GDSC Probation Task-1

Problem Statement:

Your first task is based on Exploratory Data Analysis of the given dataset.

Here is the dataset of Covid 19 patients segregated Country Wise Apply the principles of Exploratory Data Analysis (EDA) to draw your inferences about the data.
Also, depict the necessary Correlations in the data.
Submit an .ipynb file for the task with required Documentation

A drive link with notebooks and presentation is shared for your reference.
Do research on your own as well. Do not limit yourself to concepts and techniques mentioned in the material.
https://drive.google.com/drive/folders/13-SNOvxYdilzIf2tfQZhE6WZCHpK0qK2

Modules Imported

- ***Pandas***
- ***Numpy***
- ***Matplotlib***
- ***Seaborn***

Reading .csv file

```
In [37]: df=pd.read_csv("E:\GDSC\GDSC.csv")
         print(df)
```

```
          Country/Region  Confirmed  Deaths  Recovered  Active  New cases  \
0            Afghanistan      36263    1269      25198    9796        106
1                Albania       4880     144       2745    1991        117
2                Algeria      27973    1163      18837    7973        616
3                Andorra        907      52        803      52         10
4                 Angola        950      41        242     667         18
..                   ...        ...     ...        ...     ...        ...
182  West Bank and Gaza      10621      78       3752    6791        152
183      Western Sahara         10       1          8       1          0
184               Yemen       1691     483        833     375         10
185              Zambia       4552     140       2815    1597         71
186            Zimbabwe       2704      36        542    2126        192

     New deaths  New recovered  Deaths / 100 Cases  Recovered / 100 Cases  \
0            10             18                3.50                  69.49
1             6             63                2.95                  56.25
2             8            749                4.16                  67.34
3             0              0                5.73                  88.53
4             1              0                4.32                  25.47
```

```
..          ...            ...                 ...                    ...
182           2              0                0.73                  35.33
183           0              0               10.00                  80.00
184           4             36               28.56                  49.26
185           1            465                3.08                  61.84
186           2             24                1.33                  20.04

     Deaths / 100 Recovered  Confirmed last week  1 week change  \
0                      5.04                35526            737
1                      5.25                 4171            709
2                      6.17                23691           4282
3                      6.48                  884             23
4                     16.94                  749            201
..                      ...                  ...            ...
182                    2.08                 8916           1705
183                   12.50                   10              0
184                   57.98                 1619             72
185                    4.97                 3326           1226
186                    6.64                 1713            991
```

```
183                   12.50                   10              0
184                   57.98                 1619             72
185                    4.97                 3326           1226
186                    6.64                 1713            991

     1 week % increase             WHO Region
0                 2.07  Eastern Mediterranean
1                17.00                 Europe
2                18.07                 Africa
3                 2.60                 Europe
4                26.84                 Africa
..                 ...                    ...
182              19.12  Eastern Mediterranean
183               0.00                 Africa
184               4.45  Eastern Mediterranean
185              36.86                 Africa
186              57.85                 Africa

[187 rows x 15 columns]
```

Listing out the columns of the dataset

```
In [38]: l=df.columns
         l

Out[38]: Index(['Country/Region', 'Confirmed', 'Deaths', 'Recovered', 'Active',
                'New cases', 'New deaths', 'New recovered', 'Deaths / 100 Cases',
                'Recovered / 100 Cases', 'Deaths / 100 Recovered',
                'Confirmed last week', '1 week change', '1 week % increase',
                'WHO Region'],
               dtype='object')
```

Finding out the NULL values in the dataset and counting them

```
In [40]: df.info()

         <class 'pandas.core.frame.DataFrame'>
         RangeIndex: 187 entries, 0 to 186
         Data columns (total 15 columns):
          #   Column                  Non-Null Count  Dtype
         ---  ------                  --------------  -----
          0   Country/Region          187 non-null    object
          1   Confirmed               187 non-null    int64
          2   Deaths                  187 non-null    int64
          3   Recovered               187 non-null    int64
          4   Active                  187 non-null    int64
          5   New cases               187 non-null    int64
          6   New deaths              187 non-null    int64
          7   New recovered           187 non-null    int64
          8   Deaths / 100 Cases      187 non-null    float64
          9   Recovered / 100 Cases   187 non-null    float64
          10  Deaths / 100 Recovered  187 non-null    float64
          11  Confirmed last week     187 non-null    int64
          12  1 week change           187 non-null    int64
          13  1 week % increase       187 non-null    float64
          14  WHO Region              187 non-null    object
         dtypes: float64(4), int64(9), object(2)
         memory usage: 22.0+ KB
```

```
In [41]: df.isnull().sum()

Out[41]: Country/Region            0
         Confirmed                 0
         Deaths                    0
         Recovered                 0
         Active                    0
         New cases                 0
         New deaths                0
         New recovered             0
         Deaths / 100 Cases        0
         Recovered / 100 Cases     0
         Deaths / 100 Recovered    0
         Confirmed last week       0
         1 week change             0
         1 week % increase         0
         WHO Region                0
         dtype: int64
```

Finding and Removing NULL values that are present in the Row

```
In [42]: df.dropna(how='all',inplace=True)
         df
```

Out[42]:

	Country/Region	Confirmed	Deaths	Recovered	Active	New cases	New deaths	New recovered	Deaths / 100 Cases	Recovered / 100 Cases	Deaths / 100 Recovered	Confirmed last week	1 week change	1 week % increase	WHO Region
0	Afghanistan	36263	1269	25198	9796	106	10	18	3.50	69.49	5.04	35526	737	2.07	Eastern Mediterranean
1	Albania	4880	144	2745	1991	117	6	63	2.95	56.25	5.25	4171	709	17.00	Europe
2	Algeria	27973	1163	18837	7973	616	8	749	4.16	67.34	6.17	23691	4282	18.07	Africa
3	Andorra	907	52	803	52	10	0	0	5.73	88.53	6.48	884	23	2.60	Europe
4	Angola	950	41	242	667	18	1	0	4.32	25.47	16.94	749	201	26.84	Africa
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
182	West Bank and Gaza	10621	78	3752	6791	152	2	0	0.73	35.33	2.08	8916	1705	19.12	Eastern Mediterranean
183	Western Sahara	10	1	8	1	0	0	0	10.00	80.00	12.50	10	0	0.00	Africa

- Since the number of rows haven't changed, we concluded that there is no such row present in thee dataset with all values NULL in it.

Taking a part of data

- Head(<n>) function is used to take a part of the data either from the top or from the bottom
- Number of records we want can be passed in the function however by default it gives 5 records.

```
In [43]: df.head(20)
```

Out[43]:

	Country/Region	Confirmed	Deaths	Recovered	Active	New cases	New deaths	New recovered	Deaths / 100 Cases	Recovered / 100 Cases	Deaths / 100 Recovered	Confirmed last week	1 week change	1 week % increase	WHO Region
0	Afghanistan	36263	1269	25198	9796	106	10	18	3.50	69.49	5.04	35526	737	2.07	Eastern Mediterranean
1	Albania	4880	144	2745	1991	117	6	63	2.95	56.25	5.25	4171	709	17.00	Europe
2	Algeria	27973	1163	18837	7973	616	8	749	4.16	67.34	6.17	23691	4282	18.07	Africa
3	Andorra	907	52	803	52	10	0	0	5.73	88.53	6.48	884	23	2.60	Europe
4	Angola	950	41	242	667	18	1	0	4.32	25.47	16.94	749	201	26.84	Africa
5	Antigua and Barbuda	86	3	65	18	4	0	5	3.49	75.58	4.62	76	10	13.16	Americas
6	Argentina	167416	3059	72575	91782	4890	120	2057	1.83	43.35	4.21	130774	36642	28.02	Americas
7	Armenia	37390	711	26665	10014	73	6	187	1.90	71.32	2.67	34981	2409	6.89	Europe
8	Australia	15303	167	9311	5825	368	6	137	1.09	60.84	1.79	12428	2875	23.13	Western Pacific
9	Austria	20558	713	18246	1599	86	1	37	3.47	88.75	3.91	19743	815	4.13	Europe
10	Azerbaijan	30446	423	23242	6781	396	6	558	1.39	76.34	1.82	27890	2556	9.16	Europe
11	Bahamas	382	11	91	280	40	0	0	2.88	23.82	12.09	174	208	119.54	Americas
12	Bahrain	39482	141	36110	3231	351	1	421	0.36	91.46	0.39	36936	2546	6.89	Eastern Mediterranean
13	Bangladesh	226225	2965	125683	97577	2772	37	1801	1.31	55.56	2.36	207453	18772	9.05	South-East Asia
14	Barbados	110	7	94	9	0	0	0	6.36	85.45	7.45	106	4	3.77	Americas
15	Belarus	67251	538	60492	6221	119	4	67	0.80	89.95	0.89	66213	1038	1.57	Europe
16	Belgium	66428	9822	17452	39154	402	1	14	14.79	26.27	56.28	64094	2334	3.64	Europe
17	Belize	48	2	26	20	0	0	0	4.17	54.17	7.69	40	8	20.00	Americas
18	Benin	1770	35	1036	699	0	0	0	1.98	58.53	3.38	1602	168	10.49	Africa

Grouping the Data

- As inferred from the Dataset, all the countries are divided into 6 WHO Regions i.e., **Africa, America, Eastern Mediterranean, Europe, South-East Asia and Western Pacific**
- Grouping the data as per the WHO Region can help us to analyze the data better.
- **Operation-1 performed while grouping – mean()**

```
In [46]: grp=df.groupby("WHO Region").mean()
         grp
         grp=df.groupby("WHO Region").mean()
Out[46]:
```

WHO Region	Confirmed	Deaths	Recovered	Active	New cases	New deaths	New recovered	Deaths / 100 Cases	Recovered / 100 Cases	Deaths / 100 Recovered	Confirm last we
Africa	15066.812500	254.645833	9180.104167	5632.062500	253.666667	7.354167	303.395833	2.306458	57.014792	NaN	12669.1666
Americas	252551.028571	9792.342857	127674.742857	115083.942857	3700.885714	101.571429	2707.885714	3.052571	62.291429	NaN	223291.3714
Eastern Mediterranean	67761.090909	1742.681818	54609.090909	11409.318182	564.090909	20.227273	674.681818	3.563182	66.593182	NaN	63583.9545
Europe	58920.053571	3770.428571	35602.196429	19547.428571	398.107143	5.428571	209.500000	4.198393	68.635000	NaN	56193.1428
South-East Asia	183529.700000	4134.900000	115693.300000	63701.500000	4899.300000	73.400000	3758.200000	1.296000	66.704000	1.9560	147828.3000
Western Pacific	18276.750000	515.562500	12923.125000	4838.062500	205.562500	1.500000	70.437500	1.290000	76.805000	1.7875	16647.4375

- NaN depicts presence of NULL Values in the Dataset
- Methods acquired to remove the NULL values from the Dataset – **bfill()**

```
In [108]: grp.fillna(method='bfill')
Out[108]:
```

WHO Region	Confirmed	Deaths	Recovered	Active	New cases	New deaths	New recovered	Deaths / 100 Cases	Recovered / 100 Cases	Deaths / 100 Recovered	Confirme last wee
Africa	15066.812500	254.645833	9180.104167	5632.062500	253.666667	7.354167	303.395833	2.306458	57.014792	1.9560	12669.16666
Americas	252551.028571	9792.342857	127674.742857	115083.942857	3700.885714	101.571429	2707.885714	3.052571	62.291429	1.9560	223291.37142
Eastern Mediterranean	67761.090909	1742.681818	54609.090909	11409.318182	564.090909	20.227273	674.681818	3.563182	66.593182	1.9560	63583.95454
Europe	58920.053571	3770.428571	35602.196429	19547.428571	398.107143	5.428571	209.500000	4.198393	68.635000	1.9560	56193.14285
South-East Asia	183529.700000	4134.900000	115693.300000	63701.500000	4899.300000	73.400000	3758.200000	1.296000	66.704000	1.9560	147828.30000
Western Pacific	18276.750000	515.562500	12923.125000	4838.062500	205.562500	1.500000	70.437500	1.290000	76.805000	1.7875	16647.43750

- **Operation-2 performed while grouping – median()**
- No NULL values encountered

```
In [45]: a=df.groupby("WHO Region").median()
         a
         r select only columns which should be valid for the function.
           a=df.groupby("WHO Region").median()
Out[45]:
```

WHO Region	Confirmed	Deaths	Recovered	Active	New cases	New deaths	New recovered	Deaths / 100 Cases	Recovered / 100 Cases	Deaths / 100 Recovered	Confirmed last week	1 week change	1 week % increase
Africa	2129.5	47.5	1005.5	991.5	22.5	0.0	7.0	1.835	63.145	3.605	1769.5	207.5	10.555
Americas	7340.0	115.0	2905.0	2817.0	104.0	2.0	5.0	2.910	63.870	4.330	7053.0	800.0	12.970
Eastern Mediterranean	28575.0	330.5	20875.5	4391.5	211.0	3.0	82.0	1.820	76.630	2.700	26544.0	1844.5	5.755
Europe	12191.0	398.0	5574.5	2926.0	101.5	1.0	56.5	3.090	76.015	4.690	11351.0	903.5	4.335
South-East Asia	3333.0	31.5	2829.0	740.0	45.0	0.0	17.0	0.880	74.475	1.225	3124.5	222.5	7.070
Western Pacific	1009.5	14.5	977.0	73.0	5.5	0.0	2.5	1.240	87.285	1.445	1003.0	51.0	4.030

Ranking of the Data

- Ranking is done on the basis of Confirmed cases to find the most affected country and least affected country.

```
In [58]: g=df['Rank']=df['Confirmed'].rank(ascending=False)
         g

Out[58]: 0       51.0
         1       96.0
         2       57.0
         3      145.0
         4      143.0
                ...
         182     77.5
         183    187.0
         184    130.0
         185     98.0
         186    113.0
         Name: Confirmed, Length: 187, dtype: float64
```

```
In [59]: sorted_data=df.sort_values(by='Rank',ascending=True)
         sorted_data

Out[59]:
```

ountry/Region	Confirmed	Deaths	Recovered	Active	New cases	New deaths	New recovered	Deaths / 100 Cases	Recovered / 100 Cases	Deaths / 100 Recovered	Confirmed last week	1 week change	1 week % increase	WHO Region	Rank
US	4290259	148011	1325804	2816444	56336	1076	27941	3.45	30.90	11.16	3834677	455582	11.88	Americas	1.0
Brazil	2442375	87618	1846641	508116	23284	614	33728	3.59	75.61	4.74	2118646	323729	15.28	Americas	2.0
India	1480073	33408	951166	495499	44457	637	33598	2.26	64.26	3.51	1155338	324735	28.11	South-East Asia	3.0
Russia	816680	13334	602249	201097	5607	85	3077	1.63	73.74	2.21	776212	40468	5.21	Europe	4.0
South Africa	452529	7067	274925	170537	7096	298	9848	1.56	60.75	2.57	373628	78901	21.12	Africa	5.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
Dominica	18	0	18	0	0	0	0	0.00	100.00	0.00	18	0	0.00	Americas	183.0
Saint Kitts and Nevis	17	0	15	2	0	0	0	0.00	88.24	0.00	17	0	0.00	Americas	184.0
Greenland	14	0	13	1	1	0	0	0.00	92.86	0.00	13	1	7.69	Europe	185.0
Holy See	12	0	12	0	0	0	0	0.00	100.00	0.00	12	0	0.00	Europe	186.0
/estern Sahara	10	1	8	1	0	0	0	10.00	80.00	12.50	10	0	0.00	Africa	187.0

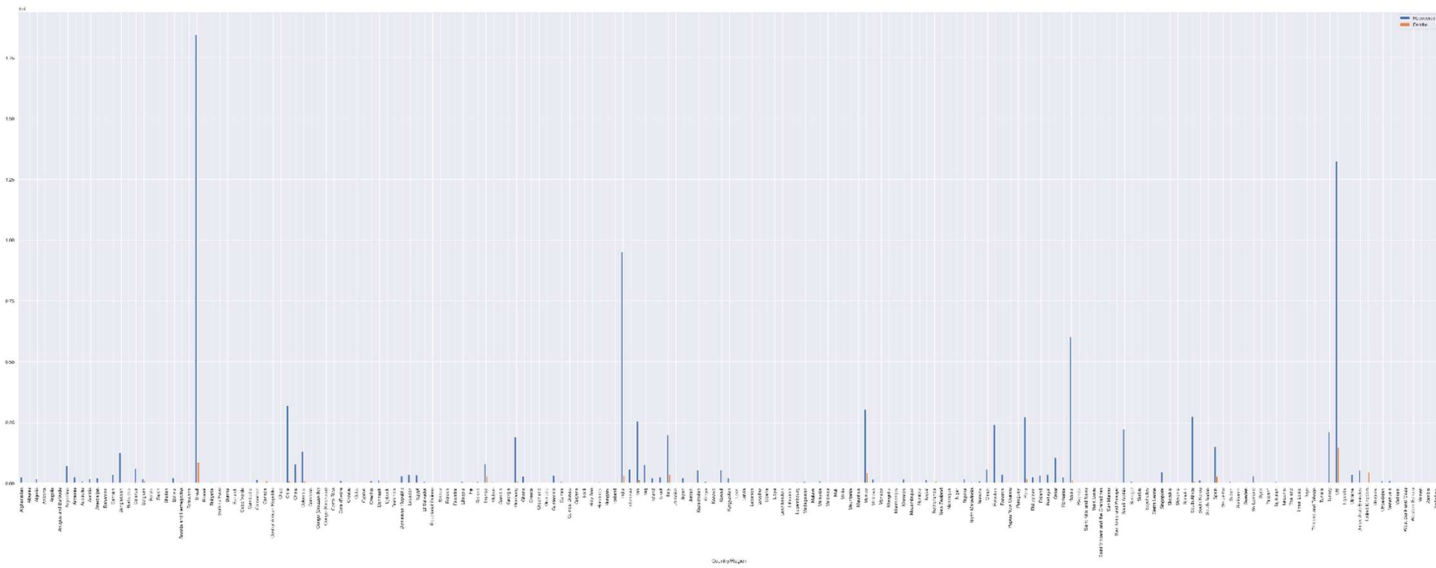
's × 16 columns

- Concluded that USA is the most affected country with 4290259 cases that is followed by Brazil [2442375 cases] and India [1480073 cases]
- However Western Sahara Region is least affected Region with only 10 confirmed cases.

Bar graph that shows Number of Recovered and Number of Deaths Country/Region wise

```
df.plot.bar(x='Country/Region',y=['Recovered','Deaths'],figsize=(60,20))
```

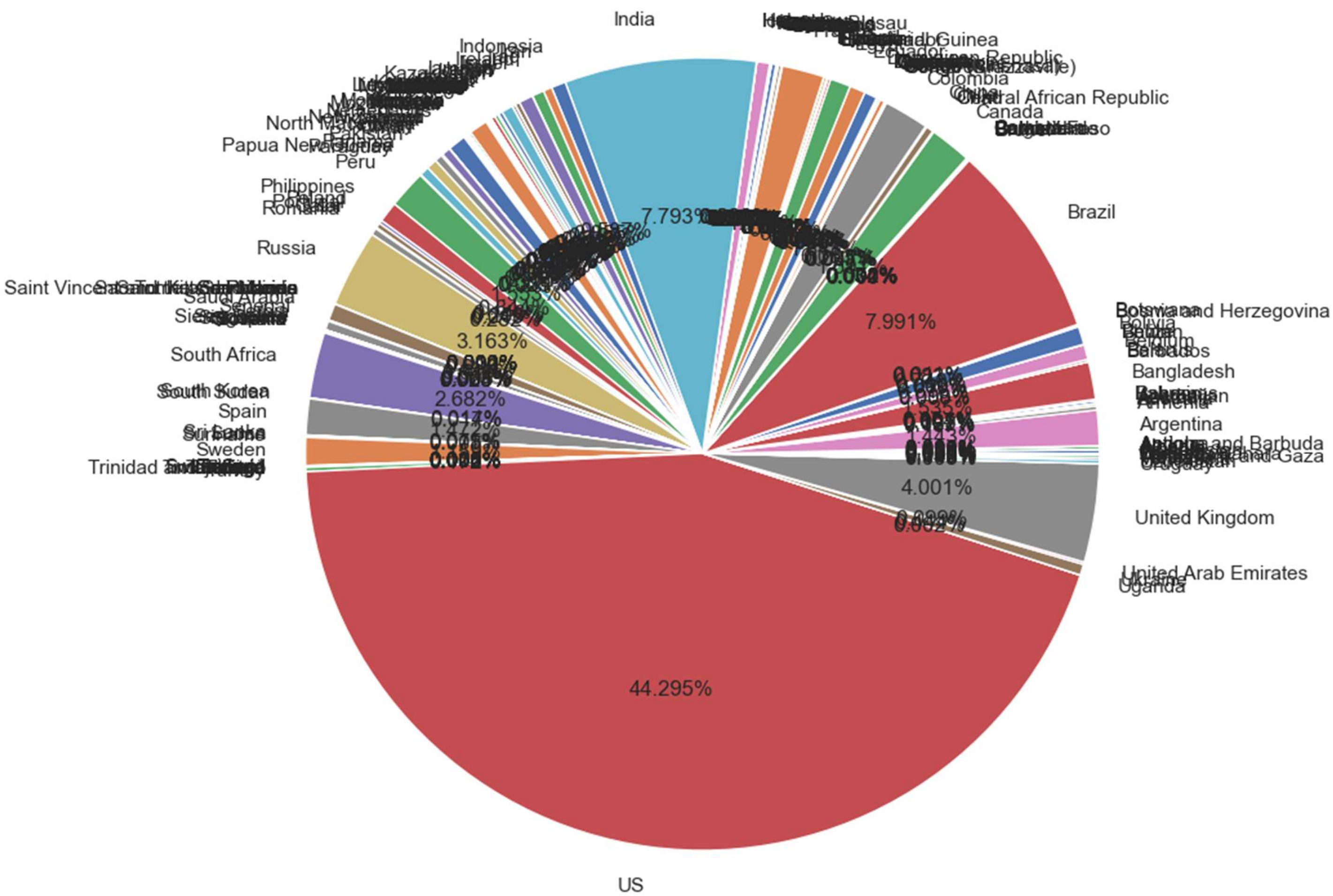
```
<Axes: xlabel='Country/Region'>
```



- As it can be seen by the graph Brazil shows the most of Recoveries which is followed by USA and then India.
- However most number of Deaths can be seen in USA.

Pie Chart to show % of Active cases country wise.

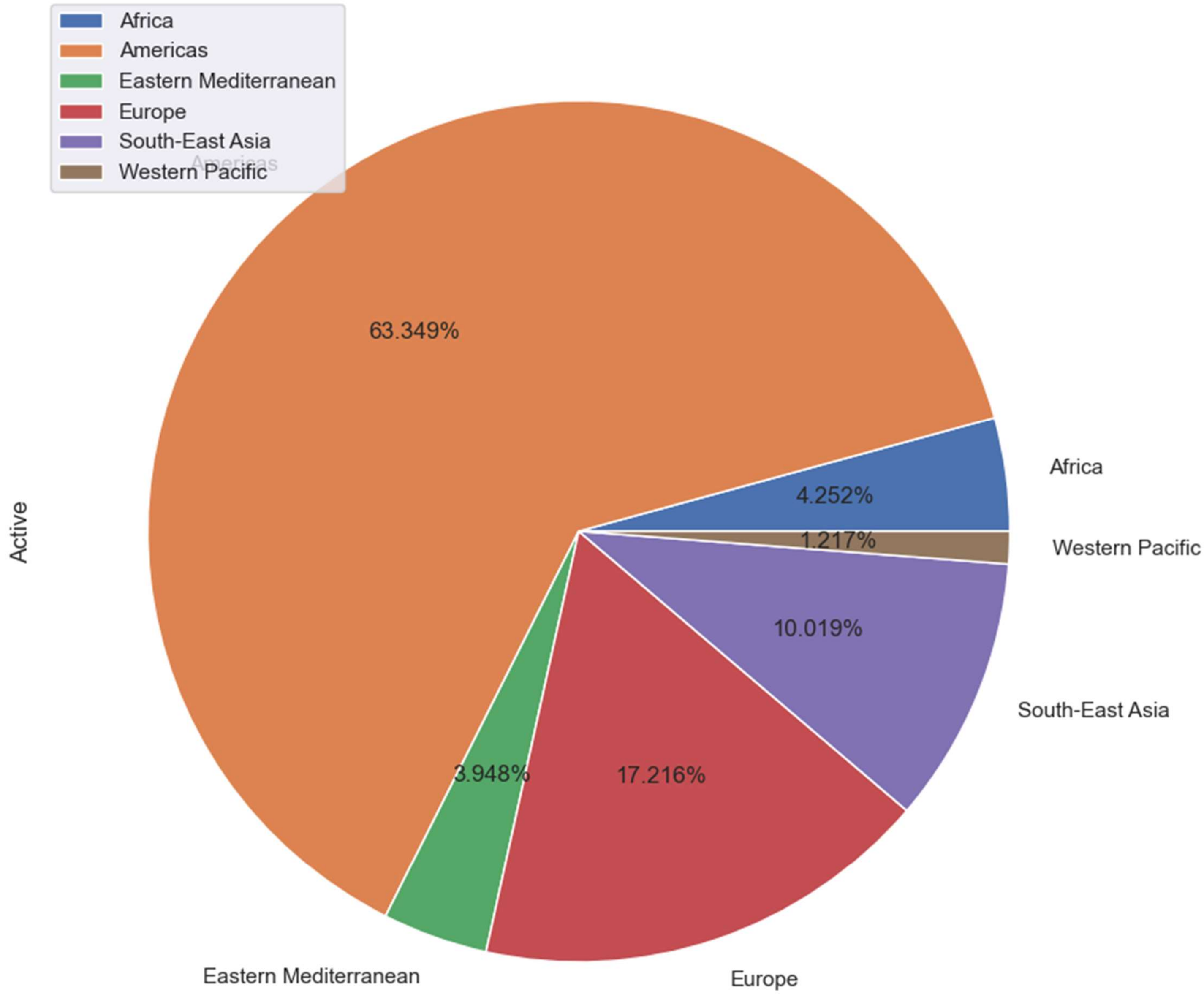
```
In [104]: plt.pie(df['Active'],labels=df['Country/Region'],autopct="%0.3f%%")
          plt.show()
```



- Highest % of Active cases is in USA i.e., 44.295%
- USA and Brazil together form nearly half of total number of Active cases all over the world.

Pie Chart to show % of Active cases WHO Region wise.

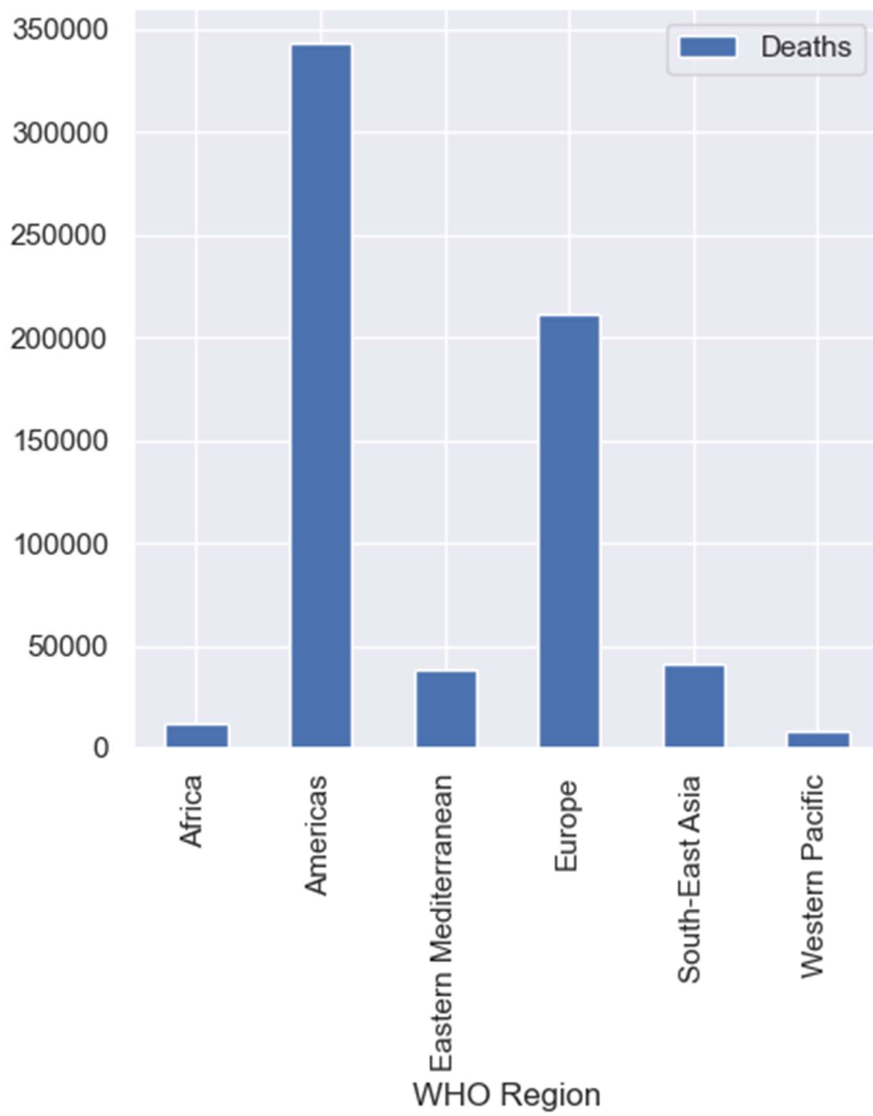
```
In [107]: df.groupby(['WHO Region']).sum().plot(kind='pie',y='Active',autopct="%0.3f%%")
```



- Highest % of Active Cases can be seen in Americas region i.e., 63.349% while Lowest % of Active Cases can be seen in Western Pacific Region i.e., 1.217%

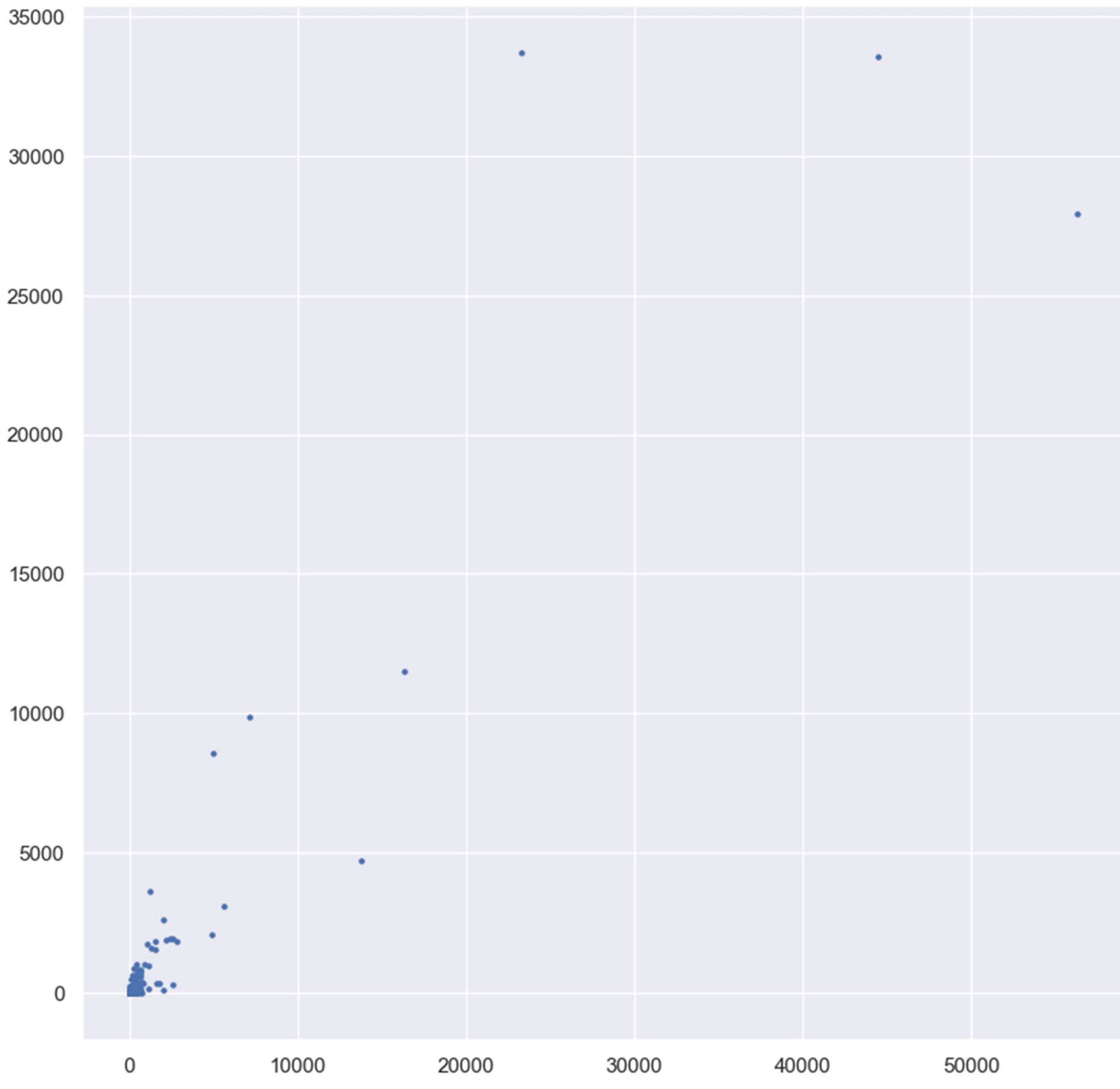
Bar graph to find the Number of Deaths per Region

```
In [106]: df.groupby(['WHO Region']).sum().plot(kind='bar',y='Deaths',figsize=(5,5))
```



Scatter plot to find the Relation between New Cases and New Deaths

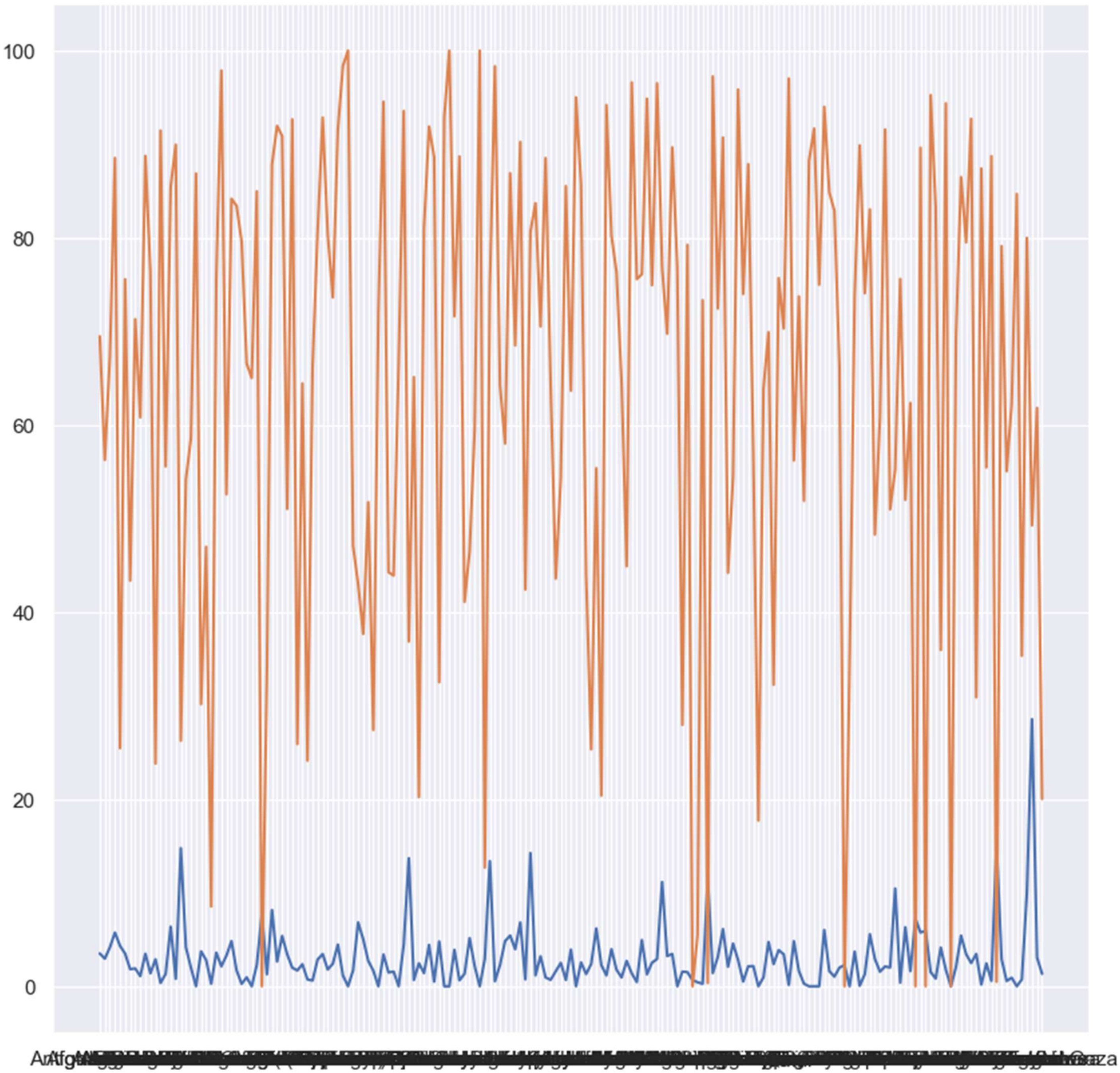
```
In [98]: x=df['New cases']
         y=df['New recovered']
         plt.scatter(x,y,s=5)

Out[98]: <matplotlib.collections.PathCollection at 0x1bedde1add0>
```



- As it can be clearly inferred from the Scatter plot that New Recoveries are much more than the New cases, which is a good sign

Line graph that depicts Death/100 cases and Recovery/100 cases country wise

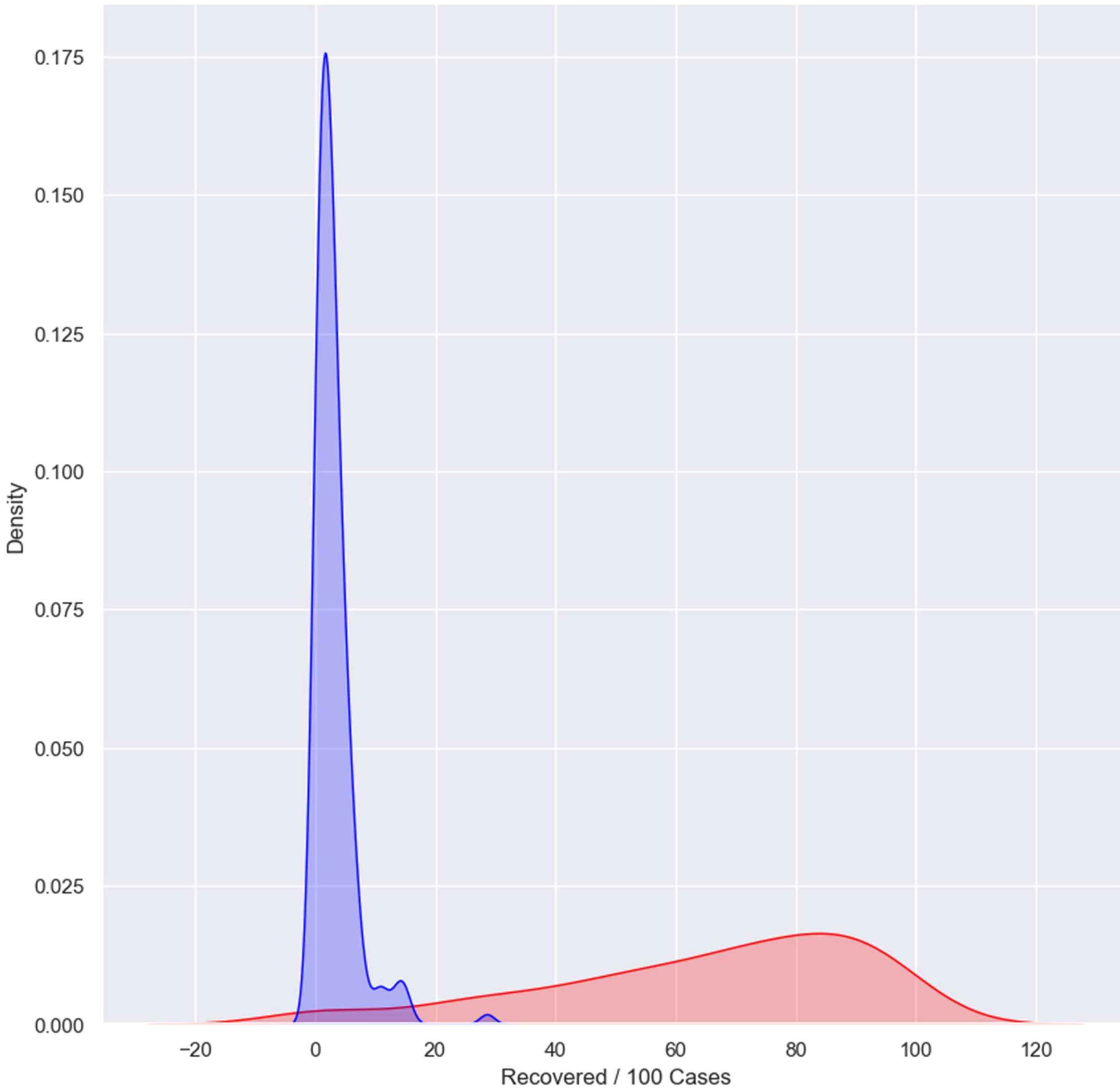
```
In [100]: n=df['Country/Region']
          x=df['Deaths / 100 Cases']
          y=df['Recovered / 100 Cases']
          plt.plot(n,x,label='Deaths/100 Cases')
          plt.plot(n,y,label='Recovered / 100 Cases')
          plt.show()
```



- As it can be clearly seen clearly that Recovery is much higher than the Deaths
- Orange Line graph depicts Recovery while Blue signifies Deaths

KDE Plot for Death/100 cases and Recovery/100 cases country wise

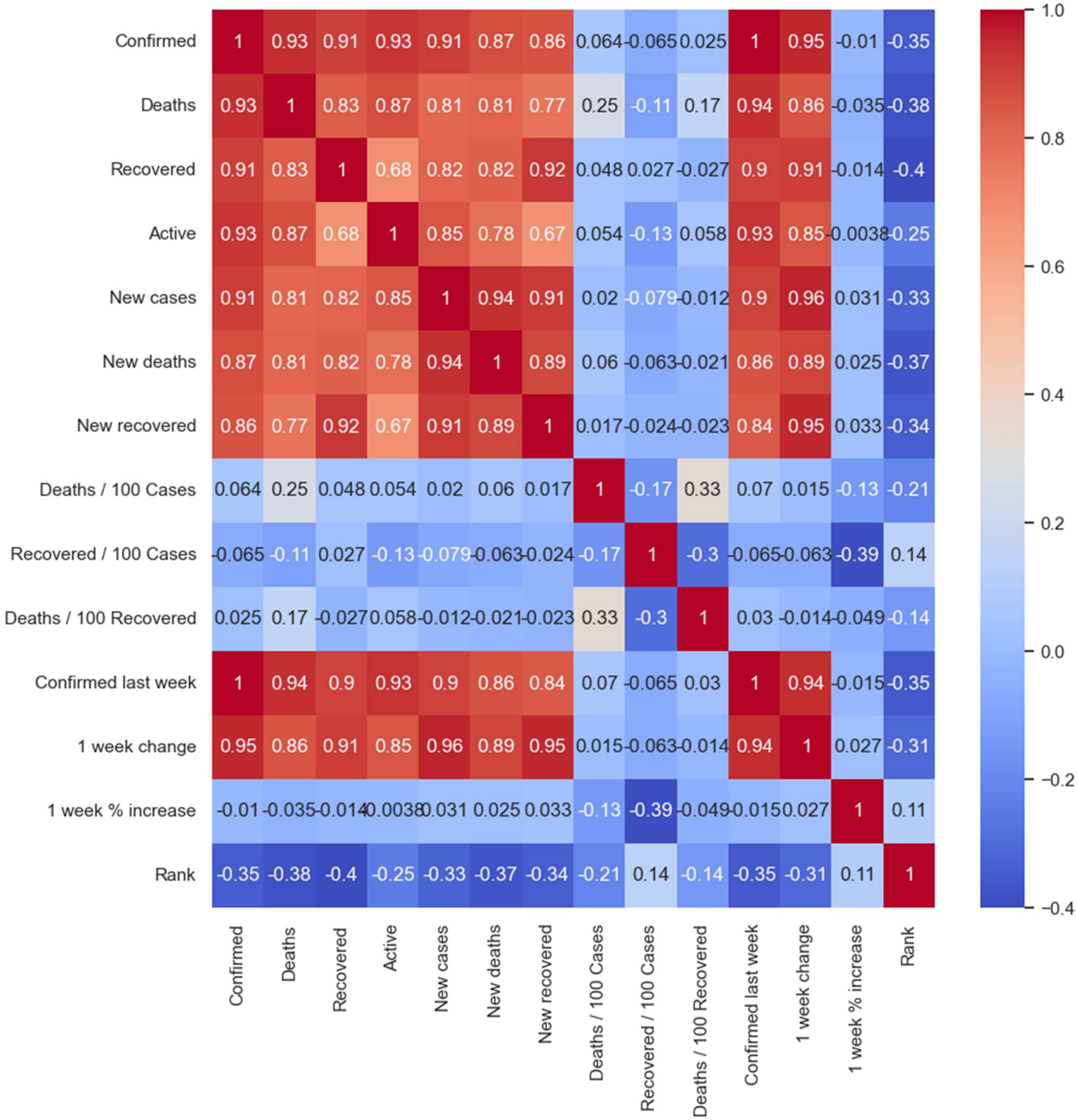
```
In [101]: x=df['Deaths / 100 Cases']
          y=df['Recovered / 100 Cases']
```



- KDE plot depicts the probably density of the Deaths and Recovered per 100 cases country wise
- If the peek is high and sharp, in KDE indicates that there is relatively high probablity of observing data points around particulatr value
- While a low peak in KDE indicates low concentration of data points.

Corelation Heat map using Seaborn

```
In [102]: sns.set(rc={'figure.figsize':(10,10)})
          corheat=sns.heatmap(df.corr(),annot=True,cmap='coolwarm')
```



- A corelation heat map shows relation between different groups, ranging from -1 to 1
- The data can be Linear or NonLinear
- The color intensity also signifies how strong the relation is between the two groups

Profiling the Data

```
In [48]: from ydata_profiling import ProfileReport
         pf=ProfileReport(df,title="Country wise Covid Report",explorative=True)
         pf
```

```
Summarize dataset: 100% 193/193 [00:28<00:00, 3.97it/s, Completed]
Generate report structure: 100% 1/1 [00:06<00:00, 6.53s/it]
Render HTML: 100% 1/1 [00:04<00:00, 4.34s/it]
```

- Pandas profiling provides an automated and comprehensive summary of a Data Frame
- .html file provided