West Bank and Gaza Strip	1
Yemen	1
Zambia	1
Zimbabwe	1

Name: rep_meth, Length: 212, dtype: int64

```
In [86]: regions003 = extract003.rep_meth.groupby(extract003.g_whoregion)
regions03 = regions003.size()
regions03
```

```
Out[86]: g_whoregion
AFR      46
AMR      44
EMR      22
EUR      53
SEA      11
WPR      36
Name: rep_meth, dtype: int64
```

```
In [87]: print("In the year",extract003.year.unique(),"",len(count03.index),"C
ountries were attended to and",len(regions03.index),"Regions were atte
nded to respectively")
```

In the year [2003] , 212 Countries were attended to and 6 Regions were attended to respectively

```
In [88]: extract004 = dataset[dataset.year == 2004]
extract004
```

```
Out[88]:
```

	iso2	iso3	iso_numeric	g_whoregion	year	rep_meth	new_sp_coh	new_sp_cur	new_sp_tot
country									
Afghanistan	AF	AFG	4	EMR	2004	100	9976.0	7846.000000	10000.000000
Albania	AL	ALB	8	EUR	2004	100	201.0	94.000000	10000.000000
Algeria	DZ	DZA	12	AFR	2004	100	8405.0	6229.000000	10000.000000
American Samoa	AS	ASM	16	WPR	2004	100	3.0	7887.186906	10000.000000
Andorra	AD	AND	20	EUR	2004	100	3.0	2.000000	10000.000000
...
Wallis and Futuna Islands	WF	WLF	876	WPR	2004	100	1.0	1.000000	10000.000000
West Bank and Gaza Strip	PS	PSE	275	EMR	2004	100	4.0	2.000000	10000.000000
Yemen	YE	YEM	887	EMR	2004	100	3501.0	2382.000000	10000.000000
Zambia	ZM	ZMB	894	AFR	2004	100	17247.0	13108.000000	10000.000000
Zimbabwe	ZW	ZWE	716	AFR	2004	100	14581.0	7248.000000	10000.000000

211 rows × 10 columns

```
In [89]: extract004.year.unique()
```

```
Out[89]: array([2004], dtype=int64)
```

```
In [90]: count004 = extract004.rep_meth.groupby(extract004.index)
count04 = count004.size()
count04
```

```
Out[90]: country
Afghanistan          1
Albania              1
Algeria              1
American Samoa       1
Andorra              1
..
Wallis and Futuna Islands  1
West Bank and Gaza Strip  1
Yemen                1
Zambia               1
Zimbabwe             1
Name: rep_meth, Length: 211, dtype: int64
```

```
In [91]: regions004 = extract004.rep_meth.groupby(extract004.g_whoregion)
regions04 = regions004.size()
regions04
```

```
Out[91]: g_whoregion
AFR      46
AMR      44
EMR      22
EUR      52
SEA      11
WPR      36
Name: rep_meth, dtype: int64
```

```
In [92]: print("In the year",extract004.year.unique(),"",len(count04.index),"C
ountries were attended to and",len(regions04.index),"Regions were atte
nded to respectively")
```

In the year [2004] , 211 Countries were attended to and 6 Regions were attended to respectively

```
In [93]: extract005 = dataset[dataset.year == 2005]
extract005
```

Out[93]:

	iso2	iso3	iso_numeric	g_whoregion	year	rep_meth	new_sp_coh	new_sp_cur	ne
country									
Afghanistan	AF	AFG	4	EMR	2005	100	10013.00000	8295.000000	
Albania	AL	ALB	8	EUR	2005	100	196.00000	85.000000	
Algeria	DZ	DZA	12	AFR	2005	100	8379.00000	6180.000000	1
American Samoa	AS	ASM	16	WPR	2005	100	4.00000	3.000000	
Andorra	AD	AND	20	EUR	2005	100	5.00000	4.000000	
...	
Wallis and Futuna Islands	WF	WLF	876	WPR	2005	100	10863.51162	7887.186906	
West Bank and Gaza Strip	PS	PSE	275	EMR	2005	100	12.00000	7.000000	
Yemen	YE	YEM	887	EMR	2005	100	3566.00000	2464.000000	
Zambia	ZM	ZMB	894	AFR	2005	100	14857.00000	11290.000000	1
Zimbabwe	ZW	ZWE	716	AFR	2005	100	12860.00000	7554.000000	1

213 rows × 71 columns

```
In [94]: extract005.year.unique()
```

Out[94]: array([2005], dtype=int64)

```
In [95]: count005 = extract005.rep_meth.groupby(extract005.index)
count05 = count005.size()
count05
```

Out[95]:

country	
Afghanistan	1
Albania	1
Algeria	1
American Samoa	1
Andorra	1
...	..
Wallis and Futuna Islands	1
West Bank and Gaza Strip	1
Yemen	1
Zambia	1
Zimbabwe	1
Name: rep_meth, Length: 213, dtype: int64	

```
In [96]: regions005 = extract005.rep_meth.groupby(extract005.g_whoregion)
regions05 = regions005.size()
regions05
```

```
Out[96]: g_whoregion
AFR      46
AMR      44
EMR      22
EUR      54
SEA      11
WPR      36
Name: rep_meth, dtype: int64
```

```
In [97]: print("In the year",extract005.year.unique(),"",len(count05.index),"C
ountries were attended to and",len(regions05.index),"Regions were atte
nded to respectively")
```

In the year [2005] , 213 Countries were attended to and 6 Regions were attended to respectively

```
In [98]: extract006 = dataset[dataset.year == 2006]
extract006
```

```
Out[98]:
```

	iso2	iso3	iso_numeric	g_whoregion	year	rep_meth	new_sp_coh	new_sp_cur	new_!
country									
Afghanistan	AF	AFG	4	EMR	2006	100	12468.0	9921.0	
Albania	AL	ALB	8	EUR	2006	100	186.0	91.0	
Algeria	DZ	DZA	12	AFR	2006	100	8285.0	7148.0	
American Samoa	AS	ASM	16	WPR	2006	100	0.0	0.0	
Andorra	AD	AND	20	EUR	2006	100	8.0	1.0	
...	
Wallis and Futuna Islands	WF	WLF	876	WPR	2006	100	4.0	2.0	
West Bank and Gaza Strip	PS	PSE	275	EMR	2006	100	16.0	8.0	
Yemen	YE	YEM	887	EMR	2006	100	3337.0	2474.0	
Zambia	ZM	ZMB	894	AFR	2006	100	14025.0	10762.0	
Zimbabwe	ZW	ZWE	716	AFR	2006	100	16205.0	8757.0	

213 rows × 71 columns

```
In [99]: extract006.year.unique()
```

```
Out[99]: array([2006], dtype=int64)
```

```
In [100]: count006 = extract006.rep_meth.groupby(extract006.index)
count06 = count006.size()
count06
```

```
Out[100]: country
Afghanistan          1
Albania              1
Algeria              1
American Samoa       1
Andorra              1
..
Wallis and Futuna Islands  1
West Bank and Gaza Strip  1
Yemen                1
Zambia               1
Zimbabwe             1
Name: rep_meth, Length: 213, dtype: int64
```

```
In [101]: regions006 = extract006.rep_meth.groupby(extract006.g_whoregion)
regions06 = regions006.size()
regions06
```

```
Out[101]: g_whoregion
AFR      46
AMR      44
EMR      22
EUR      54
SEA      11
WPR      36
Name: rep_meth, dtype: int64
```

```
In [102]: print("In the year",extract006.year.unique(),"",len(count06.index),"C
ountries were attended to and",len(regions06.index),"were attended to
respectively")
```

In the year [2006] , 213 Countries were attended to and 6 were attended to respectively


```
In [103]: extract007 = dataset[dataset.year == 2007]
extract007
```

Out[103]:

	iso2	iso3	iso_numeric	g_whoregion	year	rep_meth	new_sp_coh	new_sp_cur	ne
country									
Afghanistan	AF	AFG	4	EMR	2007	100	13213.00000	10859.000000	
Albania	AL	ALB	8	EUR	2007	100	181.00000	90.000000	
Algeria	DZ	DZA	12	AFR	2007	100	8510.00000	6699.000000	
American Samoa	AS	ASM	16	WPR	2007	102	0.00000	0.000000	
Andorra	AD	AND	20	EUR	2007	101	2.00000	2.000000	
...	
Wallis and Futuna Islands	WF	WLF	876	WPR	2007	100	10863.51162	7887.186906	
West Bank and Gaza Strip	PS	PSE	275	EMR	2007	100	14.00000	7.000000	
Yemen	YE	YEM	887	EMR	2007	100	3523.00000	2644.000000	
Zambia	ZM	ZMB	894	AFR	2007	102	13378.00000	10447.000000	
Zimbabwe	ZW	ZWE	716	AFR	2007	100	10583.00000	7455.000000	

213 rows × 71 columns

```
In [104]: extract007.year.unique()
```

Out[104]: array([2007], dtype=int64)

```
In [105]: count007 = extract007.rep_meth.groupby(extract007.index)
count07 = count007.size()
count07
```

Out[105]:

country	
Afghanistan	1
Albania	1
Algeria	1
American Samoa	1
Andorra	1
...	..
Wallis and Futuna Islands	1
West Bank and Gaza Strip	1
Yemen	1
Zambia	1
Zimbabwe	1

Name: rep_meth, Length: 213, dtype: int64

```
In [106]: regions007 = extract007.rep_meth.groupby(extract007.g_whoregion)
regions07 = regions007.size()
regions07
```

```
Out[106]: g_whoregion
AFR      46
AMR      44
EMR      22
EUR      54
SEA      11
WPR      36
Name: rep_meth, dtype: int64
```

```
In [107]: print("In the year",extract007.year.unique(),"",len(count07.index),"C
ountries were attended to and",len(regions07.index),"Regions were atte
nded to respectively")
```

In the year [2007] , 213 Countries were attended to and 6 Regions were attended to respectively

```
In [108]: extract008 = dataset[dataset.year == 2008]
extract008
```

```
Out[108]:
```

	iso2	iso3	iso_numeric	g_whoregion	year	rep_meth	new_sp_coh	new_sp_cur	new_!
country									
Afghanistan	AF	AFG	4	EMR	2008	100	13136.0	10936.0	
Albania	AL	ALB	8	EUR	2008	100	170.0	88.0	
Algeria	DZ	DZA	12	AFR	2008	102	8190.0	6584.0	
American Samoa	AS	ASM	16	WPR	2008	102	0.0	0.0	
Andorra	AD	AND	20	EUR	2008	101	3.0	1.0	
...	
Wallis and Futuna Islands	WF	WLF	876	WPR	2008	102	3.0	3.0	
West Bank and Gaza Strip	PS	PSE	275	EMR	2008	100	16.0	6.0	
Yemen	YE	YEM	887	EMR	2008	100	3540.0	2671.0	
Zambia	ZM	ZMB	894	AFR	2008	100	13173.0	10875.0	
Zimbabwe	ZW	ZWE	716	AFR	2008	100	10370.0	6973.0	

213 rows × 71 columns

```
In [109]: extract008.year.unique()
```

```
Out[109]: array([2008], dtype=int64)
```

```
In [110]: count008 = extract008.rep_meth.groupby(extract008.index)
count08 = count008.size()
count08
```

```
Out[110]: country
Afghanistan          1
Albania              1
Algeria              1
American Samoa       1
Andorra              1
..
Wallis and Futuna Islands  1
West Bank and Gaza Strip  1
Yemen                1
Zambia               1
Zimbabwe             1
Name: rep_meth, Length: 213, dtype: int64
```

```
In [111]: regions008 = extract008.rep_meth.groupby(extract008.g_whoregion)
regions08 = regions008.size()
regions08
```

```
Out[111]: g_whoregion
AFR      46
AMR      44
EMR      22
EUR      54
SEA      11
WPR      36
Name: rep_meth, dtype: int64
```

```
In [112]: print("In the year",extract008.year.unique(),"",len(count08.index),"C
ountries were attended to and",len(regions08.index),"Regions were atte
nded to respectively")
```

In the year [2008] , 213 Countries were attended to and 6 Regions were attended to respectively

```
In [113]: extract009 = dataset[dataset.year == 2009]
extract009
```

Out[113]:

	iso2	iso3	iso_numeric	g_whoregion	year	rep_meth	new_sp_coh	new_sp_cur	ne
country									
Afghanistan	AF	AFG	4	EMR	2009	100	12497.00000	10323.000000	
Albania	AL	ALB	8	EUR	2009	100	171.00000	110.000000	
Algeria	DZ	DZA	12	AFR	2009	102	8438.00000	6864.000000	
American Samoa	AS	ASM	16	WPR	2009	100	3.00000	0.000000	
Andorra	AD	AND	20	EUR	2009	102	3.00000	1.000000	
...	
Wallis and Futuna Islands	WF	WLF	876	WPR	2009	102	10863.51162	7887.186906	
West Bank and Gaza Strip	PS	PSE	275	EMR	2009	100	11.00000	2.000000	
Yemen	YE	YEM	887	EMR	2009	100	3557.00000	2804.000000	
Zambia	ZM	ZMB	894	AFR	2009	102	12995.00000	10983.000000	
Zimbabwe	ZW	ZWE	716	AFR	2009	100	10195.00000	7131.000000	

215 rows × 71 columns

```
In [114]: extract009.year.unique()
```

Out[114]: array([2009], dtype=int64)

```
In [115]: count009 = extract009.rep_meth.groupby(extract009.index)
count09 = count009.size()
count09
```

Out[115]:

country	
Afghanistan	1
Albania	1
Algeria	1
American Samoa	1
Andorra	1
...	..
Wallis and Futuna Islands	1
West Bank and Gaza Strip	1
Yemen	1
Zambia	1
Zimbabwe	1

Name: rep_meth, Length: 215, dtype: int64

```
In [116]: regions009 = extract009.rep_meth.groupby(extract009.g_whoregion)
regions09 = regions009.size()
regions09
```

```
Out[116]: g_whoregion
AFR      46
AMR      46
EMR      22
EUR      54
SEA      11
WPR      36
Name: rep_meth, dtype: int64
```

```
In [117]: print("In the year",extract009.year.unique(),"",len(count09.index),"Countries were attended to and",len(regions09.index),"Regions were attended to respectively")
```

In the year [2009] , 215 Countries were attended to and 6 Regions were attended to respectively

```
In [118]: extract010 = dataset[dataset.year == 2010]
extract010
```

```
Out[118]:
```

	iso2	iso3	iso_numeric	g_whoregion	year	rep_meth	new_sp_coh	new_sp_cur	new_sp_tot
country									
Afghanistan	AF	AFG	4	EMR	2010	100	12947.0	11175.000000	4
Albania	AL	ALB	8	EUR	2010	100	145.0	71.000000	4
Algeria	DZ	DZA	12	AFR	2010	102	7894.0	6219.000000	8
American Samoa	AS	ASM	16	WPR	2010	102	0.0	7887.186906	9
Andorra	AD	AND	20	EUR	2010	102	0.0	0.000000	4
...
Wallis and Futuna Islands	WF	WLF	876	WPR	2010	102	2.0	0.000000	4
West Bank and Gaza Strip	PS	PSE	275	EMR	2010	100	12.0	1.000000	4
Yemen	YE	YEM	887	EMR	2010	100	3584.0	2775.000000	3
Zambia	ZM	ZMB	894	AFR	2010	102	12639.0	10532.000000	7
Zimbabwe	ZW	ZWE	716	AFR	2010	100	11654.0	8377.000000	1

216 rows × 10 columns

```
In [119]: extract010.year.unique()
```

```
Out[119]: array([2010], dtype=int64)
```

```
In [120]: count010 = extract010.rep_meth.groupby(extract010.index)
count10 = count010.size()
count10
```

```
Out[120]: country
Afghanistan          1
Albania              1
Algeria              1
American Samoa       1
Andorra              1
..
Wallis and Futuna Islands  1
West Bank and Gaza Strip  1
Yemen                1
Zambia               1
Zimbabwe             1
Name: rep_meth, Length: 216, dtype: int64
```

```
In [121]: regions010 = extract010.rep_meth.groupby(extract010.g_whoregion)
regions10 = regions010.size()
regions10
```

```
Out[121]: g_whoregion
AFR      47
AMR      46
EMR      22
EUR      54
SEA      11
WPR      36
Name: rep_meth, dtype: int64
```

```
In [122]: print("In the year",extract010.year.unique(),"",len(count10.index),"C
ountries were attended to and",len(regions10.index),"Regions were atte
nded to respectively")
```

In the year [2010] , 216 Countries were attended to and 6 Regions were attended to respectively

```
In [123]: extract011 = dataset[dataset.year == 2011]
extract011
```

Out[123]:

	iso2	iso3	iso_numeric	g_whoregion	year	rep_meth	new_sp_coh	new_sp_cur	ne
country									
Afghanistan	AF	AFG	4	EMR	2011	100	13789.00000	12067.000000	
Albania	AL	ALB	8	EUR	2011	100	180.00000	117.000000	
Algeria	DZ	DZA	12	AFR	2011	102	7364.00000	5969.000000	
American Samoa	AS	ASM	16	WPR	2011	102	10863.51162	7887.186906	
Andorra	AD	AND	20	EUR	2011	102	1.00000	0.000000	
...
Wallis and Futuna Islands	WF	WLF	876	WPR	2011	102	10863.51162	7887.186906	
West Bank and Gaza Strip	PS	PSE	275	EMR	2011	100	11.00000	2.000000	
Yemen	YE	YEM	887	EMR	2011	100	3174.00000	2517.000000	
Zambia	ZM	ZMB	894	AFR	2011	102	12711.00000	10463.000000	
Zimbabwe	ZW	ZWE	716	AFR	2011	100	12596.00000	9208.000000	

216 rows × 71 columns

```
In [124]: extract011.year.unique()
```

Out[124]: array([2011], dtype=int64)

```
In [125]: count011 = extract011.rep_meth.groupby(extract011.index)
count11 = count011.size()
count11
```

Out[125]:

country	
Afghanistan	1
Albania	1
Algeria	1
American Samoa	1
Andorra	1
...	..
Wallis and Futuna Islands	1
West Bank and Gaza Strip	1
Yemen	1
Zambia	1
Zimbabwe	1

Name: rep_meth, Length: 216, dtype: int64

```
In [126]: regions011 = extract011.rep_meth.groupby(extract011.g_whoregion)
regions11 = regions011.size()
regions11
```

```
Out[126]: g_whoregion
AFR      47
AMR      46
EMR      22
EUR      54
SEA      11
WPR      36
Name: rep_meth, dtype: int64
```

```
In [127]: print("In the year",extract011.year.unique(),"",len(count11.index),"C
ountries were attended to and",len(regions11.index),"Regions were atte
nded to respectively")
```

In the year [2011] , 216 Countries were attended to and 6 Regions were attended to respectively

```
In [128]: extract012 = dataset[dataset.year == 2012]
extract012
```

```
Out[128]:
```

	iso2	iso3	iso_numeric	g_whoregion	year	rep_meth	new_sp_coh	new_sp_cur	new
country									
Afghanistan	AF	AFG	4	EMR	2012	100	10863.51162	7887.186906	9
Albania	AL	ALB	8	EUR	2012	100	10863.51162	7887.186906	9
Algeria	DZ	DZA	12	AFR	2012	102	10863.51162	7887.186906	9
American Samoa	AS	ASM	16	WPR	2012	102	10863.51162	7887.186906	9
Andorra	AD	AND	20	EUR	2012	102	10863.51162	7887.186906	9
...
Wallis and Futuna Islands	WF	WLF	876	WPR	2012	102	10863.51162	7887.186906	9
West Bank and Gaza Strip	PS	PSE	275	EMR	2012	100	10863.51162	7887.186906	9
Yemen	YE	YEM	887	EMR	2012	100	10863.51162	7887.186906	9
Zambia	ZM	ZMB	894	AFR	2012	102	10863.51162	7887.186906	9
Zimbabwe	ZW	ZWE	716	AFR	2012	100	10863.51162	7887.186906	9

216 rows × 71 columns

```
In [129]: extract012.year.unique()
```

```
Out[129]: array([2012], dtype=int64)
```



```
In [130]: count012 = extract012.rep_meth.groupby(extract012.index)
count12 = count012.size()
count12
```

```
Out[130]: country
Afghanistan          1
Albania              1
Algeria              1
American Samoa       1
Andorra              1
..
Wallis and Futuna Islands  1
West Bank and Gaza Strip  1
Yemen                1
Zambia               1
Zimbabwe             1
Name: rep_meth, Length: 216, dtype: int64
```

```
In [131]: regions012 = extract012.rep_meth.groupby(extract012.g_whoregion)
regions12 = regions012.size()
regions12
```

```
Out[131]: g_whoregion
AFR      47
AMR      46
EMR      22
EUR      54
SEA      11
WPR      36
Name: rep_meth, dtype: int64
```

```
In [132]: print("In the year",extract012.year.unique(),"",len(count12.index),"C
ountries were attended to and",len(regions012),"Regions were attended
to respectively")
```

In the year [2012] , 216 Countries were attended to and 6 Regions were attended to respectively

```
In [133]: extract013 = dataset[dataset.year == 2013]
extract013
```

Out[133]:

	iso2	iso3	iso_numeric	g_whoregion	year	rep_meth	new_sp_coh	new_sp_cur	new
country									
Afghanistan	AF	AFG	4	EMR	2013	100	10863.51162	7887.186906	9
Albania	AL	ALB	8	EUR	2013	100	10863.51162	7887.186906	9
Algeria	DZ	DZA	12	AFR	2013	102	10863.51162	7887.186906	9
American Samoa	AS	ASM	16	WPR	2013	102	10863.51162	7887.186906	9
Andorra	AD	AND	20	EUR	2013	102	10863.51162	7887.186906	9
...
Wallis and Futuna Islands	WF	WLF	876	WPR	2013	102	10863.51162	7887.186906	9
West Bank and Gaza Strip	PS	PSE	275	EMR	2013	100	10863.51162	7887.186906	9
Yemen	YE	YEM	887	EMR	2013	100	10863.51162	7887.186906	9
Zambia	ZM	ZMB	894	AFR	2013	102	10863.51162	7887.186906	9
Zimbabwe	ZW	ZWE	716	AFR	2013	100	10863.51162	7887.186906	9

216 rows × 10 columns

```
In [134]: extract013.year.unique()
```

Out[134]: array([2013], dtype=int64)

```
In [135]: count013 = extract013.rep_meth.groupby(extract013.index)
count13 = count013.size()
count13
```

Out[135]:

country	
Afghanistan	1
Albania	1
Algeria	1
American Samoa	1
Andorra	1
...	..
Wallis and Futuna Islands	1
West Bank and Gaza Strip	1
Yemen	1
Zambia	1
Zimbabwe	1

Name: rep_meth, Length: 216, dtype: int64

```
In [136]: regions013 = extract013.rep_meth.groupby(extract013.g_whoregion)
regions13 = regions013.size()
regions13
```

```
Out[136]: g_whoregion
AFR      47
AMR      46
EMR      22
EUR      54
SEA      11
WPR      36
Name: rep_meth, dtype: int64
```

```
In [137]: print("In the year",extract013.year.unique(),"",len(count13.index),"C
ountries were attended to and",len(regions13.index),"Regions were atte
nded to respectively")
```

In the year [2013] , 216 Countries were attended to and 6 Regions were attended to respectively

```
In [138]: extract014 = dataset[dataset.year == 2014]
extract014
```

```
Out[138]:
```

	iso2	iso3	iso_numeric	g_whoregion	year	rep_meth	new_sp_coh	new_sp_cur	new
country									
Afghanistan	AF	AFG	4	EMR	2014	100	10863.51162	7887.186906	9
Albania	AL	ALB	8	EUR	2014	100	10863.51162	7887.186906	9
Algeria	DZ	DZA	12	AFR	2014	102	10863.51162	7887.186906	9
American Samoa	AS	ASM	16	WPR	2014	102	10863.51162	7887.186906	9
Andorra	AD	AND	20	EUR	2014	102	10863.51162	7887.186906	9
...
Wallis and Futuna Islands	WF	WLF	876	WPR	2014	102	10863.51162	7887.186906	9
West Bank and Gaza Strip	PS	PSE	275	EMR	2014	100	10863.51162	7887.186906	9
Yemen	YE	YEM	887	EMR	2014	100	10863.51162	7887.186906	9
Zambia	ZM	ZMB	894	AFR	2014	102	10863.51162	7887.186906	9
Zimbabwe	ZW	ZWE	716	AFR	2014	100	10863.51162	7887.186906	9

216 rows × 71 columns

```
In [139]: extract014.year.unique()
```

```
Out[139]: array([2014], dtype=int64)
```

```
In [140]: count014 = extract014.rep_meth.groupby(extract014.index)
count14 = count014.size()
count14
```

```
Out[140]: country
Afghanistan          1
Albania              1
Algeria              1
American Samoa       1
Andorra              1
..
Wallis and Futuna Islands  1
West Bank and Gaza Strip  1
Yemen                1
Zambia               1
Zimbabwe             1
Name: rep_meth, Length: 216, dtype: int64
```

```
In [141]: regions014 = extract014.rep_meth.groupby(extract014.g_whoregion)
regions14 = regions014.size()
regions14
```

```
Out[141]: g_whoregion
AFR      47
AMR      46
EMR      22
EUR      54
SEA      11
WPR      36
Name: rep_meth, dtype: int64
```

```
In [142]: print("In the year",extract014.year.unique(),"",len(count14.index),"C
ountries were attended to and",len(regions14.index),"Regions were atte
nded to respectively")
```

In the year [2014] , 216 Countries were attended to and 6 Regions were attended to respectively

```
In [143]: extract015 = dataset[dataset.year == 2015]
extract015
```

Out[143]:

	iso2	iso3	iso_numeric	g_whoregion	year	rep_meth	new_sp_coh	new_sp_cur	new
country									
Afghanistan	AF	AFG	4	EMR	2015	100	10863.51162	7887.186906	9
Albania	AL	ALB	8	EUR	2015	100	10863.51162	7887.186906	9
Algeria	DZ	DZA	12	AFR	2015	102	10863.51162	7887.186906	9
American Samoa	AS	ASM	16	WPR	2015	102	10863.51162	7887.186906	9
Andorra	AD	AND	20	EUR	2015	102	10863.51162	7887.186906	9
...
Wallis and Futuna Islands	WF	WLF	876	WPR	2015	102	10863.51162	7887.186906	9
West Bank and Gaza Strip	PS	PSE	275	EMR	2015	100	10863.51162	7887.186906	9
Yemen	YE	YEM	887	EMR	2015	100	10863.51162	7887.186906	9
Zambia	ZM	ZMB	894	AFR	2015	102	10863.51162	7887.186906	9
Zimbabwe	ZW	ZWE	716	AFR	2015	100	10863.51162	7887.186906	9

216 rows × 71 columns

```
In [144]: extract015.year.unique()
```

Out[144]: array([2015], dtype=int64)

```
In [145]: count015 = extract015.rep_meth.groupby(extract015.index)
count15 = count015.size()
count15
```

Out[145]:

country	
Afghanistan	1
Albania	1
Algeria	1
American Samoa	1
Andorra	1
...	..
Wallis and Futuna Islands	1
West Bank and Gaza Strip	1
Yemen	1
Zambia	1
Zimbabwe	1

Name: rep_meth, Length: 216, dtype: int64

```
In [146]: regions015 = extract015.rep_meth.groupby(extract015.g_whoregion)
regions15 = regions015.size()
regions15
```

```
Out[146]: g_whoregion
AFR      47
AMR      46
EMR      22
EUR      54
SEA      11
WPR      36
Name: rep_meth, dtype: int64
```

```
In [147]: print("In the year",extract015.year.unique(),"",len(count15.index),"C
ountries were attended to and",len(regions15.index),"Regions were atte
nded to respectively")
```

In the year [2015] , 216 Countries were attended to and 6 Regions were attended to respectively

```
In [148]: extract016 = dataset[dataset.year == 2016]
extract016
```

```
Out[148]:
```

	iso2	iso3	iso_numeric	g_whoregion	year	rep_meth	new_sp_coh	new_sp_cur	new
country									
Afghanistan	AF	AFG	4	EMR	2016	100	10863.51162	7887.186906	9
Albania	AL	ALB	8	EUR	2016	100	10863.51162	7887.186906	9
Algeria	DZ	DZA	12	AFR	2016	102	10863.51162	7887.186906	9
American Samoa	AS	ASM	16	WPR	2016	102	10863.51162	7887.186906	9
Andorra	AD	AND	20	EUR	2016	102	10863.51162	7887.186906	9
...
Wallis and Futuna Islands	WF	WLF	876	WPR	2016	102	10863.51162	7887.186906	9
West Bank and Gaza Strip	PS	PSE	275	EMR	2016	100	10863.51162	7887.186906	9
Yemen	YE	YEM	887	EMR	2016	100	10863.51162	7887.186906	9
Zambia	ZM	ZMB	894	AFR	2016	102	10863.51162	7887.186906	9
Zimbabwe	ZW	ZWE	716	AFR	2016	100	10863.51162	7887.186906	9

216 rows × 71 columns

```
In [149]: extract016.year.unique()
```

```
Out[149]: array([2016], dtype=int64)
```

```
In [150]: count016 = extract016.rep_meth.groupby(extract016.index)
count16 = count016.size()
count16
```

```
Out[150]: country
Afghanistan          1
Albania              1
Algeria              1
American Samoa       1
Andorra              1
..
Wallis and Futuna Islands  1
West Bank and Gaza Strip  1
Yemen                1
Zambia               1
Zimbabwe             1
Name: rep_meth, Length: 216, dtype: int64
```

```
In [151]: regions016 = extract016.rep_meth.groupby(extract016.g_whoregion)
regions16 = regions016.size()
regions16
```

```
Out[151]: g_whoregion
AFR      47
AMR      46
EMR      22
EUR      54
SEA      11
WPR      36
Name: rep_meth, dtype: int64
```

```
In [152]: print("In the year",extract016.year.unique(),"",len(count16.index),"C
ountries were attended to and",len(regions16.index),"Regions were atte
nded to respectively")
```

In the year [2016] , 216 Countries were attended to and 6 Regions were attended to respectively

```
In [153]: extract017 = dataset[dataset.year == 2017]
extract017
```

Out[153]:

	iso2	iso3	iso_numeric	g_whoregion	year	rep_meth	new_sp_coh	new_sp_cur	new
country									
Afghanistan	AF	AFG	4	EMR	2017	100	10863.51162	7887.186906	9
Albania	AL	ALB	8	EUR	2017	100	10863.51162	7887.186906	9
Algeria	DZ	DZA	12	AFR	2017	102	10863.51162	7887.186906	9
American Samoa	AS	ASM	16	WPR	2017	102	10863.51162	7887.186906	9
Andorra	AD	AND	20	EUR	2017	102	10863.51162	7887.186906	9
...
Wallis and Futuna Islands	WF	WLF	876	WPR	2017	102	10863.51162	7887.186906	9
West Bank and Gaza Strip	PS	PSE	275	EMR	2017	100	10863.51162	7887.186906	9
Yemen	YE	YEM	887	EMR	2017	100	10863.51162	7887.186906	9
Zambia	ZM	ZMB	894	AFR	2017	102	10863.51162	7887.186906	9
Zimbabwe	ZW	ZWE	716	AFR	2017	100	10863.51162	7887.186906	9

216 rows × 71 columns

```
In [154]: extract017.year.unique()
```

Out[154]: array([2017], dtype=int64)

```
In [155]: count017 = extract017.rep_meth.groupby(extract017.index)
count17 = count017.size()
count17
```

Out[155]:

country	
Afghanistan	1
Albania	1
Algeria	1
American Samoa	1
Andorra	1
...	..
Wallis and Futuna Islands	1
West Bank and Gaza Strip	1
Yemen	1
Zambia	1
Zimbabwe	1

Name: rep_meth, Length: 216, dtype: int64


```
In [159]: regions94.index
```

```
Out[159]: Index(['AFR', 'AMR', 'EMR', 'EUR', 'SEA', 'WPR'], dtype='object', name='g_whoregion')
```

```
In [160]: new_index = list(regions94.index)
new_index
regions_series.index = new_index
regions_series.index.name = 'who_region'
regions_series.columns.name = "Years"
```

```
In [161]: regions_series
```

```
Out[161]:
```

	Years	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	...	2008	2009	2010
who_region															
AFR	46	46	46	46	46	46	46	46	46	46	46	...	46	46	46
AMR	44	44	44	44	44	44	44	44	44	44	44	...	44	46	44
EMR	22	22	22	22	22	22	22	22	22	22	22	...	22	22	22
EUR	53	53	53	53	53	53	53	53	53	53	53	...	54	54	54
SEA	10	10	10	10	10	10	10	10	10	11	11	...	11	11	11
WPR	36	36	36	36	36	36	36	36	36	36	36	...	36	36	36

6 rows × 24 columns

```
In [162]: regs_series = regions_series.T  
regs_series
```

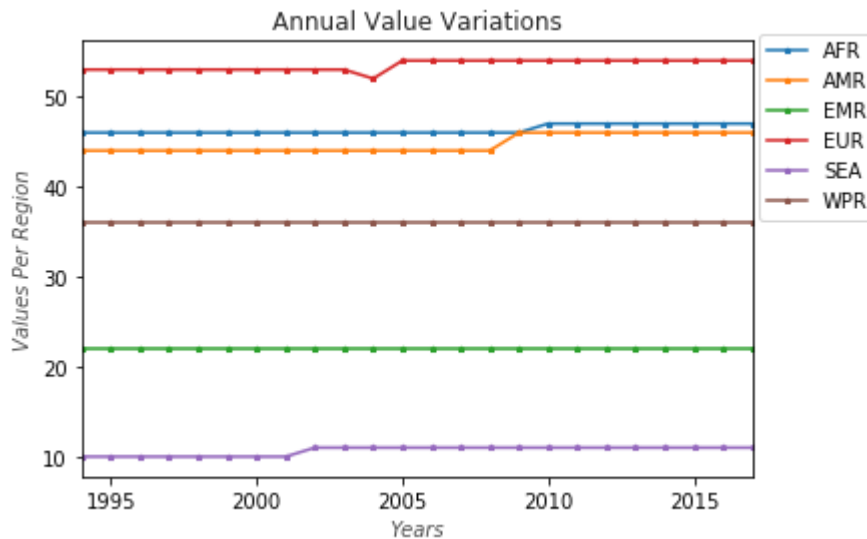
Out[162]:

who_region	AFR	AMR	EMR	EUR	SEA	WPR
Years						
1994	46	44	22	53	10	36
1995	46	44	22	53	10	36
1996	46	44	22	53	10	36
1997	46	44	22	53	10	36
1998	46	44	22	53	10	36
1999	46	44	22	53	10	36
2000	46	44	22	53	10	36
2001	46	44	22	53	10	36
2002	46	44	22	53	11	36
2003	46	44	22	53	11	36
2004	46	44	22	52	11	36
2005	46	44	22	54	11	36
2006	46	44	22	54	11	36
2007	46	44	22	54	11	36
2008	46	44	22	54	11	36
2009	46	46	22	54	11	36
2010	47	46	22	54	11	36
2011	47	46	22	54	11	36
2012	47	46	22	54	11	36
2013	47	46	22	54	11	36
2014	47	46	22	54	11	36
2015	47	46	22	54	11	36
2016	47	46	22	54	11	36
2017	47	46	22	54	11	36

make a visual plot using this DataFrame

```
In [163]: regs_series.plot(marker = '*', markersize = 3)
plt.title("Annual Value Variations", alpha = 0.85)
plt.ylabel("Values Per Region", alpha = 0.7, fontstyle = 'italic')
plt.xlabel("Years", alpha = 0.7, fontstyle = 'italic')
plt.legend(ncol = 1, loc = (1.01, 0.59))
```

Out[163]: <matplotlib.legend.Legend at 0xa3420090>



```
In [164]: champ_data = regs_series.iloc[-11:, 0:3]
champ_data
```

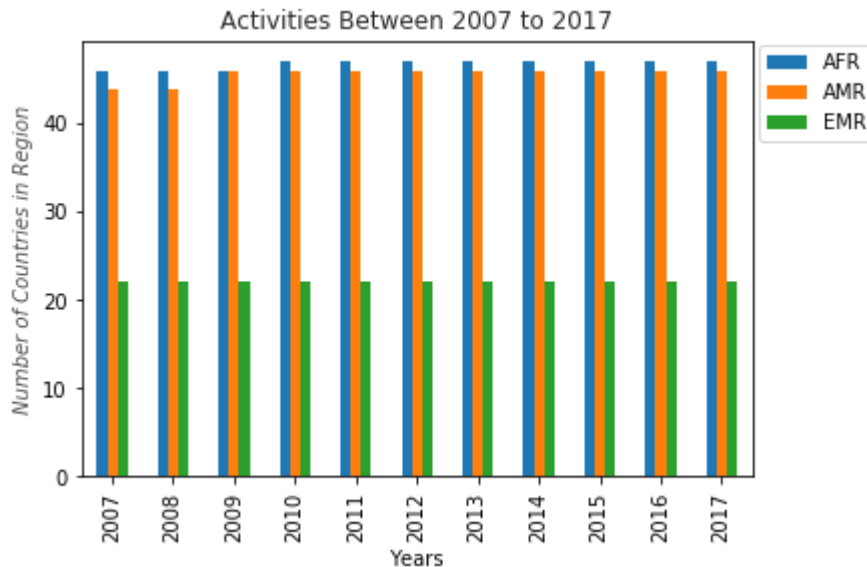
Out[164]:

who_region	AFR	AMR	EMR
Years			
2007	46	44	22
2008	46	44	22
2009	46	46	22
2010	47	46	22
2011	47	46	22
2012	47	46	22
2013	47	46	22
2014	47	46	22
2015	47	46	22
2016	47	46	22
2017	47	46	22

Make a visual plot using the concatenated Output as well

```
In [165]: champ_data.plot(kind = "bar")
plt.title("Activities Between 2007 to 2017", alpha = 0.85)
plt.legend(ncol = 1, loc = (1.01, 0.77))
plt.ylabel("Number of Countries in Region", alpha = 0.7, fontstyle = 'i
talics')
```

Out[165]: Text(0, 0.5, 'Number of Countries in Region')



```
In [166]: regions_comb = pd.concat([regions94,regions95,regions96,regions97,regions98,regions99,regions00,regions01,regions02,regions03,regions04,regions05,regions06,regions07,regions08,regions09,regions10,regions11,regions12, regions13,regions14,regions15,regions16,regions17], keys =["1994", '1995', "1996", "1997", "1998", "1999", "2000", "2001", "2002", "2003", "2004", "2005", "2006", "2007", "2008", "2009", "2010", "2011", "2012", "2013", "2014", "2015", "2016", "2017"])
regions_comb
```

Out[166]:

	g_whoregion	
1994	AFR	46
	AMR	44
	EMR	22
	EUR	53
	SEA	10
	WPR	36
2017	AMR	46
	EMR	22
	EUR	54
	SEA	11
	WPR	36

Name: rep_meth, Length: 144, dtype: int64

```
In [167]: region_comb = regions_comb.unstack()  
region_comb
```

Out[167]:

g_whoregion	AFR	AMR	EMR	EUR	SEA	WPR
1994	46	44	22	53	10	36
1995	46	44	22	53	10	36
1996	46	44	22	53	10	36
1997	46	44	22	53	10	36
1998	46	44	22	53	10	36
1999	46	44	22	53	10	36
2000	46	44	22	53	10	36
2001	46	44	22	53	10	36
2002	46	44	22	53	11	36
2003	46	44	22	53	11	36
2004	46	44	22	52	11	36
2005	46	44	22	54	11	36
2006	46	44	22	54	11	36
2007	46	44	22	54	11	36
2008	46	44	22	54	11	36
2009	46	46	22	54	11	36
2010	47	46	22	54	11	36
2011	47	46	22	54	11	36
2012	47	46	22	54	11	36
2013	47	46	22	54	11	36
2014	47	46	22	54	11	36
2015	47	46	22	54	11	36
2016	47	46	22	54	11	36
2017	47	46	22	54	11	36

```
In [168]: region_comb.index.name = "Year"
region_comb.columns.name = "Regions"
region_comb.head()
```

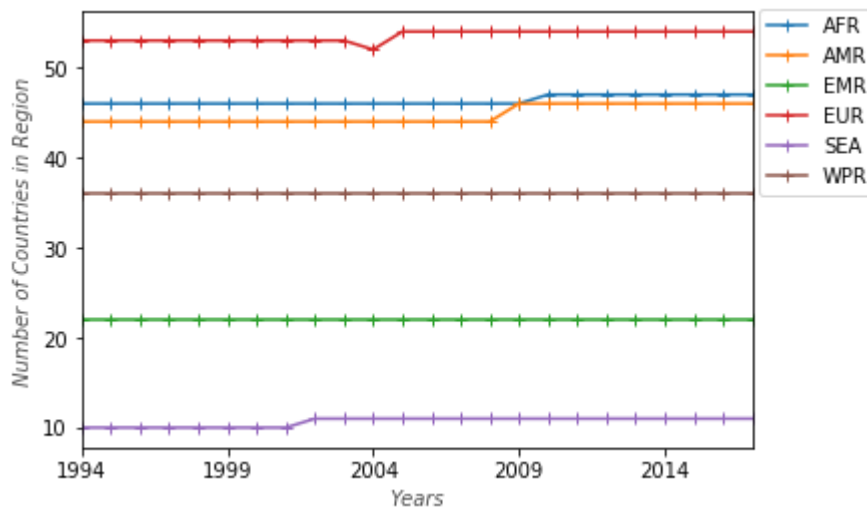
```
Out[168]:
```

Regions	AFR	AMR	EMR	EUR	SEA	WPR
Year						
1994	46	44	22	53	10	36
1995	46	44	22	53	10	36
1996	46	44	22	53	10	36
1997	46	44	22	53	10	36
1998	46	44	22	53	10	36

Make a visual plot using this output as well

```
In [169]: region_comb.plot(marker = "+")
plt.ylabel("Number of Countries in Region", alpha = 0.7, fontstyle = 'i
talic')
plt.xlabel("Years", alpha = 0.7, fontstyle = 'italic')
plt.legend(ncol = 1, loc = (1.01, 0.58))
```

```
Out[169]: <matplotlib.legend.Legend at 0xa2829b10>
```



Like always, there's no limit to length at which you can unleash and utilize your imagination

Feel free to do more just as your creativity can take you.

There's less visual plotting in this lesson, and i believe that leaves you with a huge opportunity to make that a personal task.

Make as many visual plots as you deem fit, to build you competence using the

This marks the end of another lesson using Python's amazing Libraries for Data Science / Data Analysis. Be encouraed and let your creativity guide you on this career path. Nerr feel intimidated even if you're doing it a vit poorly. Keep putting in some extra effort. It's just a matter of time before you gain mastery of it

Till I bring another lesson your way,

Happy Learning !!!

In []:

In []:

In []: