Timeline Analysis Covid 19

Create a EDA showing spread of Covid-19 cases in your country.

Identify interesting patterns and possible reasons helping Covid-19 spreadwith basic as well as advanced charts.

Dataset: Daily updated .csv file on https://bit.ly/30d2gdi

Downloding and Installing Prerequisite

Install:

pip install pycountry_convert

pip install folium

In [1]:

```
# Install pycountry_convert
!pip install pycountry_convert
!pip install folium
#!pip install tensorflow`
!wget https://raw.githubusercontent.com/tarunk04/COVID-19-CaseStudy-and-
Predictions/master/models/model_deaths.h5
!wget https://raw.githubusercontent.com/tarunk04/COVID-19-CaseStudy-and-
Predictions/master/models/model_confirmed.h5
!wget https://raw.githubusercontent.com/tarunk04/COVID-19-CaseStudy-and-
Predictions/master/models/model_usa_c.h5
```

```
Requirement already satisfied: pycountry_convert in c:\users\hp\anaconda3\lib\site-packages
(0.7.2)
Requirement already satisfied: pprintpp>=0.3.0 in c:\users\hp\anaconda3\lib\site-packages (from
pycountry_convert) (0.4.0)
Requirement already satisfied: pytest-cov>=2.5.1 in c:\users\hp\anaconda3\lib\site-packages (from
pycountry_convert) (2.10.1)
Requirement already satisfied: repoze.lru>=0.7 in c:\users\hp\anaconda3\lib\site-packages (from
pycountry_convert) (0.7)
Requirement already satisfied: pytest-mock>=1.6.3 in c:\users\hp\anaconda3\lib\site-packages (from
pycountry_convert) (3.3.1)
Requirement already satisfied: pycountry>=16.11.27.1 in c:\users\hp\anaconda3\lib\site-packages (f
rom pycountry_convert) (20.7.3)
Requirement already satisfied: wheel>=0.30.0 in c:\users\hp\anaconda3\lib\site-packages (from
pycountry_convert) (0.34.2)
Requirement already satisfied: pytest>=3.4.0 in c:\users\hp\anaconda3\lib\site-packages (from
pycountry_convert) (5.3.5)
Requirement already satisfied: coverage>=4.4 in c:\users\hp\anaconda3\lib\site-packages (from
pytest-cov>=2.5.1->pycountry_convert) (5.3)
Requirement already satisfied: py>=1.5.0 in c:\users\hp\anaconda3\lib\site-packages (from
pytest>=3.4.0->pycountry_convert) (1.8.1)
Requirement already satisfied: packaging in c:\users\hp\anaconda3\lib\site-packages (from
pytest>=3.4.0->pycountry_convert) (19.0)
Requirement already satisfied: attrs>=17.4.0 in c:\users\hp\anaconda3\lib\site-packages (from
pytest>=3.4.0->pycountry_convert) (19.3.0)
Requirement already satisfied: more-itertools>=4.0.0 in c:\users\hp\anaconda3\lib\site-packages
(from pytest>=3.4.0->pycountry_convert) (8.2.0)
Requirement already satisfied: pluggy<1.0,>=0.12 in c:\users\hp\anaconda3\lib\site-packages (from
pytest>=3.4.0->pycountry_convert) (0.13.1)
Requirement already satisfied: wcwidth in c:\users\hp\anaconda3\lib\site-packages (from
pytest>=3.4.0->pycountry_convert) (0.1.8)
Requirement already satisfied: importlib-metadata>=0.12 in c:\users\hp\anaconda3\lib\site-packages
(from pytest>=3.4.0->pycountry_convert) (1.5.0)
Requirement already satisfied: atomicwrites>=1.0 in c:\users\hp\anaconda3\lib\site-packages (from
pytest>=3.4.0->pycountry_convert) (1.3.0)
Requirement already satisfied: colorama in c:\users\hp\anaconda3\lib\site-packages (from
pytest>=3.4.0->pycountry_convert) (0.4.3)
Requirement already satisfied: pyparsing>=2.0.2 in c:\users\hp\anaconda3\lib\site-packages (from p
```

```
ackaging->pytest>=3.4.0->pycountry_convert) (2.4.6)
Requirement already satisfied: six in c:\users\hp\anaconda3\lib\site-packages (from packaging-
>pytest>=3.4.0->pycountry_convert) (1.14.0)
Requirement already satisfied: zipp>=0.5 in c:\users\hp\anaconda3\lib\site-packages (from
importlib-metadata>=0.12->pytest>=3.4.0->pycountry_convert) (2.2.0)
Requirement already satisfied: folium in c:\users\hp\anaconda3\lib\site-packages (0.11.0)
Requirement already satisfied: numpy in c:\users\hp\anaconda3\lib\site-packages (from folium)
(1.18.1)
Requirement already satisfied: branca>=0.3.0 in c:\users\hp\anaconda3\lib\site-packages (from
folium) (0.4.1)
Requirement already satisfied: jinja2>=2.9 in c:\users\hp\anaconda3\lib\site-packages (from
folium) (2.11.1)
Requirement already satisfied: requests in c:\users\hp\anaconda3\lib\site-packages (from folium) (
2.24.0)
Requirement already satisfied: MarkupSafe>=0.23 in c:\users\hp\anaconda3\lib\site-packages (from j
inja2>=2.9->folium) (1.1.1)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in
c:\users\hp\anaconda3\lib\site-packages (from requests->folium) (1.24.3)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\hp\anaconda3\lib\site-packages (from
requests->folium) (2019.11.28)
Requirement already satisfied: idna<3,>=2.5 in c:\users\hp\anaconda3\lib\site-packages (from
requests->folium) (2.8)
Requirement already satisfied: chardet<4,>=3.0.2 in c:\users\hp\anaconda3\lib\site-packages (from
requests->folium) (3.0.4)
```

```
--2020-10-19 09:05:48--  https://raw.githubusercontent.com/tarunk04/COVID-19-CaseStudy-and-
Predictions/master/models/model_deaths.h5
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 199.232.20.133
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|199.232.20.133|:443...
connected.
HTTP request sent, awaiting response... 200 OK
Length: 111008 (108K) [application/octet-stream]
Saving to: 'model_deaths.h5.4'

     0K .......... .......... .......... .......... .......... 46%  133K 0s
    50K .......... .......... .......... .......... .......... 92%  202K 0s
   100K ........                                             100%  180K=0.7s

2020-10-19 09:06:21 (162 KB/s) - 'model_deaths.h5.4' saved [111008/111008]

--2020-10-19 09:06:21--  https://raw.githubusercontent.com/tarunk04/COVID-19-CaseStudy-and-
Predictions/master/models/model_confirmed.h5
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 199.232.20.133
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|199.232.20.133|:443...
connected.
HTTP request sent, awaiting response... 200 OK
Length: 111008 (108K) [application/octet-stream]
Saving to: 'model_confirmed.h5.4'

     0K .......... .......... .......... .......... .......... 46%  271K 0s
    50K .......... .......... .......... .......... .......... 92%  307K 0s
   100K ........                                             100%  363K=0.4s

2020-10-19 09:06:25 (293 KB/s) - 'model_confirmed.h5.4' saved [111008/111008]

--2020-10-19 09:06:25--  https://raw.githubusercontent.com/tarunk04/COVID-19-CaseStudy-and-
Predictions/master/models/model_usa_c.h5
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 199.232.20.133
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|199.232.20.133|:443...
connected.
HTTP request sent, awaiting response... 200 OK
Length: 111008 (108K) [application/octet-stream]
Saving to: 'model_usa_c.h5.4'

     0K .......... .......... .......... .......... .......... 46% 52.7K 1s
    50K .......... .......... .......... .......... .......... 92%  119K 0s
   100K ........                                             100%  122K=1.4s

2020-10-19 09:06:31 (75.4 KB/s) - 'model_usa_c.h5.4' saved [111008/111008]
```

Imports and Datasets

Pandas - for dataset handeling

Numpy - Support for Pandas and calculations

Matplotlib - for visualization (Platting graphas)

pycountry_convert - Library for getting continent (name) to from their country names

folium - Library for Map

keras - Prediction Models

plotly - for interative plots

In [2]:

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from matplotlib import ticker
import pycountry_convert as pc
import folium
import branca
from datetime import datetime, timedelta,date
from scipy.interpolate import make_interp_spline, BSpline
import plotly.express as px
import json, requests

#from keras.layers import Input, Dense, Activation, LeakyReLU
#from keras import models
#from keras.optimizers import RMSprop, Adam

%matplotlib inline
```

In [3]:

```python
df_confirmed = pd.read_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-
19/master/csse_covid_19_data/csse_covid_19_time_series/time_series_covid19_confirmed_global.csv')
df_deaths = pd.read_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-
19/master/csse_covid_19_data/csse_covid_19_time_series/time_series_covid19_deaths_global.csv')

# Depricated
# df_recovered = pd.read_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-
19/master/csse_covid_19_data/csse_covid_19_time_series/time_series_19-covid-Recovered.csv')
df_covid19 = pd.read_csv("https://raw.githubusercontent.com/CSSEGISandData/COVID-19/web-
data/data/cases_country.csv")
df_table = pd.read_csv("https://raw.githubusercontent.com/CSSEGISandData/COVID-19/web-
data/data/cases_time.csv",parse_dates=['Last_Update'])
```

```
C:\Users\hp\anaconda3\lib\site-packages\IPython\core\interactiveshell.py:3063: DtypeWarning:
Columns (11) have mixed types.Specify dtype option on import or set low_memory=False.
  interactivity=interactivity, compiler=compiler, result=result)
```

In [4]:

```python
# new dataset
df_covid19.head(2)
```

Out[4]:

	Country_Region	Last_Update	Lat	Long_	Confirmed	Deaths	Recovered	Active	Incident_Rate	People_Tested	People_H
0	Afghanistan	2020-10-19 03:24:46	33.93911	67.709953	40200.0	1492.0	33614.0	5094.0	103.266666	NaN	
1	Albania	2020-10-19 03:24:46	41.15330	20.168300	17055.0	451.0	10071.0	6533.0	592.640211	NaN	

In [5]:

```python
df_confirmed.head(2)
```

Out[5]:

	Province/State	Country/Region	Lat	Long	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	1/27/20	...	10/8/20	10/9/20	10/10/2(
0	NaN	Afghanistan	33.93911	67.709953	0	0	0	0	0	0	...	39616	39693	3970;
1	NaN	Albania	41.15330	20.168300	0	0	0	0	0	0	...	14899	15066	1523

2 rows × 274 columns

Preprocessing

In [6]:

```python
df_confirmed = df_confirmed.rename(columns={"Province/State":"state","Country/Region": "country"})
df_deaths = df_deaths.rename(columns={"Province/State":"state","Country/Region": "country"})
df_covid19 = df_covid19.rename(columns={"Country_Region": "country"})
df_covid19["Active"] = df_covid19["Confirmed"]-df_covid19["Recovered"]-df_covid19["Deaths"]
# df_recovered = df_recovered.rename(columns={"Province/State":"state","Country/Region":
"country"})
```

In [7]:

```python
# Changing the conuntry names as required by pycountry_convert Lib
df_confirmed.loc[df_confirmed['country'] == "US", "country"] = "USA"
df_deaths.loc[df_deaths['country'] == "US", "country"] = "USA"
df_covid19.loc[df_covid19['country'] == "US", "country"] = "USA"
df_table.loc[df_table['Country_Region'] == "US", "Country_Region"] = "USA"
# df_recovered.loc[df_recovered['country'] == "US", "country"] = "USA"


df_confirmed.loc[df_confirmed['country'] == 'Korea, South', "country"] = 'South Korea'
df_deaths.loc[df_deaths['country'] == 'Korea, South', "country"] = 'South Korea'
df_covid19.loc[df_covid19['country'] == "Korea, South", "country"] = "South Korea"
df_table.loc[df_table['Country_Region'] == "Korea, South", "Country_Region"] = "South Korea"
# df_recovered.loc[df_recovered['country'] == 'Korea, South', "country"] = 'South Korea'

df_confirmed.loc[df_confirmed['country'] == 'Taiwan*', "country"] = 'Taiwan'
df_deaths.loc[df_deaths['country'] == 'Taiwan*', "country"] = 'Taiwan'
df_covid19.loc[df_covid19['country'] == "Taiwan*", "country"] = "Taiwan"
df_table.loc[df_table['Country_Region'] == "Taiwan*", "Country_Region"] = "Taiwan"
# df_recovered.loc[df_recovered['country'] == 'Taiwan*', "country"] = 'Taiwan'

df_confirmed.loc[df_confirmed['country'] == 'Congo (Kinshasa)', "country"] = 'Democratic Republic
of the Congo'
df_deaths.loc[df_deaths['country'] == 'Congo (Kinshasa)', "country"] = 'Democratic Republic of the
Congo'
df_covid19.loc[df_covid19['country'] == "Congo (Kinshasa)", "country"] = "Democratic Republic of
the Congo"
df_table.loc[df_table['Country_Region'] == "Congo (Kinshasa)", "Country_Region"] = "Democratic Repu
blic of the Congo"
# df_recovered.loc[df_recovered['country'] == 'Congo (Kinshasa)', "country"] = 'Democratic Republi
c of the Congo'

df_confirmed.loc[df_confirmed['country'] == "Cote d'Ivoire", "country"] = "Côte d'Ivoire"
df_deaths.loc[df_deaths['country'] == "Cote d'Ivoire", "country"] = "Côte d'Ivoire"
df_covid19.loc[df_covid19['country'] == "Cote d'Ivoire", "country"] = "Côte d'Ivoire"
df_table.loc[df_table['Country_Region'] == "Cote d'Ivoire", "Country_Region"] = "Côte d'Ivoire"
# df_recovered.loc[df_recovered['country'] == "Cote d'Ivoire", "country"] = "Côte d'Ivoire"

df_confirmed.loc[df_confirmed['country'] == "Reunion", "country"] = "Réunion"
df_deaths.loc[df_deaths['country'] == "Reunion", "country"] = "Réunion"
df_covid19.loc[df_covid19['country'] == "Reunion", "country"] = "Réunion"
df_table.loc[df_table['Country_Region'] == "Reunion", "Country_Region"] = "Réunion"
# df_recovered.loc[df_recovered['country'] == "Reunion", "country"] = "Réunion"

df_confirmed.loc[df_confirmed['country'] == 'Congo (Brazzaville)', "country"] = 'Republic of the C
ongo'
df_deaths.loc[df_deaths['country'] == 'Congo (Brazzaville)', "country"] = 'Republic of the Congo'
df_covid19.loc[df_covid19['country'] == "Congo (Brazzaville)", "country"] = "Republic of the
Congo"
df_table.loc[df_table['Country_Region'] == "Congo (Brazzaville)", "Country_Region"] = "Republic of
the Congo"
# df_recovered.loc[df_recovered['country'] == 'Congo (Brazzaville)', "country"] = 'Republic of the
Congo'
```

```
Congo
df_confirmed.loc[df_confirmed['country'] == 'Bahamas, The', "country"] = 'Bahamas'
df_deaths.loc[df_deaths['country'] == 'Bahamas, The', "country"] = 'Bahamas'
df_covid19.loc[df_covid19['country'] == "Bahamas, The", "country"] = "Bahamas"
df_table.loc[df_table['Country_Region'] == "Bahamas, The", "Country_Region"] = "Bahamas"
# df_recovered.loc[df_recovered['country'] == 'Bahamas, The', "country"] = 'Bahamas'

df_confirmed.loc[df_confirmed['country'] == 'Gambia, The', "country"] = 'Gambia'
df_deaths.loc[df_deaths['country'] == 'Gambia, The', "country"] = 'Gambia'
df_covid19.loc[df_covid19['country'] == "Gambia, The", "country"] = "Gambia"
df_table.loc[df_table['Country_Region'] == "Gambia", "Country_Region"] = "Gambia"
# df_recovered.loc[df_recovered['country'] == 'Gambia, The', "country"] = 'Gambia'

# getting all countries
countries = np.asarray(df_confirmed["country"])
countries1 = np.asarray(df_covid19["country"])
# Continent_code to Continent_names
continents = {
    'NA': 'North America',
    'SA': 'South America',
    'AS': 'Asia',
    'OC': 'Australia',
    'AF': 'Africa',
    'EU' : 'Europe',
    'na' : 'Others'
}

# Defininng Function for getting continent code for country.
def country_to_continent_code(country):
    try:
        return pc.country_alpha2_to_continent_code(pc.country_name_to_country_alpha2(country))
    except :
        return 'na'

#Collecting Continent Information
df_confirmed.insert(2,"continent", [continents[country_to_continent_code(country)] for country in c
ountries[:]])
df_deaths.insert(2,"continent",  [continents[country_to_continent_code(country)] for country in cou
ntries[:]])
df_covid19.insert(1,"continent",  [continents[country_to_continent_code(country)] for country in co
untries1[:]])
df_table.insert(1,"continent",  [continents[country_to_continent_code(country)] for country in df_t
able["Country_Region"].values])
# df_recovered.insert(2,"continent",  [continents[country_to_continent_code(country)] for country
in countries[:]] )
```

In [8]:

```
df_table = df_table[df_table["continent"] != "Others"]
```

In [9]:

```
df_deaths[df_deaths["continent" ]== 'Others']
```

Out[9]:

	state	country	continent	Lat	Long	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	...	10/8/20	10/9/20	10/10/20	10/11/2
34	NaN	Burma	Others	21.916200	95.956000	0	0	0	0	0	...	535	566	598	64
102	NaN	Diamond Princess	Others	0.000000	0.000000	0	0	0	0	0	...	13	13	13	
139	NaN	Holy See	Others	41.902900	12.453400	0	0	0	0	0	...	0	0	0	
156	NaN	Kosovo	Others	42.602636	20.902977	0	0	0	0	0	...	638	638	645	64
168	NaN	MS Zaandam	Others	0.000000	0.000000	0	0	0	0	0	...	2	2	2	
238	NaN	Timor-Leste	Others	-8.874217	125.727539	0	0	0	0	0	...	0	0	0	
262	NaN	West Bank and Gaza	Others	31.952200	35.233200	0	0	0	0	0	...	359	367	378	38
263	NaN	Western	Others	24.215500	-12.885800	0	0	0	0	0	...	1	1	1	

8 rows × 275 columns

Defining Functions

plot_params()

visualize_covid_cases()

get_mortality_rate()

In [12]:

```python
def plot_params(ax,axis_label= None, plt_title = None,label_size=15, axis_fsize = 15, title_fsize =
20, scale = 'linear' ):
    # Tick-Parameters
    ax.xaxis.set_minor_locator(ticker.AutoMinorLocator())
    ax.yaxis.set_minor_locator(ticker.AutoMinorLocator())
    ax.tick_params(which='both', width=1,labelsize=label_size)
    ax.tick_params(which='major', length=6)
    ax.tick_params(which='minor', length=3, color='0.8')

    # Grid
    plt.grid(lw = 1, ls = '-', c = "0.7", which = 'major')
    plt.grid(lw = 1, ls = '-', c = "0.9", which = 'minor')

    # Plot Title
    plt.title( plt_title,{'fontsize':title_fsize})

    # Yaxis sacle
    plt.yscale(scale)
    plt.minorticks_on()
    # Plot Axes Labels
    xl = plt.xlabel(axis_label[0],fontsize = axis_fsize)
    yl = plt.ylabel(axis_label[1],fontsize = axis_fsize)

def visualize_covid_cases(confirmed, deaths, continent=None , country = None , state = None, period
= None, figure = None, scale = "linear"):
    x = 0
    if figure == None:
        f = plt.figure(figsize=(10,10))
        # Sub plot
        ax = f.add_subplot(111)
    else :
        f = figure[0]
        # Sub plot
        ax = f.add_subplot(figure[1],figure[2],figure[3])

    plt.tight_layout(pad=10, w_pad=5, h_pad=5)

    stats = [confirmed, deaths]
    label = ["Confirmed", "Deaths"]

    if continent != None:
        params = ["continent",continent]
    elif country != None:
        params = ["country",country]
    else:
        params = ["All", "All"]
    color = ["darkcyan","crimson"]
    marker_style = dict(linewidth=3, linestyle='-', marker='o',markersize=4, markerfacecolor='#ffff
ff')
    for i,stat in enumerate(stats):
        if params[1] == "All" :
            cases = np.sum(np.asarray(stat.iloc[:,5:]),axis = 0)[x:]
        else :
            cases = np.sum(np.asarray(stat[stat[params[0]] == params[1]].iloc[:,5:]),axis = 0)[x:]
        date = np.arange(1,cases.shape[0]+1)[x:]
        plt.plot(date,cases,label = label[i]+" (Total : "+str(cases[-1])+")",color=color[i],**marke
r_style)

    if params[1] == "All" :
        Total_confirmed = np.sum(np.asarray(stats[0].iloc[:,5:]),axis = 0)[x:]
```

```python
        Total_deaths = np.sum(np.asarray(stats[1].iloc[:,5:]),axis = 0)[x:]
    else :
        Total_confirmed =  np.sum(np.asarray(stats[0][stat[params[0]] == params[1]].iloc[:,5:]),axi
s = 0)[x:]
        Total_deaths = np.sum(np.asarray(stats[1][stat[params[0]] == params[1]].iloc[:,5:]),axis =
0)[x:]

    text = "From "+stats[0].columns[5]+" to "+stats[0].columns[-1]+"\n"
    text += "Mortality rate : "+ str(int(Total_deaths[-1]/(Total_confirmed[-1])*10000)/100)+"\n"
    text += "Last 5 Days:\n"
    text += "Confirmed : " + str(Total_confirmed[-1] - Total_confirmed[-6])+"\n"
    text += "Deaths : " + str(Total_deaths[-1] - Total_deaths[-6])+"\n"
    text += "Last 24 Hours:\n"
    text += "Confirmed : " + str(Total_confirmed[-1] - Total_confirmed[-2])+"\n"
    text += "Deaths : " + str(Total_deaths[-1] - Total_deaths[-2])+"\n"

    plt.text(0.02, 0.78, text, fontsize=15, horizontalalignment='left', verticalalignment='top', tr
ansform=ax.transAxes,bbox=dict(facecolor='white', alpha=0.4))

    # Plot Axes Labels
    axis_label = ["Days ("+df_confirmed.columns[5]+" - "+df_confirmed.columns[-1]+")","No of Cases"
]

    # Plot Parameters
    plot_params(ax,axis_label,scale = scale)

    # Plot Title
    if params[1] == "All" :
        plt.title("COVID-19 Cases World",{'fontsize':25})
    else:
        plt.title("COVID-19 Cases for "+params[1] ,{'fontsize':25})

    # Legend Location
    l = plt.legend(loc= "best",fontsize = 15)

    if figure == None:
        plt.show()
def get_total_cases(cases, country = "All"):
    if(country == "All") :
        return np.sum(np.asarray(cases.iloc[:,5:]),axis = 0)[-1]
    else :
        return np.sum(np.asarray(cases[cases["country"] == country].iloc[:,5:]),axis = 0)[-1]

def get_mortality_rate(confirmed,deaths, continent = None, country = None):
    if continent != None:
        params = ["continent",continent]
    elif country != None:
        params = ["country",country]
    else :
        params = ["All", "All"]

    if params[1] == "All" :
        Total_confirmed = np.sum(np.asarray(confirmed.iloc[:,5:]),axis = 0)
        Total_deaths = np.sum(np.asarray(deaths.iloc[:,5:]),axis = 0)
        mortality_rate = np.round((Total_deaths/Total_confirmed)*100,2)
    else :
        Total_confirmed =  np.sum(np.asarray(confirmed[confirmed[params[0]] == params[1]].iloc[:,5:
]),axis = 0)
        Total_deaths = np.sum(np.asarray(deaths[deaths[params[0]] == params[1]].iloc[:,5:]),axis =
0)
        mortality_rate = np.round((Total_deaths/Total_confirmed)*100,2)

    return np.nan_to_num(mortality_rate)
def dd(date1,date2):
    return (datetime.strptime(date1,'%m/%d/%y') - datetime.strptime(date2,'%m/%d/%y')).days


out = ""+"output/"
```

General Analysis of Data

Getting country wise and continent wise data.

```python
df_countries_cases = df_covid19.copy().drop(['Lat','Long_','continent','Last_Update'],axis =1)
df_countries_cases.index = df_countries_cases["country"]
df_countries_cases = df_countries_cases.drop(['country'],axis=1)

df_continents_cases = df_covid19.copy().drop(['Lat','Long_','country','Last_Update'],axis =1)
df_continents_cases = df_continents_cases.groupby(["continent"]).sum()
```

Global Reported Cases till Date

Total number of confirmed cases, deaths reported, revoveries and active cases all across the world

In [14]:

```python
pd.DataFrame(df_countries_cases.sum()).transpose().style.background_gradient(cmap='Wistia',axis=1)
```

Out[14]:

	Confirmed	Deaths	Recovered	Active	Incident_Rate	People_Tested	People_Hospitalized	Mortality_
0	39898689.000000	1112588.000000	27420679.000000	11232746.000000	122521.770540	0.000000	0.000000	465.22

Coninent Wise Reported Cases

Coninent Wise reported confirmed cases, recovered cases, deaths, active cases

In [15]:

```python
df_continents_cases.style.background_gradient(cmap='Wistia')
```

Out[15]:

continent	Confirmed	Deaths	Recovered	Active	Incident_Rate	People_Tested	People_Hospitalized	Morta
Africa	1646268.000000	39746.000000	1354438.000000	252084.000000	9519.943977	0.000000	0.000000	1
Asia	12379254.000000	222138.000000	10770245.000000	1386871.000000	33300.342191	0.000000	0.000000	9
Australia	29901.000000	939.000000	27502.000000	1457.000000	157.250603	0.000000	0.000000	1
Europe	6954084.000000	238929.000000	2839114.000000	3743375.000000	43081.617670	0.000000	0.000000	10
North America	9811606.000000	329569.000000	4565349.000000	4916688.000000	14906.866313	0.000000	0.000000	5
Others	100838.000000	1957.000000	72945.000000	25929.000000	5264.512742	0.000000	0.000000	4
South America	8976738.000000	279310.000000	7791086.000000	906342.000000	16291.237045	0.000000	0.000000	3

Country Wise Reported Cases

Country Wise reported confirmed cases, recovered cases, deaths, active cases

In [16]:

```python
df_countries_cases.sort_values('Confirmed', ascending= False).style.background_gradient(cmap='Wistia')
```

```
C:\Users\hp\anaconda3\lib\site-packages\matplotlib\colors.py:527: RuntimeWarning: invalid value
encountered in less
  xa[xa < 0] = -1
C:\Users\hp\anaconda3\lib\site-packages\pandas\io\formats\style.py:1089: RuntimeWarning: All-NaN s
lice encountered
  smin = np.nanmin(s.to_numpy()) if vmin is None else vmin
C:\Users\hp\anaconda3\lib\site-packages\pandas\io\formats\style.py:1090: RuntimeWarning: All-NaN s
lice encountered
  smax = np.nanmax(s.to_numpy()) if vmax is None else vmax
```

Out[16]:

country	Confirmed	Deaths	Recovered	Active	Incident_Rate	People_Tested	People_Hospitalized	Mor
USA	8154206.000000	219672.000000	3234138.000000	4700396.000000	2474.974351	nan	nan	
India	7494551.000000	114031.000000	6597209.000000	783311.000000	543.081680	nan	nan	
Brazil	5224362.000000	153675.000000	4526393.000000	544294.000000	2457.836153	nan	nan	
Russia	1390824.000000	24039.000000	1065608.000000	301177.000000	953.047005	nan	nan	
Argentina	989680.000000	26267.000000	803965.000000	159448.000000	2189.762110	nan	nan	
Colombia	959572.000000	28970.000000	858294.000000	72308.000000	1885.844364	nan	nan	
Spain	936560.000000	33775.000000	150376.000000	752409.000000	2003.131958	nan	nan	
France	876342.000000	33325.000000	108014.000000	735003.000000	1342.569096	nan	nan	
Peru	865549.000000	33702.000000	774356.000000	57491.000000	2625.115379	nan	nan	
Mexico	851227.000000	86167.000000	720973.000000	44087.000000	666.102021	nan	nan	
United Kingdom	725292.000000	43736.000000	2589.000000	678967.000000	1068.396956	nan	nan	
South Africa	703793.000000	18471.000000	634543.000000	50779.000000	1186.660842	nan	nan	
Iran	530380.000000	30375.000000	427400.000000	72605.000000	631.457737	nan	nan	
Chile	491760.000000	13635.000000	463943.000000	14182.000000	2572.476583	nan	nan	
Iraq	426634.000000	10254.000000	360477.000000	55903.000000	1060.684861	nan	nan	
Italy	414241.000000	36543.000000	251461.000000	126237.000000	685.128144	nan	nan	
Bangladesh	388569.000000	5660.000000	303972.000000	78937.000000	235.940528	nan	nan	
Germany	368671.000000	9794.000000	293220.000000	65657.000000	440.025831	nan	nan	
Indonesia	361867.000000	12511.000000	285324.000000	64032.000000	132.298263	nan	nan	
Philippines	356618.000000	6652.000000	310158.000000	39808.000000	325.437552	nan	nan	
Turkey	347493.000000	9296.000000	304003.000000	34194.000000	412.019023	nan	nan	
Saudi Arabia	342202.000000	5185.000000	328538.000000	8479.000000	982.947399	nan	nan	
Pakistan	323452.000000	6659.000000	307409.000000	9384.000000	146.429710	nan	nan	
Ukraine	307301.000000	5762.000000	130338.000000	171201.000000	702.663130	nan	nan	
Israel	303109.000000	2209.000000	268093.000000	32807.000000	3501.907044	nan	nan	
Netherlands	234203.000000	6811.000000	5206.000000	222186.000000	1366.820752	nan	nan	
Belgium	222253.000000	10413.000000	21157.000000	190683.000000	1917.690802	nan	nan	
Canada	200804.000000	9816.000000	169751.000000	21237.000000	530.445849	nan	nan	
Romania	180388.000000	5872.000000	130894.000000	43622.000000	937.680538	nan	nan	
Poland	175766.000000	3573.000000	92651.000000	79542.000000	464.416822	nan	nan	
Czechia	173885.000000	1422.000000	72134.000000	100329.000000	1623.730435	nan	nan	
Morocco	173632.000000	2928.000000	143972.000000	26732.000000	470.412829	nan	nan	
Ecuador	153289.000000	12387.000000	134187.000000	6715.000000	868.834545	nan	nan	
Bolivia	139771.000000	8481.000000	104483.000000	26807.000000	1197.384158	nan	nan	
Nepal	132246.000000	739.000000	92166.000000	39341.000000	453.879505	nan	nan	
Qatar	129431.000000	224.000000	126406.000000	2801.000000	4492.478463	nan	nan	
Panama	124745.000000	2564.000000	101041.000000	21140.000000	2891.117205	nan	nan	
Dominican Republic	121347.000000	2199.000000	98207.000000	20941.000000	1118.621625	nan	nan	
Kuwait	116146.000000	701.000000	107860.000000	7585.000000	2719.688247	nan	nan	
United Arab Emirates	115602.000000	463.000000	107516.000000	7623.000000	1168.830381	nan	nan	
Oman	109953.000000	1101.000000	95624.000000	13228.000000	2153.145465	nan	nan	
Kazakhstan	109508.000000	1768.000000	105001.000000	2739.000000	583.211955	nan	nan	
Egypt	105424.000000	6120.000000	98247.000000	1057.000000	103.019119	nan	nan	
Sweden	103200.000000	5918.000000	nan	nan	1021.856035	nan	nan	

	Confirmed	Deaths	Recovered	Active	Incident_Rate	People_Tested	People_Hospitalized	Mor
Guatemala	101360.000000	3530.000000	90610.000000	7220.000000	565.764957	nan	nan	
Portugal	99911.000000	2181.000000	59000.000000	38730.000000	979.835941	nan	nan	
Costa Rica	95514.000000	1183.000000	58816.000000	35515.000000	1874.987486	nan	nan	
Japan	93098.000000	1672.000000	84461.000000	6965.000000	73.608956	nan	nan	
China	90972.000000	4739.000000	85819.000000	414.000000	6.476367	nan	nan	
Ethiopia	89137.000000	1352.000000	42649.000000	45136.000000	77.534988	nan	nan	
Belarus	87698.000000	929.000000	79757.000000	7012.000000	928.087849	nan	nan	
Honduras	87594.000000	2563.000000	34662.000000	50369.000000	884.376242	nan	nan	
Venezuela	86636.000000	736.000000	79694.000000	6206.000000	304.670747	nan	nan	
Bahrain	77902.000000	300.000000	74320.000000	3282.000000	4578.207469	nan	nan	
Switzerland	74422.000000	2123.000000	50600.000000	21699.000000	859.910859	nan	nan	
Moldova	67050.000000	1584.000000	47842.000000	17624.000000	1662.137208	nan	nan	
Austria	64806.000000	893.000000	49561.000000	14352.000000	719.554983	nan	nan	
Armenia	64694.000000	1081.000000	48104.000000	15509.000000	2183.222790	nan	nan	
Uzbekistan	63124.000000	525.000000	60080.000000	2519.000000	188.603259	nan	nan	
Lebanon	62286.000000	520.000000	28062.000000	33704.000000	912.556286	nan	nan	
Nigeria	61440.000000	1125.000000	56611.000000	3704.000000	29.805047	nan	nan	
Singapore	57911.000000	28.000000	57807.000000	76.000000	989.873585	nan	nan	
Paraguay	54724.000000	1188.000000	36068.000000	17468.000000	767.245283	nan	nan	
Algeria	54402.000000	1856.000000	38088.000000	14458.000000	124.060903	nan	nan	
Kyrgyzstan	52044.000000	1111.000000	45736.000000	5197.000000	797.708099	nan	nan	
Ireland	49962.000000	1852.000000	23364.000000	24746.000000	1011.827949	nan	nan	
Libya	48790.000000	725.000000	26889.000000	21176.000000	710.056209	nan	nan	
Ghana	47310.000000	310.000000	46618.000000	382.000000	152.254638	nan	nan	
West Bank and Gaza	47135.000000	408.000000	40498.000000	6229.000000	923.959152	nan	nan	
Hungary	46290.000000	1142.000000	14088.000000	31060.000000	479.175185	nan	nan	
Azerbaijan	44964.000000	626.000000	40037.000000	4301.000000	443.468034	nan	nan	
Kenya	44881.000000	832.000000	31857.000000	12192.000000	83.466459	nan	nan	
Tunisia	40542.000000	626.000000	5032.000000	34884.000000	343.035032	nan	nan	
Afghanistan	40200.000000	1492.000000	33614.000000	5094.000000	103.266666	nan	nan	
Jordan	37573.000000	345.000000	6912.000000	30316.000000	368.249382	nan	nan	
Serbia	36160.000000	776.000000	nan	nan	413.854512	nan	nan	
Burma	36025.000000	880.000000	17076.000000	18069.000000	66.210506	nan	nan	
Denmark	35893.000000	680.000000	29741.000000	5472.000000	619.677867	nan	nan	
Bosnia and Herzegovina	34112.000000	984.000000	24995.000000	8133.000000	1039.741650	nan	nan	
El Salvador	31666.000000	922.000000	27000.000000	3744.000000	488.205654	nan	nan	
Slovakia	29835.000000	88.000000	7359.000000	22388.000000	546.464302	nan	nan	
Bulgaria	29503.000000	986.000000	16943.000000	11574.000000	424.598597	nan	nan	
Australia	27399.000000	905.000000	25108.000000	1386.000000	107.617136	nan	nan	
Croatia	25580.000000	363.000000	20053.000000	5164.000000	623.101829	nan	nan	
Greece	25370.000000	509.000000	1347.000000	23514.000000	243.402703	nan	nan	
South Korea	25275.000000	444.000000	23368.000000	1463.000000	49.298621	nan	nan	
North Macedonia	23628.000000	834.000000	17239.000000	5555.000000	1134.118596	nan	nan	
Cameroon	21441.000000	423.000000	20117.000000	901.000000	80.769645	nan	nan	
Malaysia	20498.000000	187.000000	13262.000000	7049.000000	63.331895	nan	nan	
Côte d'Ivoire	20323.000000	121.000000	20021.000000	181.000000	77.044462	nan	nan	
Georgia	17477.000000	136.000000	8060.000000	9281.000000	438.110637	nan	nan	
Albania	17055.000000	451.000000	10071.000000	6533.000000	592.640211	nan	nan	
Kosovo	16891.000000	653.000000	14661.000000	1577.000000	933.015755	nan	nan	

country	Confirmed	Deaths	Recovered	Active	Incident_Rate	People_Tested	People_Hospitalized	Mor...
Madagascar	16810.000000	238.000000	16215.000000	357.000000	60.705603	nan	nan	nan
Norway	16457.000000	278.000000	11863.000000	4316.000000	303.565124	nan	nan	
Zambia	15853.000000	346.000000	15005.000000	502.000000	86.232800	nan	nan	
Montenegro	15615.000000	236.000000	10994.000000	4385.000000	2486.219513	nan	nan	
Senegal	15418.000000	317.000000	13814.000000	1287.000000	92.081130	nan	nan	
Sudan	13697.000000	836.000000	6764.000000	6097.000000	31.236553	nan	nan	
Finland	13424.000000	351.000000	9100.000000	3973.000000	242.279069	nan	nan	
Slovenia	13142.000000	188.000000	6313.000000	6641.000000	632.151509	nan	nan	
Namibia	12293.000000	131.000000	10422.000000	1740.000000	483.801905	nan	nan	
Guinea	11518.000000	70.000000	10427.000000	1021.000000	87.704123	nan	nan	
Maldives	11210.000000	37.000000	10164.000000	1009.000000	2073.844401	nan	nan	
Democratic Republic of the Congo	11006.000000	302.000000	10356.000000	348.000000	12.288776	nan	nan	
Luxembourg	10888.000000	133.000000	8468.000000	2287.000000	1739.363809	nan	nan	
Mozambique	10866.000000	75.000000	8513.000000	2278.000000	34.765154	nan	nan	
Uganda	10590.000000	97.000000	6992.000000	3501.000000	23.152095	nan	nan	
Tajikistan	10493.000000	80.000000	9511.000000	902.000000	110.016711	nan	nan	
Haiti	8964.000000	231.000000	7213.000000	1520.000000	78.614111	nan	nan	
Gabon	8881.000000	54.000000	8430.000000	397.000000	399.015513	nan	nan	
Jamaica	8274.000000	171.000000	3859.000000	4244.000000	279.417431	nan	nan	
Zimbabwe	8147.000000	231.000000	7678.000000	238.000000	54.814237	nan	nan	
Cabo Verde	7752.000000	86.000000	6526.000000	1140.000000	1394.274697	nan	nan	
Angola	7622.000000	247.000000	3030.000000	4345.000000	23.190951	nan	nan	
Mauritania	7608.000000	163.000000	7347.000000	98.000000	163.624867	nan	nan	
Lithuania	7521.000000	113.000000	3097.000000	4311.000000	276.274652	nan	nan	
Cuba	6220.000000	125.000000	5768.000000	327.000000	54.914901	nan	nan	
Malawi	5857.000000	181.000000	4742.000000	934.000000	30.616904	nan	nan	
Eswatini	5780.000000	116.000000	5415.000000	249.000000	498.205426	nan	nan	
Bahamas	5703.000000	122.000000	3300.000000	2281.000000	1450.229880	nan	nan	
Sri Lanka	5538.000000	13.000000	3403.000000	2122.000000	25.862492	nan	nan	
Djibouti	5459.000000	61.000000	5375.000000	23.000000	552.529246	nan	nan	
Nicaragua	5353.000000	154.000000	4225.000000	974.000000	80.805440	nan	nan	
Trinidad and Tobago	5297.000000	96.000000	3652.000000	1549.000000	378.494753	nan	nan	
Botswana	5242.000000	20.000000	905.000000	4317.000000	222.909690	nan	nan	
Republic of the Congo	5156.000000	92.000000	3887.000000	1177.000000	93.438094	nan	nan	
Suriname	5130.000000	109.000000	4944.000000	77.000000	874.480511	nan	nan	
Syria	5077.000000	248.000000	1528.000000	3301.000000	29.010339	nan	nan	
Equatorial Guinea	5070.000000	83.000000	4954.000000	33.000000	361.372360	nan	nan	
Rwanda	4974.000000	34.000000	4783.000000	157.000000	38.402716	nan	nan	
Central African Republic	4855.000000	62.000000	1924.000000	2869.000000	100.522510	nan	nan	
Malta	4628.000000	45.000000	3236.000000	1347.000000	1048.152032	nan	nan	
Estonia	4078.000000	68.000000	3211.000000	799.000000	307.416518	nan	nan	
Iceland	4055.000000	11.000000	2804.000000	1240.000000	1188.278388	nan	nan	
Somalia	3864.000000	99.000000	3089.000000	676.000000	24.312255	nan	nan	
Guyana	3734.000000	109.000000	2654.000000	971.000000	474.725990	nan	nan	
Thailand	3686.000000	59.000000	3481.000000	146.000000	5.280804	nan	nan	
Gambia	3649.000000	118.000000	2649.000000	882.000000	150.993270	nan	nan	

country	Confirmed	Deaths	Recovered	Active	Incident_Rate	People_Tested	People_Hospitalized	Mor
Latvia	3450.000000	44.000000	1329.000000	2077.000000	182.907239	nan	nan	
Mali	3388.000000	132.000000	2586.000000	670.000000	16.730175	nan	nan	
Andorra	3377.000000	59.000000	2057.000000	1261.000000	4370.672361	nan	nan	
South Sudan	2842.000000	55.000000	1290.000000	1497.000000	25.389216	nan	nan	
Belize	2813.000000	44.000000	1670.000000	1099.000000	707.457604	nan	nan	
Cyprus	2644.000000	25.000000	1444.000000	1175.000000	218.990012	nan	nan	
Uruguay	2531.000000	51.000000	2105.000000	375.000000	72.861224	nan	nan	
Benin	2496.000000	41.000000	2330.000000	125.000000	20.588627	nan	nan	
Guinea-Bissau	2389.000000	41.000000	1782.000000	566.000000	121.392400	nan	nan	
Burkina Faso	2381.000000	65.000000	1774.000000	542.000000	11.390558	nan	nan	
Sierra Leone	2330.000000	73.000000	1760.000000	497.000000	29.209031	nan	nan	
Togo	2057.000000	51.000000	1531.000000	475.000000	24.846785	nan	nan	
Yemen	2056.000000	597.000000	1338.000000	121.000000	6.893322	nan	nan	
New Zealand	1886.000000	25.000000	1824.000000	37.000000	39.110512	nan	nan	
Lesotho	1833.000000	42.000000	961.000000	830.000000	85.564163	nan	nan	
Chad	1379.000000	93.000000	1181.000000	105.000000	8.395299	nan	nan	
Liberia	1377.000000	82.000000	1268.000000	27.000000	27.225938	nan	nan	
Niger	1210.000000	69.000000	1126.000000	15.000000	4.998629	nan	nan	
Vietnam	1134.000000	35.000000	1031.000000	68.000000	1.165006	nan	nan	
Sao Tome and Principe	933.000000	15.000000	898.000000	20.000000	425.714429	nan	nan	
San Marino	759.000000	42.000000	685.000000	32.000000	2236.431139	nan	nan	
Diamond Princess	712.000000	13.000000	659.000000	40.000000	nan	nan	nan	
Papua New Guinea	581.000000	7.000000	540.000000	34.000000	6.493777	nan	nan	
Burundi	542.000000	1.000000	497.000000	44.000000	4.558153	nan	nan	
Taiwan	535.000000	7.000000	491.000000	37.000000	2.246316	nan	nan	
Tanzania	509.000000	21.000000	183.000000	305.000000	0.852108	nan	nan	
Comoros	502.000000	7.000000	485.000000	10.000000	57.728023	nan	nan	
Eritrea	452.000000	0.000000	388.000000	64.000000	12.745222	nan	nan	
Mauritius	417.000000	10.000000	364.000000	43.000000	32.789025	nan	nan	
Bhutan	325.000000	0.000000	299.000000	26.000000	42.119615	nan	nan	
Mongolia	324.000000	0.000000	312.000000	12.000000	9.883195	nan	nan	
Cambodia	283.000000	0.000000	280.000000	3.000000	1.692688	nan	nan	
Monaco	265.000000	2.000000	217.000000	46.000000	675.262461	nan	nan	
Liechtenstein	224.000000	1.000000	132.000000	91.000000	587.356111	nan	nan	
Barbados	221.000000	7.000000	200.000000	14.000000	76.904072	nan	nan	
Seychelles	149.000000	0.000000	148.000000	1.000000	151.515152	nan	nan	
Brunei	147.000000	3.000000	143.000000	1.000000	33.601306	nan	nan	
Antigua and Barbuda	119.000000	3.000000	101.000000	15.000000	121.517850	nan	nan	
Saint Vincent and the Grenadines	67.000000	0.000000	64.000000	3.000000	60.389195	nan	nan	
Dominica	33.000000	0.000000	29.000000	4.000000	45.839063	nan	nan	
Saint Lucia	33.000000	0.000000	27.000000	6.000000	17.971018	nan	nan	
Fiji	32.000000	2.000000	30.000000	0.000000	3.569660	nan	nan	
Timor-Leste	29.000000	0.000000	28.000000	1.000000	2.199566	nan	nan	
Holy See	27.000000	0.000000	15.000000	12.000000	3337.453646	nan	nan	
Grenada	27.000000	0.000000	24.000000	3.000000	23.995947	nan	nan	
Laos	23.000000	0.000000	22.000000	1.000000	0.316127	nan	nan	
Saint Kitts	19.000000	0.000000	19.000000	0.000000	35.719657	nan	nan	

country	Confirmed	Deaths	Recovered	Active	Incident_Rate	People_Tested	People_Hospitalized	Mor
and Nevis	10.000000	0.000000	10.000000	0.000000	35.743557	nan	nan	
Western Sahara	10.000000	1.000000	8.000000	1.000000	1.674116	nan	nan	
MS Zaandam	9.000000	2.000000	nan	nan	nan	nan	nan	
Solomon Islands	3.000000	0.000000	nan	nan	0.459518	nan	nan	

Top 10 countries (Confirmed Cases and Deaths)

In [17]:

```python
f = plt.figure(figsize=(10,5))
f.add_subplot(111)

plt.axes(axisbelow=True)
plt.barh(df_countries_cases.sort_values('Confirmed')["Confirmed"].index[-10:],df_countries_cases.so
rt_values('Confirmed')["Confirmed"].values[-10:],color="darkcyan")
plt.tick_params(size=5,labelsize = 13)
plt.xlabel("Confirmed Cases",fontsize=18)
plt.title("Top 10 Countries (Confirmed Cases)",fontsize=20)
plt.grid(alpha=0.3)
```
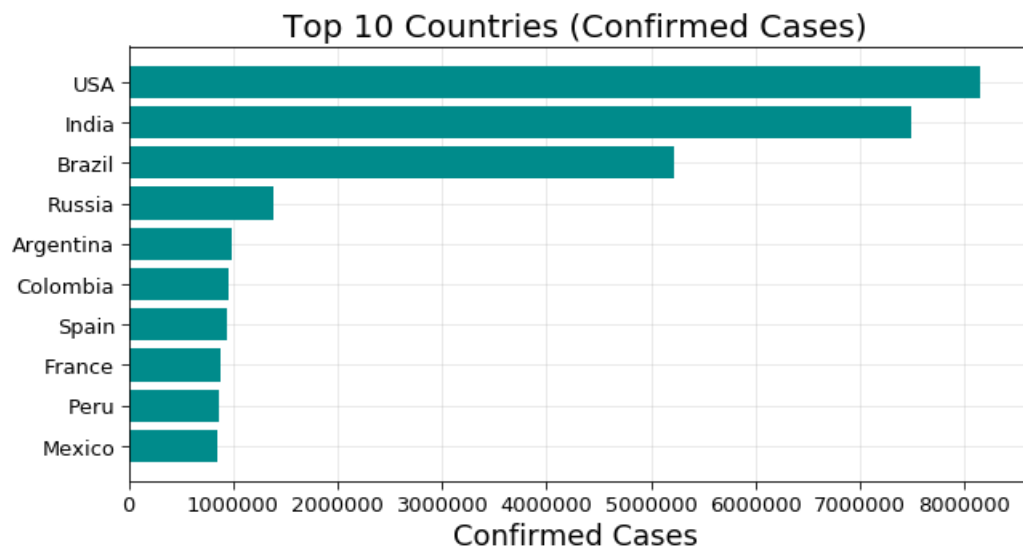

In [18]:

```python
f = plt.figure(figsize=(10,5))
f.add_subplot(111)

plt.axes(axisbelow=True)
plt.barh(df_countries_cases.sort_values('Deaths')["Deaths"].index[-10:],df_countries_cases.sort_val
ues('Deaths')["Deaths"].values[-10:],color="crimson")
plt.tick_params(size=5,labelsize = 13)
plt.xlabel("Deaths Cases",fontsize=18)
plt.title("Top 10 Countries (Deaths Cases)",fontsize=20)
plt.grid(alpha=0.3,which='both')
```

France
Iran

0   50000   100000   150000   200000

Deaths Cases
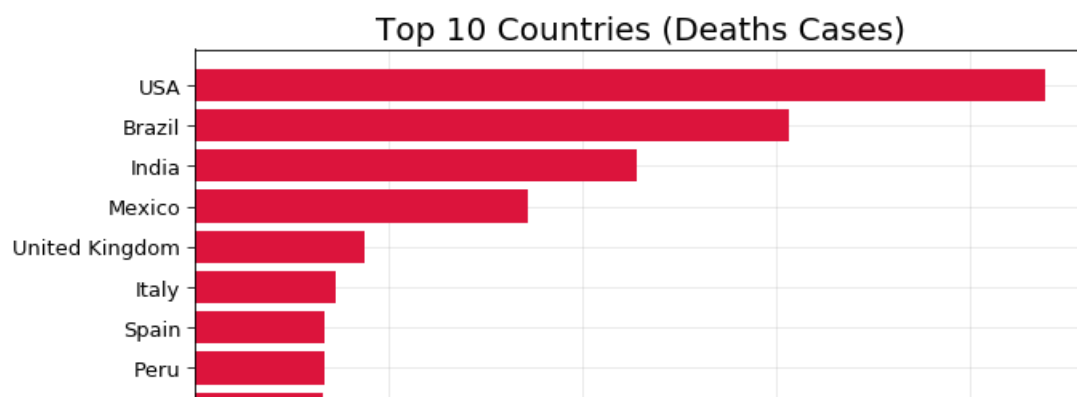
In [19]:

```
f = plt.figure(figsize=(10,5))
f.add_subplot(111)

plt.axes(axisbelow=True)
plt.barh(df_countries_cases.sort_values('Active')["Active"].index[-10:],df_countries_cases.sort_val
ues('Active')["Active"].values[-10:],color="darkorange")
plt.tick_params(size=5,labelsize = 13)
plt.xlabel("Active Cases",fontsize=18)
plt.title("Top 10 Countries (Active Cases)",fontsize=20)
plt.grid(alpha=0.3,which='both')
```


Top 10 Countries (Active Cases)
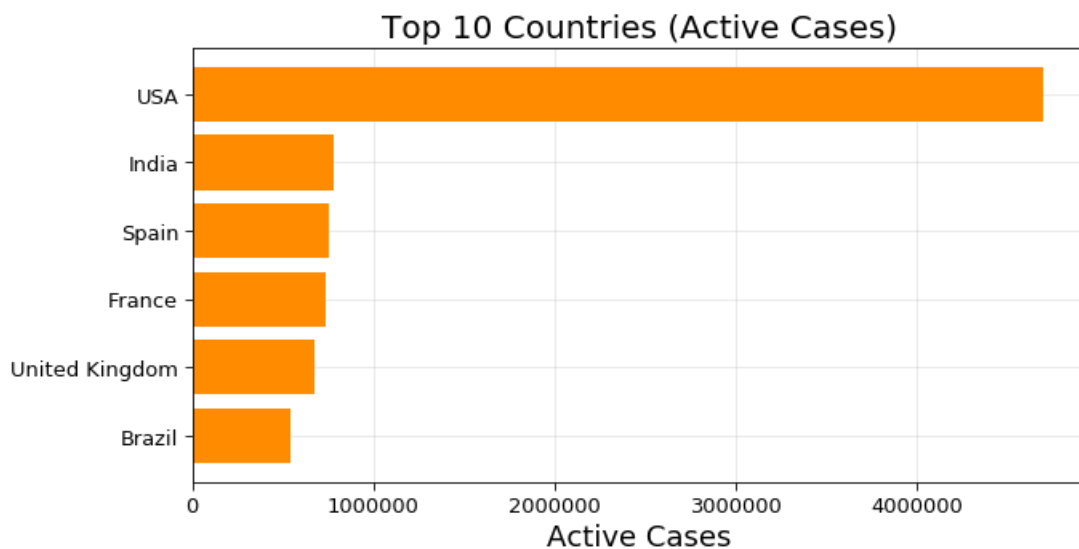
In [20]:

```
f = plt.figure(figsize=(10,5))
f.add_subplot(111)

plt.axes(axisbelow=True)
plt.barh(df_countries_cases.sort_values('Recovered')["Recovered"].index[-10:],df_countries_cases.so
rt_values('Recovered')["Recovered"].values[-10:],color="limegreen")
plt.tick_params(size=5,labelsize = 13)
plt.xlabel("Recovered Cases",fontsize=18)
plt.title("Top 10 Countries (Recovered Cases)",fontsize=20)
plt.grid(alpha=0.3,which='both')
```
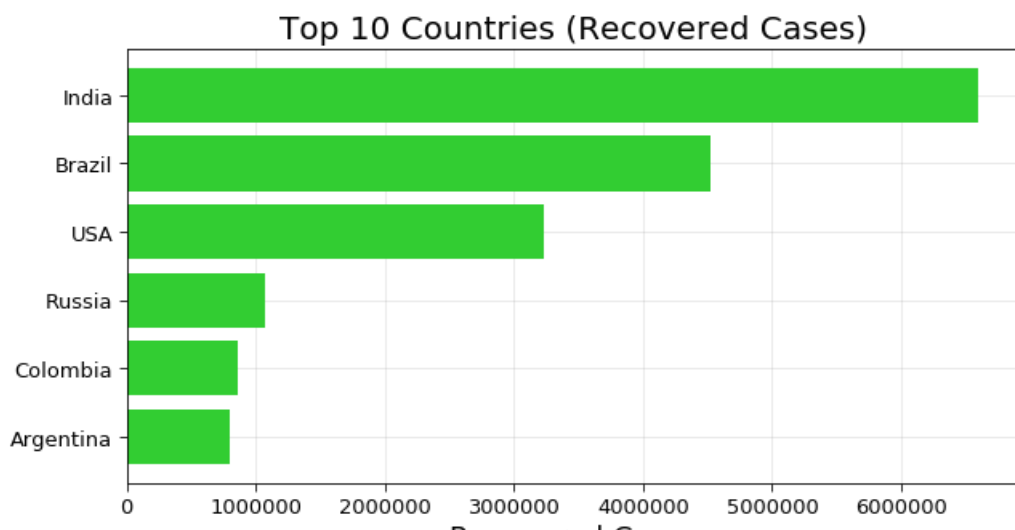

Top 10 Countries (Recovered Cases)

Correlation Analysis

Plotting Heat map of correlation of confirmed cases, recovered cases, deaths and active cases.

Country wise Correlation

In [21]:

```
df_countries_cases.corr().style.background_gradient(cmap='Reds')
```

Out[21]:

	Confirmed	Deaths	Recovered	Active	Incident_Rate	People_Tested	People_Hospitalized	Mortality_Rate	
Confirmed	1.000000	0.933599	0.935600	0.786954	0.230247	nan	nan	0.025499	0.0
Deaths	0.933599	1.000000	0.821802	0.818133	0.256028	nan	nan	0.116305	0.0
Recovered	0.935600	0.821802	1.000000	0.518555	0.196777	nan	nan	0.025971	0.0
Active	0.786954	0.818133	0.518555	1.000000	0.210659	nan	nan	0.040722	0.0
Incident_Rate	0.230247	0.256028	0.196777	0.210659	1.000000	nan	nan	-0.065401	0.0
People_Tested	nan	nan	nan	nan	nan	nan	nan	nan	
People_Hospitalized	nan	nan	nan	nan	nan	nan	nan	nan	
Mortality_Rate	0.025499	0.116305	0.025971	0.040722	-0.065401	nan	nan	1.000000	0.3
UID	-0.013810	-0.014221	-0.022375	0.030262	-0.094630	nan	nan	0.323996	1.0

Continent Wise Correlation

In [22]:

```
df_continents_cases.corr().style.background_gradient(cmap='Reds')
```

Out[22]:

	Confirmed	Deaths	Recovered	Active	Incident_Rate	People_Tested	People_Hospitalized	Mortality_Rate	
Confirmed	1.000000	0.910790	0.917187	0.605240	0.687956	nan	nan	0.273001	0.1
Deaths	0.910790	1.000000	0.712899	0.790698	0.622446	nan	nan	0.171968	0.2
Recovered	0.917187	0.712899	1.000000	0.238127	0.554049	nan	nan	0.214225	0.0
Active	0.605240	0.790698	0.238127	1.000000	0.560402	nan	nan	0.235580	0.0
Incident_Rate	0.687956	0.622446	0.554049	0.560402	1.000000	nan	nan	0.642148	0.3
People_Tested	nan	nan	nan	nan	nan	nan	nan	nan	
People_Hospitalized	nan	nan	nan	nan	nan	nan	nan	nan	
Mortality_Rate	0.273001	0.171968	0.214225	0.235580	0.642148	nan	nan	1.000000	0.7
UID	-0.118890	-0.252957	-0.096877	-0.093093	0.304452	nan	nan	0.781978	1.0

Visualization on Map

In [23]:

```
world_map = folium.Map(location=[10,0], tiles="cartodbpositron", zoom_start=2,max_zoom=6,min_zoom=2
)
for i in range(0,len(df_confirmed)):
    folium.Circle(
        location=[df_confirmed.iloc[i]['Lat'], df_confirmed.iloc[i]['Long']],
        tooltip = "<h5 style='text-align:center;font-weight: bold'>"+df_confirmed.iloc[i]['country'
]+"</h5>"+
                    "<div style='text-align:center;'>"+str(np.nan_to_num(df_confirmed.iloc[i]
['state']))+"</div>"+
                    "<hr style='margin:10px;'>"+
                    "<ul style='color: #444;list-style-type:circle;align-item:left;padding-
left:20px;padding-right:20px'>"+
        "<li>Confirmed: "+str(df_confirmed.iloc[i,-1])+"</li>"+
        "<li>Deaths:    "+str(df_deaths.iloc[i,-1])+"</li>"+
        "<li>Mortality Rate:    "+str(np.round(df_deaths.iloc[i,-1]/(df_confirmed.iloc[i,-1]+1.00001
)*100,2))+"</li>"+
        "</ul>"
        ,
        radius=(int((np.log(df_confirmed.iloc[i,-1]+1.00001)))+0.2)*50000,
        color='#ff6600',
        fill_color='#ff8533',
        fill=True).add_to(world_map)

world_map
```

Out[23]:

Make this Notebook Trusted to load map: File -> Trust Notebook

Global Confirmed Cases Heat Map

In [24]:

```
temp_df = pd.DataFrame(df_countries_cases['Confirmed'])
temp_df = temp_df.reset_index()
fig = px.choropleth(temp_df, locations="country",
                    color=np.log10(temp_df.iloc[:,-1]), # lifeExp is a column of gapminder
                    hover_name="country", # column to add to hover information
                    hover_data=["Confirmed"],
                    color_continuous_scale=px.colors.sequential.Plasma,locationmode="country names"
)
fig.update_geos(fitbounds="locations", visible=False)
fig.update_layout(title_text="Confirmed Cases Heat Map (Log Scale)")
fig.update_coloraxes(colorbar_title="Confirmed Cases(Log Scale)",colorscale="Reds")
# fig.to_image("Global Heat Map confirmed.png")
fig.show()
```

COVID-19 Spread Analysis

Spread Analysis is in two sections

Spread Across Globe

Spread Trends in the World, Continents and few most affected Countries

1. Spread Across Globe

Number of countries affected over the time

```python
case_nums_country = df_confirmed.groupby("country").sum().drop(['Lat','Long'],axis
=1).apply(lambda x: x[x > 0].count(), axis =0)
d = [datetime.strptime(date,'%m/%d/%y').strftime("%d %b") for date in case_nums_country.index]

f = plt.figure(figsize=(15,8))
f.add_subplot(111)
marker_style = dict(c="crimson",linewidth=6, linestyle='-', marker='o',markersize=8, markerfacecolo
r='#ffffff')
plt.plot(d, case_nums_country,**marker_style)
plt.tick_params(labelsize = 14)
plt.xticks(list(np.arange(0,len(d),int(len(d)/5))),d[-1:int(len(d)/5)]+[d[-1]])

#labels
plt.xlabel("Dates",fontsize=18)
plt.ylabel("Number of Countries/Regions",fontsize=1)
plt.grid(alpha = 0.3)


plt.show()
plt.close()
```
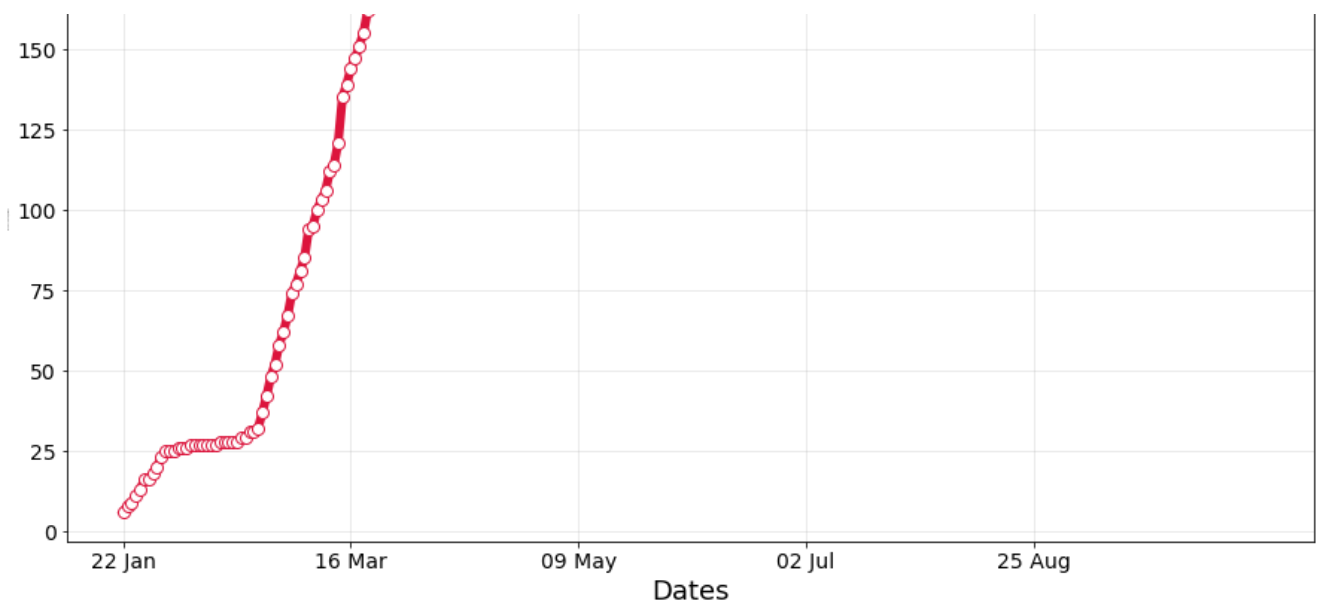
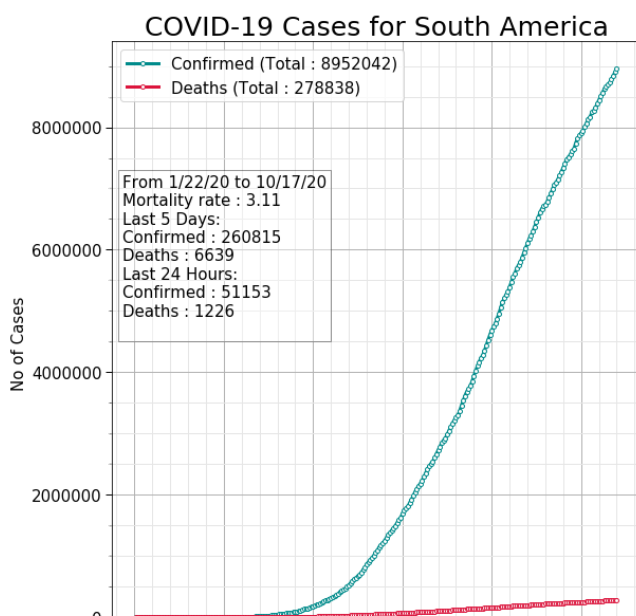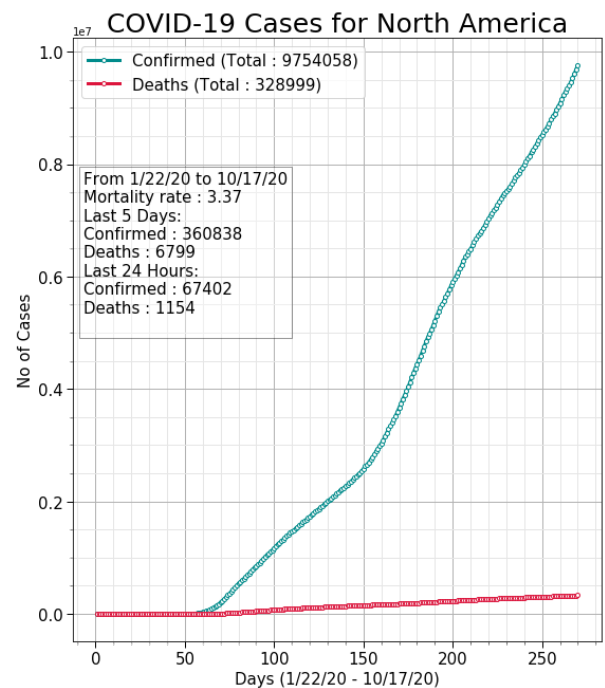## 2. Spread Trends in the World, Continents and few most affected Countries

COVID-19 Global Spread Trends

COVID-19 Spread Trends in Different Continents

COVID-19 Spread Trends in Few Most Affected Countries

COVID-19 Spread Comparison of few most affected countries

COVID-19 Spread Comparison of in different continents

1. COVID-19 Global Spread Trends

In [26]:

```
cols = 1
rows = 1
f = plt.figure(figsize=(10,10*rows))
visualize_covid_cases(df_confirmed, df_deaths,continent = "All",figure = [f,rows,cols, 1])

#plt.savefig(out+'COIVD-19-World.png')
plt.show()
```


COVID-19 Cases World

Days (1/22/20 - 10/17/20)
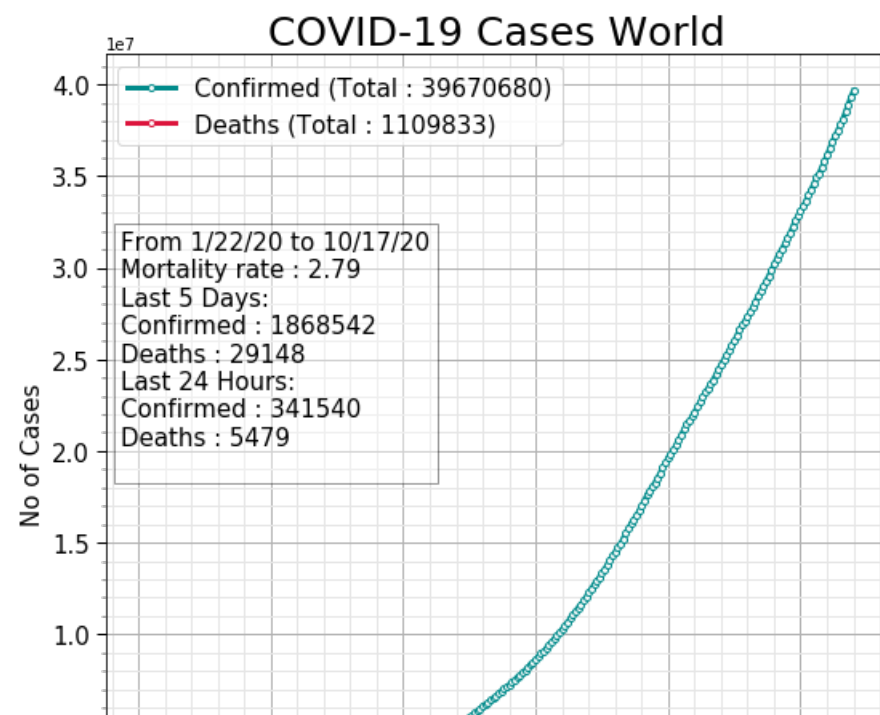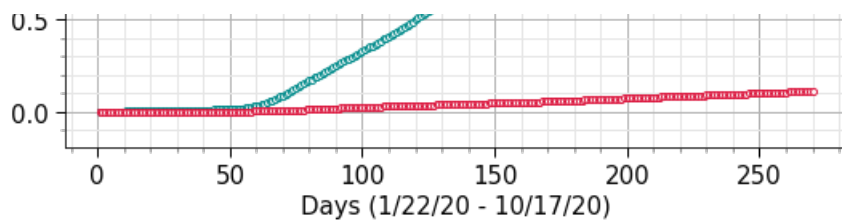
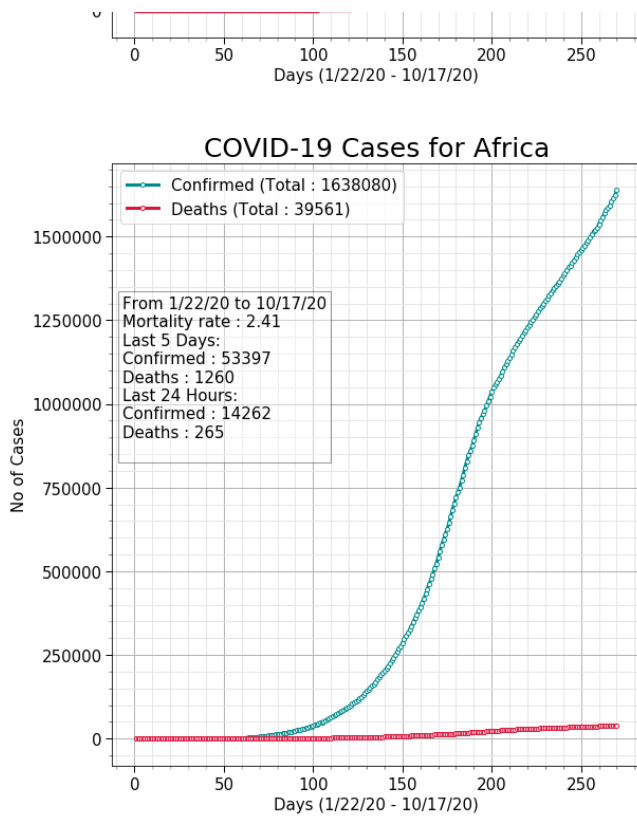## 2 . COVID-19 Spread Trends in Different Continents

In [27]:

```
df_continents= df_confirmed.groupby(["continent"]).sum()
continents = df_continents.sort_values(df_continents.columns[-1],ascending = False).index

cols =2
rows = int(np.ceil(continents.shape[0]/cols))
f = plt.figure(figsize=(20,10*rows))
for i,continent in enumerate(continents):
    visualize_covid_cases(df_confirmed, df_deaths, continent = continent,figure = [f,rows,cols, i+1
])

plt.show()
```


COVID-19 Cases for Asia

le7
- Confirmed (Total : 12345911)
- Deaths (Total : 221443)

From 1/22/20 to 10/17/20
Mortality rate : 1.79
Last 5 Days:
Confirmed : 496518
Deaths : 7868
Last 24 Hours:
Confirmed : 95377
Deaths : 1771

COVID-19 Cases for North America

le7
- Confirmed (Total : 9754058)
- Deaths (Total : 328999)

From 1/22/20 to 10/17/20
Mortality rate : 3.37
Last 5 Days:
Confirmed : 360838
Deaths : 6799
Last 24 Hours:
Confirmed : 67402
Deaths : 1154

COVID-19 Cases for South America

- Confirmed (Total : 8952042)
- Deaths (Total : 278838)

From 1/22/20 to 10/17/20
Mortality rate : 3.11
Last 5 Days:
Confirmed : 260815
Deaths : 6639
Last 24 Hours:
Confirmed : 51153
Deaths : 1226

COVID-19 Cases for Europe

- Confirmed (Total : 6851683)
- Deaths (Total : 238146)

From 1/22/20 to 10/17/20
Mortality rate : 3.47
Last 5 Days:
Confirmed : 688873
Deaths : 6384
Last 24 Hours:
Confirmed : 111628
Deaths : 1024

COVID-19 Cases for Africa

From 1/22/20 to 10/17/20
Mortality rate : 2.41
Last 5 Days:
Confirmed : 53397
Deaths : 1260
Last 24 Hours:
Confirmed : 14262
Deaths : 265

Confirmed (Total : 1638080)
Deaths (Total : 39561)

COVID-19 Cases for Others

From 1/22/20 to 10/17/20
Mortality rate : 1.92
Last 5 Days:
Confirmed : 7990
Deaths : 193
Last 24 Hours:
Confirmed : 1700
Deaths : 39

Confirmed (Total : 99014)
Deaths (Total : 1908)

COVID-19 Cases for Australia

From 1/22/20 to 10/17/20
Mortality rate : 3.13
Last 5 Days:
Confirmed : 111
Deaths : 5
Last 24 Hours:
Confirmed : 18
Deaths : 0

Confirmed (Total : 29892)
Deaths (Total : 938)

4. COVID-19 Spread Comparison of few most affected countries and INDIA

In [28]:

```
temp = df_confirmed.groupby('country').sum().drop(["Lat","Long"],axis =1).sort_values(df_confirmed
.columns[-1], ascending= False)

threshold = 50
f = plt.figure(figsize=(10,12))
ax = f.add_subplot(111)
for i,country in enumerate(temp.index):
    if i >= 9:
        if country != "India" and country != "Japan" :
            continue
    x = 30
    t = temp.loc[temp.index== country].values[0]
    t = t[t>threshold][:x]
```

```python
    date = np.arange(0,len(t[:x]))
    xnew = np.linspace(date.min(), date.max(), 30)
    spl = make_interp_spline(date, t, k=1)  # type: BSpline
    power_smooth = spl(xnew)
    if country != "India":
        plt.plot(xnew,power_smooth,'-o',label = country,linewidth =3, markevery=[-1])
    else:
        marker_style = dict(linewidth=4, linestyle='-', marker='o',markersize=10, markerfacecolor='
#ffffff')
        plt.plot(date,t,"-.",label = country,**marker_style)

plt.tick_params(labelsize = 14)
plt.xticks(np.arange(0,30,7),[ "Day "+str(i) for i in range(30)][::7])

# Reference lines
x = np.arange(0,18)
y = 2**(x+np.log2(threshold))
plt.plot(x,y,"--",linewidth =2,color = "gray")
plt.annotate("No. of cases doubles every day",(x[-2],y[-1]),xycoords="data",fontsize=14,alpha = 0.5
)

x = np.arange(0,26)
y = 2**(x/2+np.log2(threshold))
plt.plot(x,y,"--",linewidth =2,color = "gray")
plt.annotate(".. every socend day",(x[-3],y[-1]),xycoords="data",fontsize=14,alpha = 0.5)

x = np.arange(0,26)
y = 2**(x/7+np.log2(threshold))
plt.plot(x,y,"--",linewidth =2,color = "gray")
plt.annotate(".. every week",(x[-3],y[-1]),xycoords="data",fontsize=14,alpha = 0.5)

x = np.arange(0,26)
y = 2**(x/30+np.log2(threshold))
plt.plot(x,y,"--",linewidth =2,color = "gray")
plt.annotate(".. every month",(x[-3],y[-1]),xycoords="data",fontsize=14,alpha = 0.5)


# India is following trend similar to doulbe the cases in 4 days but it may increase the rate
x = np.arange(0,26)
y = 2**(x/4+np.log2(threshold))
plt.plot(x,y,"--",linewidth =2,color = "Red")
plt.annotate(".. every 4 days",(x[-3],y[-1]),color="Red",xycoords="data",fontsize=14,alpha = 0.8)

# plot Params
plt.xlabel("Days",fontsize=17)
plt.ylabel("Number of Confirmed Cases",fontsize=17)
plt.title("Trend Comparison of Different Countries\n and India (confirmed) ",fontsize=22)
plt.legend(loc = "upper left")
plt.yscale("log")
plt.grid(which="both")
#plt.savefig(out+'Trend Comparison with India (confirmed).png')
plt.show()
```

5. COVID-19 Spread Comparison of in different continents

```python
temp = df_confirmed.groupby('continent').sum().drop(["Lat","Long"],axis =1).sort_values(df_confirme
d.columns[-1], ascending= False)

threshold = 50
f = plt.figure(figsize=(10,12))
ax = f.add_subplot(111)
for i,country in enumerate(temp.index):
    if i > 10:
        break
    x = 30
    t = temp.loc[temp.index== country].values[0]
    t = t[t>threshold][:x]

    date = np.arange(0,len(t[:x]))
    xnew = np.linspace(date.min(), date.max(), 30)
    spl = make_interp_spline(date, t, k=1)   # type: BSpline
    power_smooth = spl(xnew)
    plt.plot(xnew,power_smooth,'-o',label = country,linewidth =3, markevery=[-1])

plt.tick_params(labelsize = 14)
plt.xticks(np.arange(0,30,7),[ "Day "+str(i) for i in range(30)][::7])

# Reference lines
x = np.arange(0,18)
y = 2**(x+np.log2(threshold))
plt.plot(x,y,"--",linewidth =2,color = "gray")
plt.annotate("No. of cases doubles every day",(x[-2],y[-1]),xycoords="data",fontsize=14,alpha = 0.5
)

x = np.arange(0,26)
y = 2**(x/2+np.log2(threshold))
plt.plot(x,y,"--",linewidth =2,color = "gray")
plt.annotate(".. every socend day",(x[-3],y[-1]),xycoords="data",fontsize=14,alpha = 0.5)

x = np.arange(0,26)
y = 2**(x/7+np.log2(threshold))
plt.plot(x,y,"--",linewidth =2,color = "gray")
plt.annotate(".. every week",(x[-3],y[-1]),xycoords="data",fontsize=14,alpha = 0.5)

x = np.arange(0,26)
y = 2**(x/30+np.log2(threshold))
plt.plot(x,y,"--",linewidth =2,color = "gray")
plt.annotate(".. every month",(x[-3],y[-1]),xycoords="data",fontsize=14,alpha = 0.5)


# India is following trend similar to doulbe the cases in 4 days but it may increase the rate
x = np.arange(0,26)
y = 2**(x/4+np.log2(threshold))
```
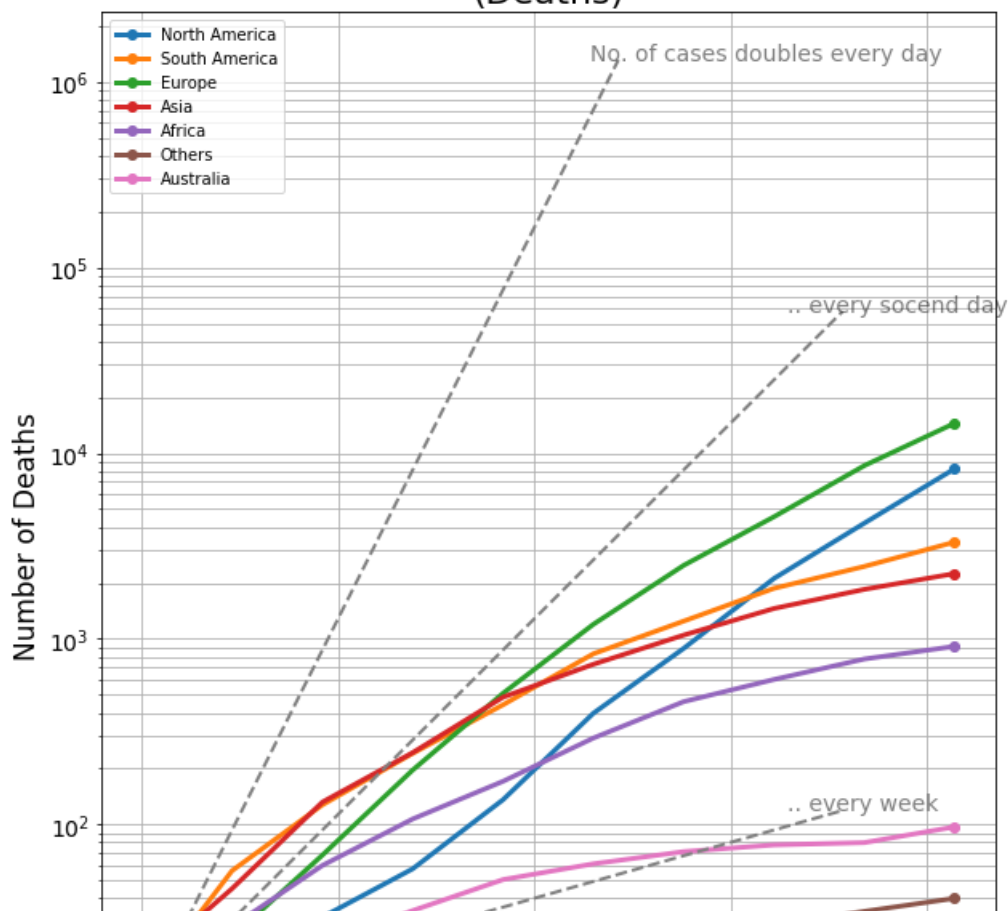
```
plt.plot(x,y,"--",linewidth =2,color = "Red")
plt.annotate(".. every 4 days",(x[-3],y[-1]),xycoords="data",fontsize=14,alpha = 0.8)

# plot Params
plt.xlabel("Days",fontsize=17)
plt.ylabel("Number of Confirmed Cases",fontsize=17)
plt.title("Trend Comparison of Different Continents \n(confirmed) ",fontsize=22)
plt.legend(loc = "upper left")
plt.yscale("log")
plt.grid(which="both")
#plt.savefig(out+'Trend Comparison of continents (Confirmed).png')
plt.show()
```


Trend Comparison of Different Continents (confirmed)
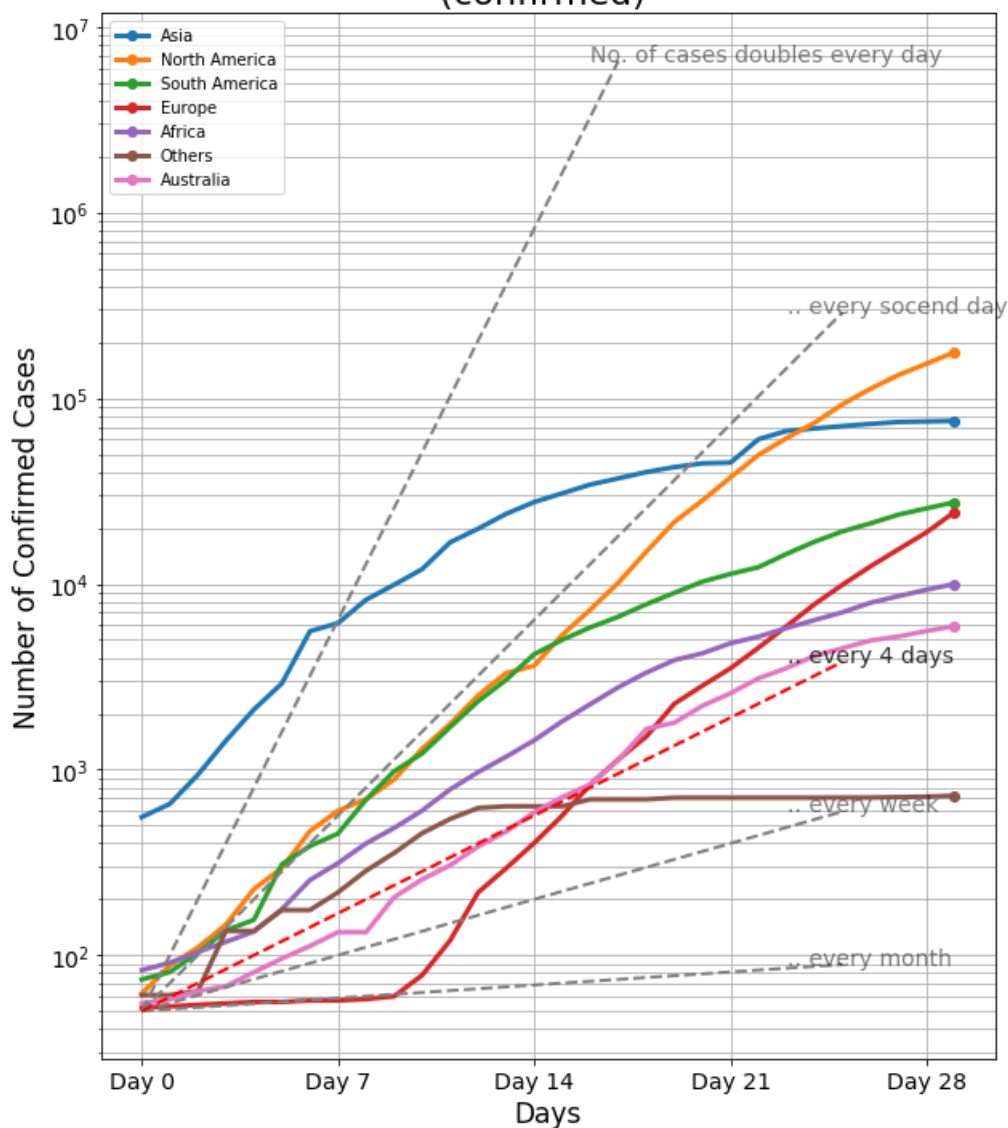
```
temp = df_deaths.groupby('continent').sum().drop(["Lat","Long"],axis
=1).sort_values(df_deaths.columns[-1], ascending= False)

threshold = 10
f = plt.figure(figsize=(10,12))
ax = f.add_subplot(111)
for i,country in enumerate(temp.index):
    if i > 10:
        break
    x = 30
    t = temp.loc[temp.index== country].values[0]
    t = t[t>threshold][:x]

    date = np.arange(0,len(t[:x]))
    xnew = np.linspace(date.min(), date.max(), 10)
```

```
    spl = make_interp_spline(date, t, k=1)   # type: BSpline
    power_smooth = spl(xnew)
    plt.plot(xnew,power_smooth,'-o',label = country,linewidth =3, markevery=[-1])


plt.tick_params(labelsize = 14)
plt.xticks(np.arange(0,30,7),[ "Day "+str(i) for i in range(30)][::7])

# Reference lines
x = np.arange(0,18)
y = 2**(x+np.log2(threshold))
plt.plot(x,y,"--",linewidth =2,color = "gray")
plt.annotate("No. of cases doubles every day",(x[-2],y[-1]),xycoords="data",fontsize=14,alpha = 0.5
)

x = np.arange(0,26)
y = 2**(x/2+np.log2(threshold))
plt.plot(x,y,"--",linewidth =2,color = "gray")
plt.annotate(".. every socend day",(x[-3],y[-1]),xycoords="data",fontsize=14,alpha = 0.5)

x = np.arange(0,26)
y = 2**(x/7+np.log2(threshold))
plt.plot(x,y,"--",linewidth =2,color = "gray")
plt.annotate(".. every week",(x[-3],y[-1]),xycoords="data",fontsize=14,alpha = 0.5)

x = np.arange(0,26)
y = 2**(x/30+np.log2(threshold))
plt.plot(x,y,"--",linewidth =2,color = "gray")
plt.annotate(".. every month",(x[-3],y[-1]),xycoords="data",fontsize=14,alpha = 0.5)

# plot Params
plt.xlabel("Days",fontsize=17)
plt.ylabel("Number of Deaths",fontsize=17)
plt.title("Trend Comparison of Different Continents \n(Deaths)",fontsize=22)
plt.legend(loc = "upper left")
plt.yscale("log")
plt.grid(which="both")
#plt.savefig(out+'Trend Comparison continents (deaths).png')
plt.show()
```
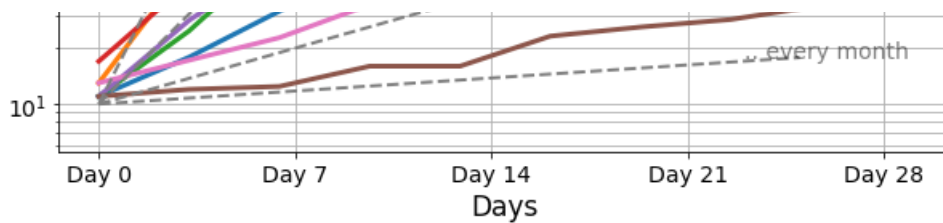
Global Prediction

Global Trend:

It is useful to understand the global trend of an increase in the number of cases over time. There is always a pattern in any data, but the concern is how strongly data follows a pattern. COVID-19 spreads exponentially, positive cases of COVID-19 takes 67 days to reach 1 Lakhs while it takes only 11 days to reach 2 Lakhs, 4 days to reach 3 Lakhs, and just 2 days to reach 5 Lakhs. This trend shows how fast it spreads.

In [31]:

```python
temp_data = df_confirmed.iloc[:,5:].sum(axis =0)
f = plt.figure(figsize=(20,12))
f.add_subplot(111)

threshold = 100000

t = temp_data.values
t = t[t >threshold]

date = np.arange(0,len(t[:]))
xnew = np.linspace(date.min(), date.max(), 10)
spl = make_interp_spline(date, t, k=1)  # type: BSpline
power_smooth = spl(xnew)

marker_style = dict(linewidth=4, linestyle='-', marker='o',markersize=10, markerfacecolor='#ffffff'
)
plt.plot(date,t,"-.",label="Confirmed Cases",**marker_style)


# Reference lines
x = np.arange(0,32)
y = 2**(x+np.log2(threshold))
plt.plot(x,y,"--",linewidth =2,color = "gray")
plt.annotate("No. of Cases Doubles Every Day",
(np.log2((t.max()-threshold)/threshold),t.max()-threshold/2),xycoords="data",fontsize=14,alpha = 0.
5)

x = np.arange(0,32)
y = 2**(x/3+np.log2(threshold))
plt.plot(x,y,"--",linewidth =2,color = "gray")
plt.annotate("...Every Third Day",
(np.log2((t.max()-threshold)/threshold)*3,t.max()-threshold),xycoords="data",fontsize=14,alpha = 0.
5)

x = np.arange(0,32)
y = 2**(x/7+np.log2(threshold))
plt.plot(x,y,"--",linewidth =2,color = "gray")
plt.annotate("... Every Week",(np.log2((t.max()-threshold)/threshold)*7,t.max()-threshold),xycoords
="data",fontsize=14,alpha = 0.5)

x = np.arange(0,32)
y = 2**(x/30+np.log2(threshold))
plt.plot(x,y,"--",linewidth =2,color = "gray")
plt.annotate(".. Every Month",(18,2**(17/30+np.log2(threshold))),xycoords="data",fontsize=14,alpha
= 0.5)

x = np.arange(0,32)
y = 2**(x/9+np.log2(threshold))
plt.plot(x,y,"--",linewidth =2,color = "Red")
plt.annotate(".. Every 9 Days",(np.log2((t.max()-threshold)/threshold)*9,t.max()-threshold),color="
Red",xycoords="data",fontsize=14,alpha = 0.8)
```

```
plt.xlim(date[0],date[-1])
plt.ylim(threshold - threshold/10,t.max()+threshold)
# plot Params
# plt.tight_layout()
plt.tick_params(labelsize = 16)
plt.xticks(np.arange(0,len(t[:])+7,7),[ "Day "+str(i) for i in range(len(t[:])+7)][::7])
plt.xlabel("Days",fontsize=19)
plt.ylabel("Number of Confirmed Cases",fontsize=19)
plt.title("World Trend",fontsize=24)
plt.legend(loc = "upper left")
plt.yscale("log")
plt.grid(which="both")
#plt.savefig(out+"World Trend Confirmed cases.png")
plt.show()
```
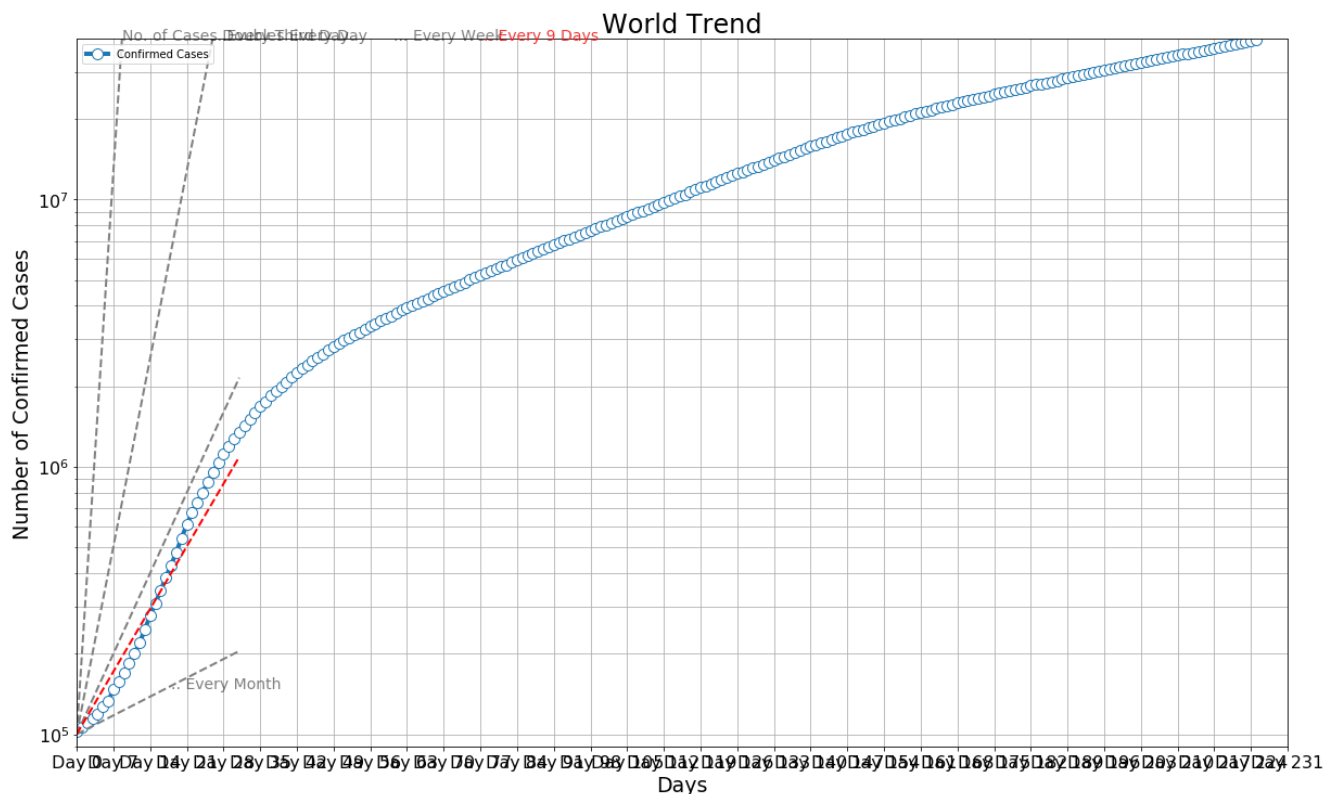

COVID-19 Mortality Rate Variation Over Period of Time

For any epidemic the one of the most important evaluation is Mortality Rate. It is the measure of number of deaths in a particular population during a specific interval.

1st curve shows how the mortality rate varies from 22 JAN 2020 to till date all over the world.

2nd Curve shows the variation of mortality rate in different continents over time.

In [41]:

```
df_continents= df_confirmed.groupby(["continent"]).sum()
continents = df_continents.sort_values(df_continents.columns[-1],ascending = False).index
continents = ["All"]+list(continents)

cols =1
rows = 2
axis_label = ["Days ("+df_confirmed.columns[5]+" - "+df_confirmed.columns[-1]+")","Mortality Rate (
of 100)"]

f = plt.figure(figsize=(15,10*rows))

#SubPlot 1
ax = f.add_subplot(211)
mortality_rate = get_mortality_rate(df_confirmed,df_deaths,continent=continents[0])
plt.plot(np.arange(1,mortality_rate.shape[0]+1),mortality_rate,label = "World : Current Mortality R
ate "+str(mortality_rate[-1]))
```

```
plt_title = "COVID-19: World Mortality Rate Curve"
plot_params(ax,axis_label,plt_title)
# Legend Location
l = plt.legend(loc= "best")


#SubPlot 2
ax = f.add_subplot(212)
for i, continent in enumerate(continents[1:]):
    mortality_rate = get_mortality_rate(df_confirmed,df_deaths,continent=continent)
    plt.plot(np.arange(1+mortality_rate[mortality_rate == 0].shape[0],mortality_rate[mortality_rate
== 0].shape[0]+mortality_rate[mortality_rate > 0].shape[0]+1),mortality_rate[mortality_rate > 0],la
bel = continents[i+1]+" "+str(mortality_rate[-1]))

plt_title = "COVID-19: Mortality Rate Curve for all Continents"
plot_params(ax,axis_label,plt_title)

# Legend Location
l = plt.legend(loc= "best")

plt.minorticks_on()
#plt.savefig(out+'Mortality rate.png')
plt.show()
```

C:\Users\hp\anaconda3\lib\site-packages\ipykernel_launcher.py:112: RuntimeWarning:

invalid value encountered in true_divide

C:\Users\hp\anaconda3\lib\site-packages\ipykernel_launcher.py:112: RuntimeWarning:

invalid value encountered in true_divide

C:\Users\hp\anaconda3\lib\site-packages\ipykernel_launcher.py:112: RuntimeWarning:
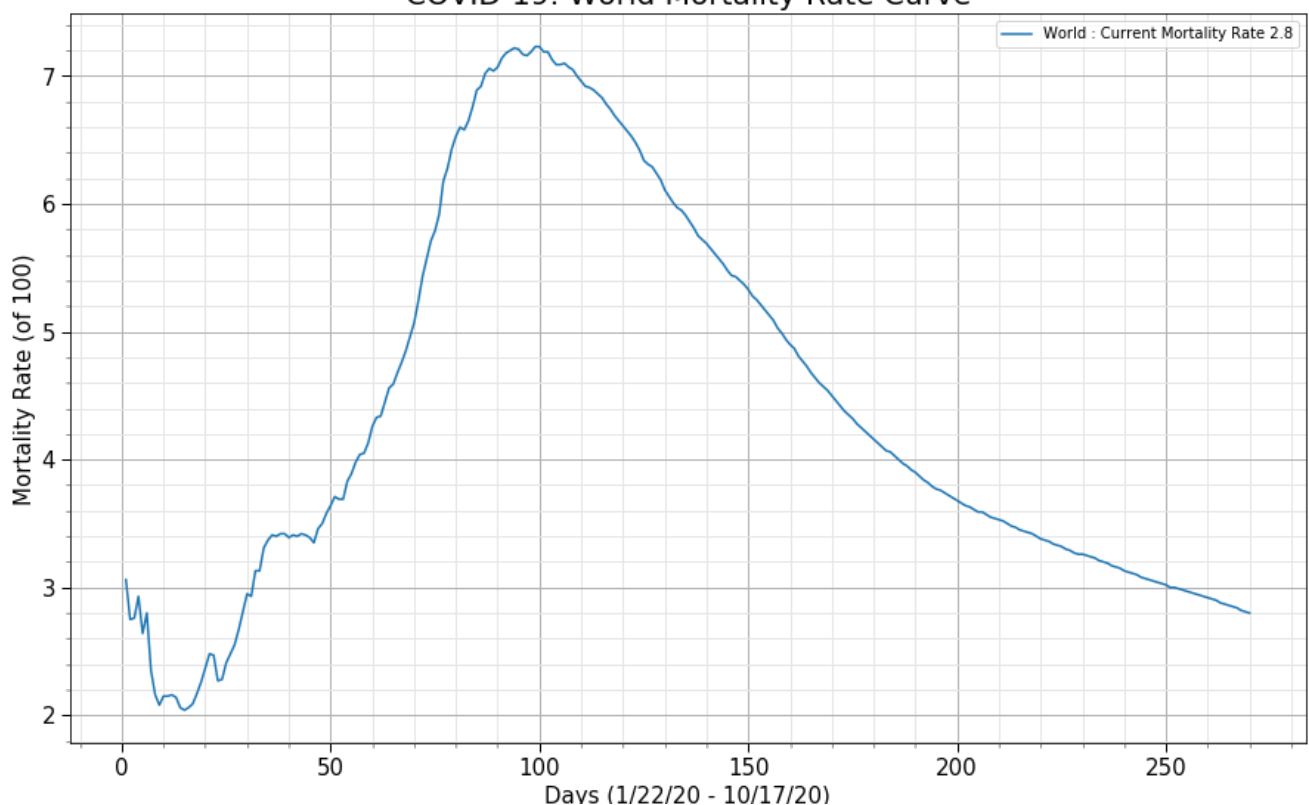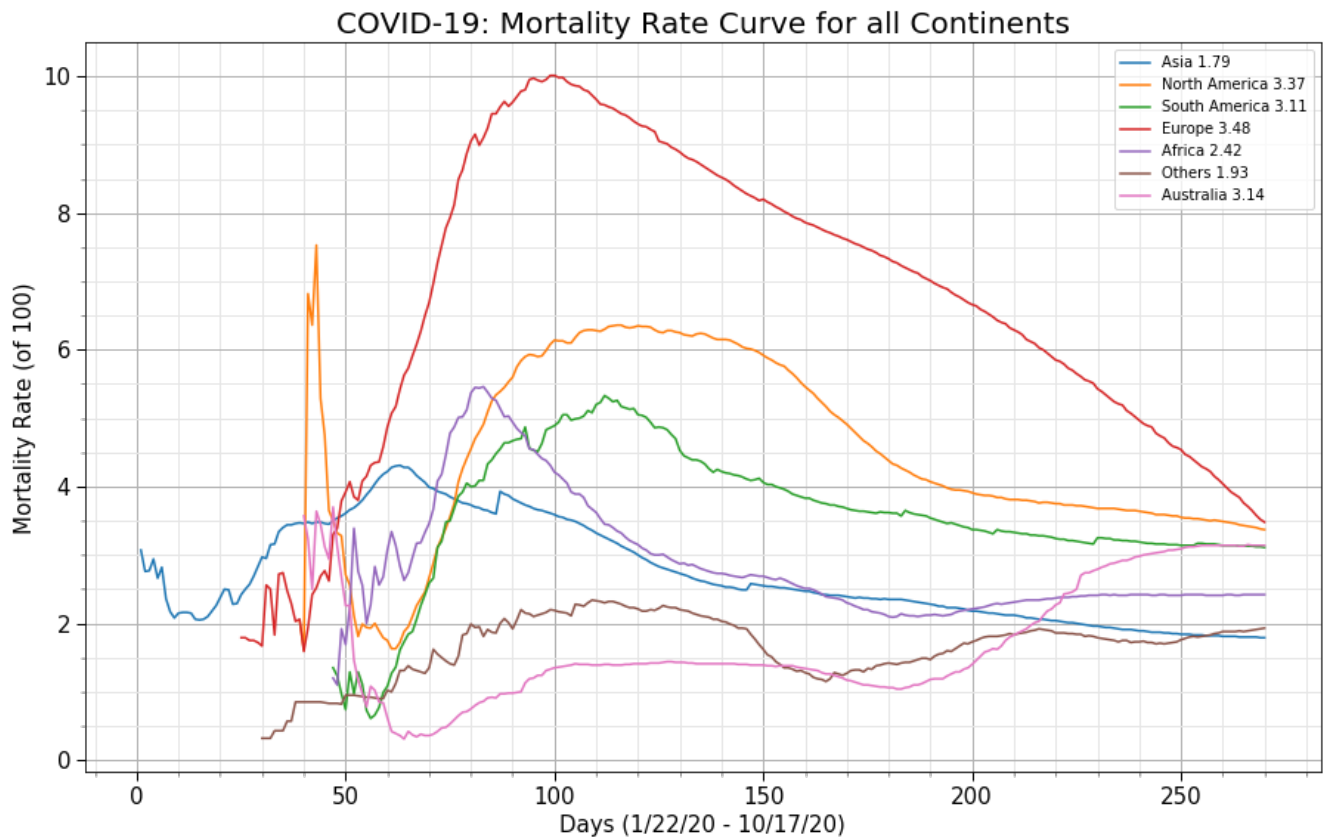
invalid value encountered in true_divide

C:\Users\hp\anaconda3\lib\site-packages\ipykernel_launcher.py:112: RuntimeWarning:

invalid value encountered in true_divide

C:\Users\hp\anaconda3\lib\site-packages\ipykernel_launcher.py:112: RuntimeWarning:

invalid value encountered in true_divide

COVID-19: Mortality Rate Curve for all Continents

COVID-19: Mortality Rate Curve for all Continents

Legend:
- Asia 1.79
- North America 3.37
- South America 3.11
- Europe 3.48
- Africa 2.42
- Others 1.93
- Australia 3.14

Y-axis: Mortality Rate (of 100)
X-axis: Days (1/22/20 - 10/17/20)

COVID-19: Change in Mortality Rate of Each Countries Over Time

```python
df_data = df_table.groupby(['Last_Update', 'Country_Region'])['Confirmed', 'Deaths','continent'].max().reset_index()
df_data["Last_Update"] = pd.to_datetime( df_data["Last_Update"]).dt.strftime('%m/%d/%Y')

fig = px.scatter(df_data, y=100*df_data["Deaths"]/(df_data["Confirmed"]+1),
                 x= df_data["Confirmed"]+1,
                 range_y = [-1,18],
                 range_x = [1,df_data["Confirmed"].max()+10000],
                 color= "continent", hover_name="Country_Region",
                 hover_data=["Confirmed","Deaths"],
                 range_color= [0, max(np.power(df_data["Confirmed"],0.3))],
                 animation_frame="Last_Update",
                 animation_group="Country_Region",
                 color_continuous_scale=px.colors.sequential.Plasma,
                 title='COVID-19: Change in Mortality Rate of Each Countries Over Time',
                 size = np.power(df_data["Confirmed"]+1,0.3)-0.5,
                 size_max = 30,
                 log_x=True,
                 height =700,
                 )
fig.update_coloraxes(colorscale="hot")
fig.update(layout_coloraxis_showscale=False)
fig.update_xaxes(title_text="Confirmed Cases (Log Scale)")
fig.update_yaxes(title_text="Mortality Rate (%)")
fig.show()
```

```
C:\Users\hp\anaconda3\lib\site-packages\ipykernel_launcher.py:1: FutureWarning:

Indexing with multiple keys (implicitly converted to a tuple of keys) will be deprecated, use a li
st instead.
```

Cumulative Confirmed Cases and Cumulative Recovery Vs Cumulative Deaths Analysis

The variation of Cumulative Confirmed Cases and Cumulative Recovery with Cumulative Deaths can show a trend. These 2 curves depict the same. Also, these curves should be a straight line as shown in the 1st curve, but the 2nd curve is not showing that trend, and as the number of recovered cases is increasing, death is increasing at a faster rate.

1st curve: Cumulative Confirmed Cases VS Cumulative Deaths

2nd curve: Cululative Recovery VS Cumulative Deaths

In [43]:

```python
cols =1
rows = 2

f = plt.figure(figsize=(15,10*rows))

# SubPlot 1
ax = f.add_subplot(211)
plt.plot(np.sum(np.asarray(df_confirmed.iloc[:,5:]),axis = 0),np.sum(np.asarray(df_deaths.iloc[:,5:]),axis = 0))

axis_label = ["Cumulative Confirmed Cases","Cumulative Deaths"]
plt_title = "COVID-19: World - \nCumulative Confirmed Cases Vs Cumulative Deaths Curve"
plot_params(ax,axis_label,plt_title)


# # SubPlot 2
# ax = f.add_subplot(212)
# mortality_rate = get_mortality_rate(df_confirmed,df_deaths,continent=continents[0])
# plt.plot(np.sum(np.asarray(df_recovered.iloc[:,5:]),axis = 0),np.sum(np.asarray(df_deaths.iloc[:,5:]),axis = 0))

# axis_label = ["Cumulative Recoveries","Cumulative Deaths"]
# plt_title = "COVID-19: World - Cumulative Recovery Vs Cumulative Deaths Curve"
```
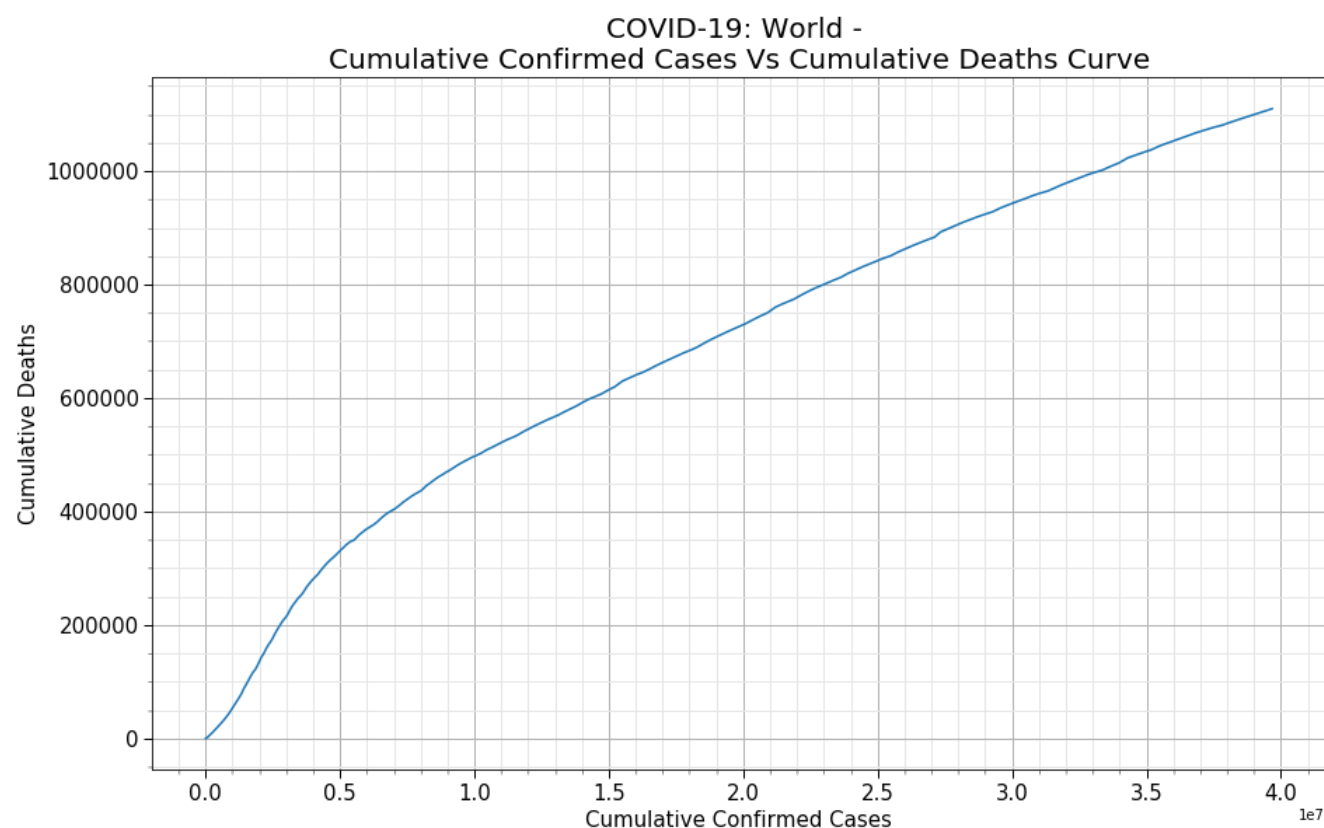
```
# plot_params(ax,axis_label,plt_title)
plt.minorticks_on()
#plt.savefig(out+'Cumulative Confirmed Cases Vs Cumulative Deaths Curve.png')
plt.show()
```


Variation of Deaths vs Confirmed cases of different cuntries over time

In [44]:

```
df_data = df_table.groupby(['Last_Update', 'Country_Region'])['Confirmed', 'Deaths','continent'].ma
x().reset_index()
df_data["Last_Update"] = pd.to_datetime( df_data["Last_Update"]).dt.strftime('%m/%d/%Y')

fig = px.scatter(df_data, y=df_data["Deaths"],
                 x= df_data["Confirmed"]+1,
                 range_y = [1,df_data["Deaths"].max()+1000],
                 range_x = [1,df_data["Confirmed"].max()+10000],
                 color= "continent", hover_name="Country_Region",
                 hover_data=["Confirmed","Deaths"],
                 range_color= [0, max(np.power(df_data["Confirmed"],0.3))],
                 animation_frame="Last_Update",
                 animation_group="Country_Region",
                 color_continuous_scale=px.colors.sequential.Plasma,
                 title='COVID-19: Change Deaths vs Cofirmed of Each Countries Over Time',
                 size = np.power(df_data["Confirmed"]+1,0.3)-0.5,
                 size_max = 30,
                 log_x=True,
                 log_y=True,
                 height =700,
                 )
fig.update_coloraxes(colorscale="hot")
fig.update(layout_coloraxis_showscale=False)
fig.update_xaxes(title_text="Confirmed Cases (Log Scale)")
fig.update_yaxes(title_text="Deaths Rate (Log Scale)")
fig.show()
```

```
C:\Users\hp\anaconda3\lib\site-packages\ipykernel_launcher.py:1: FutureWarning:

Indexing with multiple keys (implicitly converted to a tuple of keys) will be deprecated, use a li
st instead.
```

COVID-19 : INDIA

Few basic visualization related to India. I will be updating with more visualization in further commits. i am also working on Paitent Data Insights. Adding Soon

Dataset: This dataset is provided by https://api.rootnet.in/

Analysis of Tests done in India

In [45]:

```python
#For runing this command connect to your internet
df_india_test = pd.io.json.json_normalize(requests.get('https://api.rootnet.in/covid19-
in/stats/testing/history').json()['data']).rename(columns =
{"totalIndividualsTested":"c_individualtest","totalPositiveCases":"c_positive","totalSamplesTested"
:"c_tests"})
```

```
C:\Users\hp\anaconda3\lib\site-packages\ipykernel_launcher.py:2: FutureWarning:

pandas.io.json.json_normalize is deprecated, use pandas.json_normalize instead
```

In [46]:

```python
df_india_test["p2t_ratio"]= np.round(100*df_india_test["c_positive"]/df_india_test["c_tests"],2)
df_india_test["positive"] = df_india_test["c_positive"].diff()
df_india_test["tests"] = df_india_test["c_tests"].diff()
df_india_test["p2t_ratio"]= np.round(100*df_india_test["positive"]/df_india_test["tests"],2)
```

Total tests done till date (30 March 2020) in India

In [47]:

```
df_india_test["c_tests"][-1:].values[0]
```

Out[47]:

94224190.0

Test Conducted per Million People

In [48]:

```
np.round(1000000*df_india_test["c_tests"][-1:].values[0]/1300000000,2)
```

Out[48]:

72480.15

COVID19 Cases in India

In [49]:

```
india_data_json = requests.get('https://api.rootnet.in/covid19-
in/unofficial/covid19india.org/statewise').json()
df_india = pd.io.json.json_normalize(india_data_json['data']['statewise'])
df_india = df_india.set_index("state")
```

```
C:\Users\hp\anaconda3\lib\site-packages\ipykernel_launcher.py:2: FutureWarning:

pandas.io.json.json_normalize is deprecated, use pandas.json_normalize instead
```

In [50]:

```
total = df_india.sum()
total.name = "Total"
pd.DataFrame(total).transpose().style.background_gradient(cmap='Wistia',axis=1)
```

Out[50]:

	confirmed	recovered	deaths	active
Total	7549247	6660817	114646	772558

In [51]:

```
df_india.style.background_gradient(cmap='Wistia')
```

Out[51]:

state	confirmed	recovered	deaths	active
Maharashtra	1595381	1369810	42115	182973
Andhra Pradesh	783132	740229	6429	36474
Karnataka	765586	645825	10478	109264
Tamil Nadu	687400	637637	10642	39121
Uttar Pradesh	455146	415592	6658	32896
Delhi	331017	301716	6009	23292

state	confirmed	recovered	deaths	active
Kerala	341860	245399	1162	95203
West Bengal	321036	281053	6056	33927
Odisha	268364	246837	1188	20339
Telangana	222111	198790	1271	22050
Bihar	204212	192594	996	10621
Assam	200709	171680	868	28158
Rajasthan	173266	150379	1747	21140
Gujarat	159726	141752	3638	14336
Madhya Pradesh	160188	144134	2773	13281
Chhattisgarh	160396	132168	1478	26750
Haryana	150033	138351	1640	10042
Punjab	127630	117883	4012	5735
Jharkhand	96352	89011	839	6502
Jammu and Kashmir	87942	77886	1379	8677
Uttarakhand	58024	50982	927	5728
Goa	40587	36395	544	3648
Puducherry	33141	28290	574	4277
Tripura	29465	26199	326	2917
Himachal Pradesh	18967	16037	263	2630
Manipur	15463	11741	116	3606
Chandigarh	13646	12554	208	884
Arunachal Pradesh	13406	10552	30	2824
Meghalaya	8508	6282	75	2151
Nagaland	7816	6142	21	1582
Ladakh	5598	4615	66	917
Andaman and Nicobar Islands	4108	3868	56	184
Sikkim	3597	3184	60	272
Dadra and Nagar Haveli and Daman and Diu	3181	3102	2	52
Mizoram	2253	2148	0	105
State Unassigned	0	0	0	0
Lakshadweep	0	0	0	0

States with Reported Deaths

In [52]:

```python
df_india[df_india['deaths'] > 0].style.background_gradient(cmap='Wistia')
```

Out[52]:

state	confirmed	recovered	deaths	active
Maharashtra	1595381	1369810	42115	182973
Andhra Pradesh	783132	740229	6429	36474
Karnataka	765586	645825	10478	109264
Tamil Nadu	687400	637637	10642	39121
Uttar Pradesh	455146	415592	6658	32896
Delhi	331017	301716	6009	23292
Kerala	341860	245399	1162	95203
West Bengal	321036	281053	6056	33927
Odisha	268364	246837	1188	20339

state	confirmed	recovered	deaths	active
Telangana	222111	198790	1271	22050
Bihar	204212	192594	996	10621
Assam	200709	171680	868	28158
Rajasthan	173266	150379	1747	21140
Gujarat	159726	141752	3638	14336
Madhya Pradesh	160188	144134	2773	13281
Chhattisgarh	160396	132168	1478	26750
Haryana	150033	138351	1640	10042
Punjab	127630	117883	4012	5735
Jharkhand	96352	89011	839	6502
Jammu and Kashmir	87942	77886	1379	8677
Uttarakhand	58024	50982	927	5728
Goa	40587	36395	544	3648
Puducherry	33141	28290	574	4277
Tripura	29465	26199	326	2917
Himachal Pradesh	18967	16037	263	2630
Manipur	15463	11741	116	3606
Chandigarh	13646	12554	208	884
Arunachal Pradesh	13406	10552	30	2824
Meghalaya	8508	6282	75	2151
Nagaland	7816	6142	21	1582
Ladakh	5598	4615	66	917
Andaman and Nicobar Islands	4108	3868	56	184
Sikkim	3597	3184	60	272
Dadra and Nagar Haveli and Daman and Diu	3181	3102	2	52

In [53]:

```
f = plt.figure(figsize=(10,5))
f.add_subplot(111)

plt.axes(axisbelow=True)
plt.barh(df_india.sort_values('confirmed')["confirmed"].index[-10:],df_india.sort_values('confirmed')["confirmed"].values[-10:],color="darkcyan")
plt.tick_params(size=5,labelsize = 13)
plt.xlabel("Confirmed Cases",fontsize=18)
plt.title("Top 10 States: India (Confirmed Cases)",fontsize=20)
plt.grid(alpha=0.3)
#plt.savefig(out+'Top 10 States_India (Confirmed Cases).png')
```
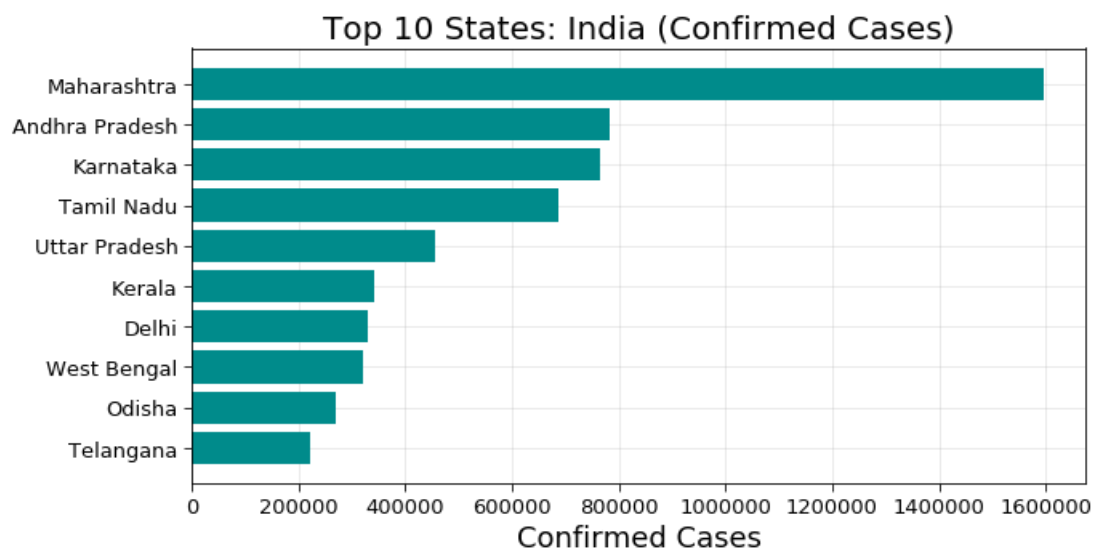

Top 10 States: India (Confirmed Cases)

```
f = plt.figure(figsize=(10,5))
f.add_subplot(111)

plt.axes(axisbelow=True)
plt.barh(df_india.sort_values('deaths')["deaths"].index[-10:],df_india.sort_values('deaths')["death
s"].values[-10:],color="crimson")
plt.tick_params(size=5,labelsize = 13)
plt.xlabel("Deaths",fontsize=18)
plt.title("Top 10 States: India (Deaths Cases)",fontsize=20)
plt.grid(alpha=0.3)
#plt.savefig(out+'Top 10 States_India (Deaths Cases).png')
```
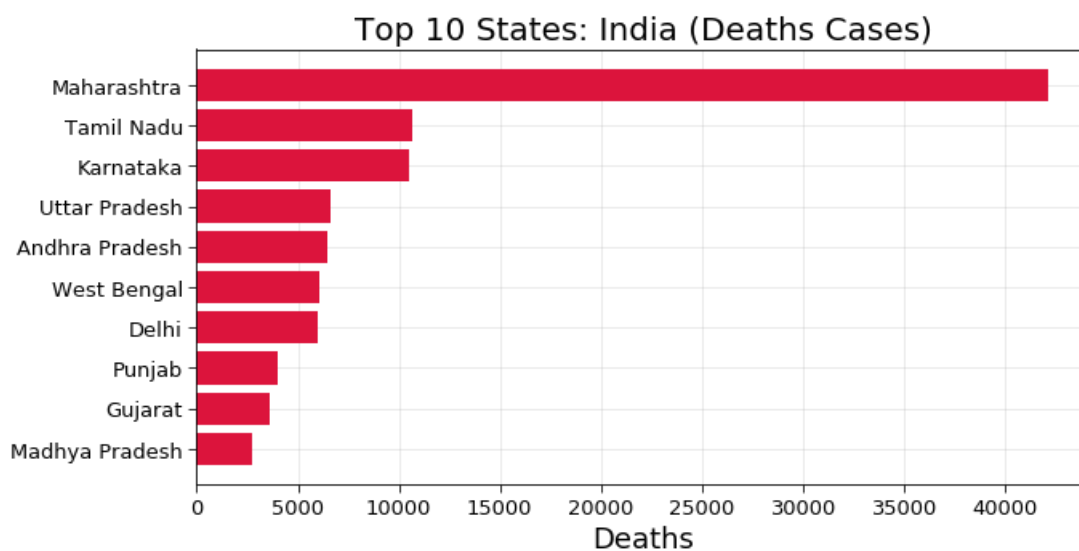

Correlation

In [55]:

```
df_india.corr().style.background_gradient(cmap='Reds')
```

Out[55]:

	confirmed	recovered	deaths	active
confirmed	1.000000	0.998287	0.921107	0.905507
recovered	0.998287	1.000000	0.911909	0.879502
deaths	0.921107	0.911909	1.000000	0.861551
active	0.905507	0.879502	0.861551	1.000000

Conclusion

There are hundreds of coronaviruses, most of which circulate in animals. Only seven of these viruses infect humans and four of them cause symptoms of the common cold. But, three times in the last 20 years, a coronavirus has jumped from animals to humans to cause severe disease.

SARS, a beta coronavirus emerged in 2002 and was controlled mainly by aggressive public health measures. There have been no new cases since 2004. MERS emerged in 2012, still exists in camels, and can infect people who have close contact with them.

COVID-19, a new and sometimes deadly respiratory illness that is believed to have originated in a live animal market in China, has spread rapidly throughout that country and the world.

The new coronavirus was first detected in Wuhan, China in December 2019. Tens of thousands of people were infected in China, with the virus spreading easily from person-to-person in many parts of that country.

The novel coronavirus infections were at first associated with travel from Wuhan, but the virus has now established itself in 177

countries and territories around the world in a rapidly expanding pandemic. Health officials in the United States and around the world are working to contain the spread of the virus through public health measures such as social distancing, contact tracing, testing, quarantines and travel restrictions. Scientists are working to find medications to treat the disease and to develop a vaccine.

The World Health Organization declared the novel coronavirus outbreak "a public health emergency of international concern" on January 30. On March 11, 2020 after sustained spread of the disease outside of China, the World Health Organization declared the COVID-19 epidemic a pandemic. Public health measures like ones implemented in China and now around the world, will hopefully blunt the spread of the virus while treatments and a vaccine are developed to stop it.

In [ ]: