In [1]:
```
import os
import numpy as np
import requests
```

In [3]:
```
url = 'https://www.imdb.com/chart/top/'
```

Out[3]:
```
<Response [403]>
```

In [4]:
```
url1 = 'https://www.worldometers.info/coronavirus/#countries'
```

Out[4]:
```
<Response [200]>
```

In [5]:
```
url2 = 'https://www.ambitionbox.com/list-of-companies?page=1'
```

Out[5]:
```
<Response [403]>
```

In [6]:
```
url3 = 'https://www.learnbay.co/'
```

Out[6]:
```
<Response [200]>
```

In [7]:
```
headers={'User-Agent':'Mozilla/5.0 (Windows NT 6.3; Win 64 ; X64) Apple WeKit /537.36(KHTML, like Gecko) Chrome/80.0.3987.162 Saf
```

In [8]:
```
html = requests.get('https://www.imdb.com/chart/top/', headers=headers).text
```

In [9]:
```
soup = BeautifulSoup(html)
```

Out[9]:
```
<!DOCTYPE html>
<html lang="en-US" xmlns:fb="http://www.facebook.com/2008/fbml" xmlns:og="http://opengraphprotocol.org/schema/"><head><meta c
harset="utf-8"/><meta content="width=device-width" name="viewport"/><script>if(typeof uet === 'function'){ uet('bb', 'LoadTit
le', {wb: 1}); }</script><script>window.addEventListener('load', (event) => {
        if (typeof window.csa !== 'undefined' && typeof window.csa === 'function') {
            var csaLatencyPlugin = window.csa('Content', {
                element: {
                    slotId: 'LoadTitle',
                    type: 'service-call'
                }
            });
            csaLatencyPlugin('mark', 'clickToBodyBegin', 1708874283973);
        }
    })</script><title>IMDb Top 250 Movies</title><meta content="As rated by regular IMDb voters." data-id="main" name="descri
ption"/><meta content="IMDb" property="og:site_name"/><meta content="IMDb Top 250 Movies" property="og:title"/><meta content
="As rated by regular IMDb voters." property="og:description"/><meta content="website" property="og:type"/><meta content="htt
ps://m.media-amazon.com/images/G/01/imdb/images/social/imdb_logo.png" property="og:image"/><meta content="1000" property="og:
image:height"/><meta content="1000" property="og:image:width"/><meta content="en_US" property="og:locale"/><meta content="es_
ES" property="og:locale:alternate"/><meta content="es_MX" property="og:locale:alternate"/><meta content="fr_FR" property="og:
```

In [10]:
```
soup.find('title').getText()
```

Out[10]:
```
'IMDb Top 250 Movies'
```

In [11]:
```
soup.findAll('h1')
```

Out[11]:
```
[<h1 class="ipc-title__text chart-layout-specific-title-text">IMDb Top 250 Movies</h1>]
```

In [12]:
```
soup.findAll('h2', {'class':'ipc-title__text'})
```

Out[12]:
```
[]
```

In [13]:
```
titleList = soup.findAll('h3', {'class':'ipc-title__text'})
```

```
<h3 class="ipc-title__text">23. The Silence of the Lambs</h3>,
<h3 class="ipc-title__text">24. Saving Private Ryan</h3>,
<h3 class="ipc-title__text">25. City of God</h3>,
<h3 class="ipc-title__text">26. Life Is Beautiful</h3>,
<h3 class="ipc-title__text">27. The Green Mile</h3>,
<h3 class="ipc-title__text">28. Terminator 2: Judgment Day</h3>,
<h3 class="ipc-title__text">29. Star Wars: Episode IV - A New Hope</h3>,
<h3 class="ipc-title__text">30. Back to the Future</h3>,
<h3 class="ipc-title__text">31. Spirited Away</h3>,
<h3 class="ipc-title__text">32. Spider-man: Across the Spider-verse</h3>,
<h3 class="ipc-title__text">33. The Pianist</h3>,
<h3 class="ipc-title__text">34. Parasite</h3>,
<h3 class="ipc-title__text">35. Psycho</h3>,
<h3 class="ipc-title__text">36. Gladiator</h3>,
<h3 class="ipc-title__text">37. The Lion King</h3>,
<h3 class="ipc-title__text">38. Léon</h3>,
<h3 class="ipc-title__text">39. The Departed</h3>,
<h3 class="ipc-title__text">40. American History X</h3>,
<h3 class="ipc-title__text">41. Whiplash</h3>,
<h3 class="ipc-title__text">42. The Prestige</h3>,
```

In [14]:
```
for title in titleList:
```

```
4. The Godfather: Part II
5. 12 Angry Men
6. Schindler's List
7. The Lord of the Rings: The Return of the King
8. Pulp Fiction
9. The Lord of the Rings: The Fellowship of the Ring
10. Il Buono, Il Brutto, Il Cattivo
11. Forrest Gump
12. The Lord of the Rings: The Two Towers
13. Fight Club
14. Inception
15. Star Wars: Episode V - The Empire Strikes Back
16. The Matrix
17. GoodFellas
18. One Flew Over the Cuckoo's Nest
19. Se7en
20. It's a Wonderful Life
21. Interstellar
22. Shichinin No Samurai
23. The Silence of the Lambs
```

https://www.worldometers.info/coronavirus/#countries (https://www.worldometers.info/coronavirus/#countries)

In [15]:
```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
%matplotlib inline
import seaborn as sns
sns.set()
import warnings
warnings.filterwarnings('ignore')
import sklearn, scipy
from datetime import date, datetime
from urllib.request import Request, urlopen
from bs4 import BeautifulSoup as soup
import plotly.graph_objects as go
import plotly.express as px
import plotly.offline as py
```

In [18]:
```
url = 'https://www.worldometers.info/coronavirus/#countries'
```

Out[18]:
```
'\n<!DOCTYPE html>\n<!--[if IE 8]> <html lang="en" class="ie8"> <![endif]-->\n<!--[if IE 9]> <html lang="en" class="ie9"> <![endif]-->\n<!--[if !IE]><!-->\n<html lang="en">\n<!--<![endif]-->\n\n\n\n<head>\n    <meta charset="utf-8">\n    <meta http-equiv="X-UA-Compatible" content="IE=edge">\n    <meta name="viewport" content="width=device-width, initial-scale=1">\n\n    <title>COVID - Coronavirus Statistics - Worldometer</title>\n    <meta name="description" content="Daily and weekly updated statistics tracking the number of COVID-19 cases, recovered, and deaths. Historical data with cumulative charts, graphs, and updates.">\n\n\n    \n\t<!-- Favicon -->\n\t<link rel="shortcut icon" href="/favicon/favicon.ico" type="image/x-icon">\n\t<link rel="apple-touch-icon" sizes="57x57" href="/favicon/apple-icon-57x57.png">\n\t<link rel="apple-touch-icon" sizes="60x60" href="/favicon/apple-icon-60x60.png">\n\t<link rel="apple-touch-icon" sizes="72x72" href="/favicon/apple-icon-72x72.png">\n\t<link rel="apple-touch-icon" sizes="76x76" href="/favicon/apple-icon-76x76.png">\n\t<link rel="apple-touch-icon" sizes="114x114" href="/favicon/apple-icon-114x114.png">\n\t<link rel="apple-touch-icon" sizes="120x120" href="/favicon/apple-icon-120x120.png">\n\t<link rel="apple-touch-icon" sizes="144x144" href="/favicon/apple-icon-144x144.png">\n\t<link rel="apple-touch-icon" sizes="152x152" href="/favicon/apple-icon-152x152.png">\n\t<link rel="apple-touch-icon" sizes="180x180" href="/favicon/apple-icon-180x180.png">\n\t<link rel="icon" type="image/png" sizes="192x192"  href="/favicon/android-icon-192x192.png">\n\t<link rel="icon" type="image/png" sizes="32x32" href="/favicon/favicon-32x32.png">\n\t<link rel="icon" type="image/png" sizes="96x96" href="/favicon/favicon-96x96.png">\n\t<link rel="icon" type="image/png" sizes="16x16" href="/favicon/favicon-16x16.png">\n\t<link rel="manifest" href="/favicon/manifest.json">\n\t<meta name="msapplication-TileColor" content="#ffffff">\n\t<meta name="msapplication-TileImage" content="/favicon/ms-icon-144x144.png">\n\t<meta name="theme-color" content="#ffffff">\n\t\n\t<!-- og image -->\n\t<meta property="og:image" content="http://www.worldometers.info/img/worldometers-fb.jpg" />\n\t\n    <!-- Bootstrap -->\n    <link href="/css/bootstrap.min.css" rel="stylesheet">\n\t<link href="/wm16.css" rel="stylesheet">\n\t<!-- fon
```

In [19]:
```
url = 'https://www.worldometers.info/coronavir/#countries'
```

Out[19]:
```
'<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"\n"http://www.w3.org/TR/html4/loose.dtd">\n<html>\n<head>\n<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">\n<title>404 Not Found</title>\n<style type="text/css">\n<!--\n.style1 {\n\tfont-family: Arial, Helvetica, sans-serif;\n\tcolor: #CCCCCC;\n\tfont-size: 18px;\n}\n.style3 {font-family: Arial, Helvetica, sans-serif; color: #999999; font-size: 36px; }\n.style4 {\n\tfont-size: 90px;\n\tcolor: #CCCCCC;\n}\n-->\n</style>\n</head>\n\n<body>\n\n<script>\n  (function(i,s,o,g,r,a,m){i[\'GoogleAnalyticsObject\']=r;i[r]=i[r]||function(){\n  (i[r].q=i[r].q||[]).push(arguments)},i[r].l=1*new Date();a=s.createElement(o),\n  m=s.getElementsByTagName(o)[0];a.async=1;a.src=g;m.parentNode.insertBefore(a,m)\n  })(window,document,\'script\',\'//www.google-analytics.com/analytics.js\',\'ga\');\n\n  ga(\'create\', \'UA-1438574-2\', \'auto\');\nga(\'send\', \'pageview\', \'404.html?page=\'+ document.location.pathname + document.location.search +\'&from=\' + document.referrer);\n\n</script>\n\n<div align="center">\n  <p> </p>\n  <table width="48%"  border="0" cellspacing="0" cellpadding="0">\n    <tr>\n      <td height="142" valign="top"><p class="style3"><span class="style4">Not Found<br> \n          </span>the requested webpage was not found<br>\n      </p>\n        <p class="style3"><a href="/"><img src="/img/logo-fr-s.gif" alt="Worldometers Home" width="360" height="45" border="0"></a><br>\n          <br>\n          <span class="style1">We might have removed the page from our website, changed its URL, or the URL you typed in or followed from a link is incorrect.<br>\n          <br>\n        </span><span class="style1"><a href="/contact/">Contact us</a> for more information or go to <a href="/">Worldometers home page</a></span></p>\n      <p class="style3"> </p></td>\n    </tr>\n  </table>\n  <p><span class="style3"><br>\n  </span></p>\n  <p> </p>\n  <p> </p>\n  <p><br>\n  </p>\n</div>\n</body>\n</html>\n'
```

In [20]:
```
"""
url = "https://www.ambitionbox.com/list-of-companies?page=1"
requests.get(url).text
"""
```

Out[20]:
```
'\nurl = "https://www.ambitionbox.com/list-of-companies?page=1"\nrequests.get(url).text\n'
```

In [21]:
```
url = 'https://www.worldometers.info/coronavirus/#countries'
req = Request(url)
webpage = urlopen(req)
```

```
<http.client.HTTPResponse object at 0x000001E39B62D2A0>
```

In [22]:
```
page_soup = BeautifulSoup(webpage, 'html.parser') # "lxml"
#page_soup = BeautifulSoup(webpage, 'lxml')
```

```
<script class="init" type="text/javascript">
$(document).ready(function() {
        $('#popbycountry').dataTable();
} );
        </script>
</link></meta></head>
<body>
<!-- Google tag (gtag.js) -->
<script async="" src="https://www.googletagmanager.com/gtag/js?id=G-ZDP3BFSX60"></script>
<script>
  window.dataLayer = window.dataLayer || [];
  function gtag(){dataLayer.push(arguments);}
  gtag('js', new Date());

  gtag('config', 'G-ZDP3BFSX60');
</script>
<script async="" crossorigin="anonymous" src="https://pagead2.googlesyndication.com/pagead/js/adsbygoogle.js?client=ca-pub-37
01697624350410"></script>
<style type="text/css">
```

In [23]:
```
today = datetime.now()
```

```
2024-02-25 21:22:54.500747
```

In [24]:
```
yesterday = "%s %d, %d" %(date.today().strftime('%b'), today.day-1, today.year )
print(yesterday)
print()
```

```
Feb 24, 2024

2024-02-25 21:22:54.500747
```

In [25]:
```
date.today().strftime('%b')
```

Out[25]:
```
'Feb'
```

In [26]:
```
table = page_soup.findAll("table",{'id':'main_table_countries_yesterday'})
```

Out[26]:
```
[<table class="table table-bordered table-hover main_table_countries" id="main_table_countries_yesterday" style="width:100%;m
argin-top: 0px !important;display:none;">
 <thead>
 <tr>
 <th width="1%">#</th>
 <th width="100">Country,<br/>Other</th>
 <th width="20">Total<br/>Cases</th>
 <th width="30">New<br/>Cases</th>
 <th width="30">Total<br/>Deaths</th>
 <th width="30">New<br/>Deaths</th>
 <th width="30">Total<br/>Recovered</th>
 <th width="30">New<br/>Recovered</th>
 <th width="30">Active<br/>Cases</th>
 <th width="30">Serious,<br/>Critical</th>
 <th width="30">Tot Cases/<br/>1M pop</th>
 <th width="30">Deaths/<br/>1M pop</th>
 <th width="30">Total<br/>Tests</th>
 <th width="30">Tests/<br/>
 <nobr>1M pop</nobr>
```

In [27]:
```
containers = table[0].findAll('tr',{"style":""})
title = containers[0]
```

Out[27]:
```
<tr>
<th width="1%">#</th>
<th width="100">Country,<br/>Other</th>
<th width="20">Total<br/>Cases</th>
<th width="30">New<br/>Cases</th>
<th width="30">Total<br/>Deaths</th>
<th width="30">New<br/>Deaths</th>
<th width="30">Total<br/>Recovered</th>
<th width="30">New<br/>Recovered</th>
<th width="30">Active<br/>Cases</th>
<th width="30">Serious,<br/>Critical</th>
<th width="30">Tot Cases/<br/>1M pop</th>
<th width="30">Deaths/<br/>1M pop</th>
<th width="30">Total<br/>Tests</th>
<th width="30">Tests/<br/>
<nobr>1M pop</nobr>
</th>
<th width="30">Population</th>
<th style="display:none" width="30">Continent</th>
<th width="30">1 Case<br/>every X ppl</th><th width="30">1 Death<br/>every X ppl</th><th width="30">1 Test<br/>every X ppl</th>
<th width="30">New Cases/1M pop</th>
<th width="30">New Deaths/1M pop</th>
<th width="30">Active Cases/1M pop</th>
</tr>
```

In [28]:
```
containers[1]
```

Out[28]:
```
<tr class="total_row_world">
<td></td>
<td style="text-align:left;">World</td>
<td>703,659,398</td>
<td>+4,709</td>
<td>6,985,937</td>
<td>+74</td>
<td>674,392,205</td>
<td>+17,897</td>
<td>22,281,256</td>
<td>35,667</td>
<td>90,273</td>
<td>896.2</td>
<td></td>
<td></td>
<td></td>
<td data-continent="all" style="display:none">All</td>
<!-- 1 Case every X -->
<td>
</td>
<!-- 1 Death every X -->
<td></td>
<!-- 1 test every X -->
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
```

In [29]:
```
del containers[0]
```

In [31]:
```
table = page_soup.findAll("table",{'id':'main_table_countries_yesterday'})
containers = table[0].findAll('tr',{"style":""})
title = containers[0]
del containers[0]


all_data  = []
clean = True

for country in containers:
    country_data = []
    country_container = country.findAll('td')
    if country_container[1].text =='China':
        continue
    for i in range(1, len(country_container)):
        final_feature = country_container[i].text
        if clean:
            if i!= 1 and i!=len(country_container)-1:
                final_feature = final_feature.replace(",","")
                if final_feature.find("+") != -1:
                    final_feature = final_feature.replace("+","")
                    final_feature = float(final_feature)
                #if final_feature.find("_") != -1:
                    #final_feature = float(final_feature)*-1
            if final_feature =='N/A':
                final_feature = 0
            elif final_feature =="" or final_feature == " ":
                final_feature= -1
            country_data.append(final_feature)
```

In [32]:
```
all_data
```

```
  '90273',
  '896.2',
  -1,
  -1,
  -1,
  'All',
  '\n',
  -1,
  -1,
  -1,
  -1,
  -1],
 ['USA',
  '111424589',
  2660.0,
  '1199434                                ',
  6.0,
  '109118877',
  9283.0,
  '1106278',
```

In [33]:
```
df = pd.DataFrame(all_data)
```

Out[33]:

	1	2	3	4	5	6	7	8	9	...	11	12	13	14	15	16	17	18	19	20
	703659398	4709.0	6985937	74.0	674392205	17897.0	22281256	35667	90273	...	-1	-1	-1	All	\n	-1	-1	-1	-1	-1
	111424589	2660.0	1199434	6.0	109118877	9283.0	1106278	1743	332804	...	1186764898	3544642	334805269	North America	3	279	0	8	0.02	3,304
	45028887	322.0	533477	1.0	0	0.0	0	0	32012	...	935879495	665334	1406631776	Asia	31	2637	2	0.2	-1	0.6
	38820489	431.0	182541	33.0	38240600	-1.0	397348	0	462790	...	122332384	1458359	83883596	Europe	2	460	1	5	0.4	4,737
	38452504	-1.0	709963	-1.0	36249161	-1.0	1493380	0	178555	...	63776166	296146	215353593	South America	6	303	3	-1	-1	6,935

columns

In [34]:
```
df.drop([15,16,17,18,19,20], axis=1, inplace=True)
```

In [35]:
```
df.head()
```

Out[35]:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	World	703659398	4709.0	6985937	74.0	674392205	17897.0	22281256	35667	90273	896.2	-1	-1	-1	All
1	USA	111424589	2660.0	1199434	6.0	109118877	9283.0	1106278	1743	332804	3582	1186764898	3544642	334805269	North America
2	India	45028887	322.0	533477	1.0	0	0.0	0	0	32012	379	935879495	665334	1406631776	Asia
3	Germany	38820489	431.0	182541	33.0	38240600	-1.0	397348	0	462790	2176	122332384	1458359	83883596	Europe
4	Brazil	38452504	-1.0	709963	-1.0	36249161	-1.0	1493380	0	178555	3297	63776166	296146	215353593	South America

In [36]:
```
column_labels = ['Country', 'Total Cases','New Cases','Total Deaths','New Deaths','Total Recovered',
                 'New Recovered','Active Cases','Serious Critical','Total Cases/1M pop',
```

In [37]:
```
df.columns = column_labels
```

Out[37]:

	Country	Total Cases	New Cases	Total Deaths	New Deaths	Total Recovered	New Recovered	Active Cases	Serious Critical	Total Cases/1M pop	Death/1M pop	Total Tests	Tests/1M pop	Population	
0	World	703659398	4709.0	6985937	74.0	674392205	17897.0	22281256	35667	90273	896.2	-1	-1	-1	
1	USA	111424589	2660.0	1199434	6.0	109118877	9283.0	1106278	1743	332804	3582	1186764898	3544642	334805269	Nc
2	India	45028887	322.0	533477	1.0	0	0.0	0	0	32012	379	935879495	665334	1406631776	
3	Germany	38820489	431.0	182541	33.0	38240600	-1.0	397348	0	462790	2176	122332384	1458359	83883596	
4	Brazil	38452504	-1.0	709963	-1.0	36249161	-1.0	1493380	0	178555	3297	63776166	296146	215353593	So
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
200	Saint Helena	2166	-1.0	-1	-1.0	2	-1.0	2164	-1	354211	-1	-1	-1	6115	
201	Montserrat	1403	-1.0	8	-1.0	1376	-1.0	19	-1	282578	1611	17762	3577442	4965	Nc
202	Niue	1059	-1.0	-1	-1.0	1056	-1.0	3	-1	652898	-1	-1	-1	1622	Austra
203	Tokelau	80	-1.0	-1	-1.0	-1	-1.0	80	-1	58055	-1	-1	-1	1378	Austra
204	Total:	703659398	4709.0	6985937	74.0	674392205	17897.0	22281256	35667	90272.9	896.2	-1	-1	-1	

205 rows × 15 columns

In [38]:
```
df.info()
```

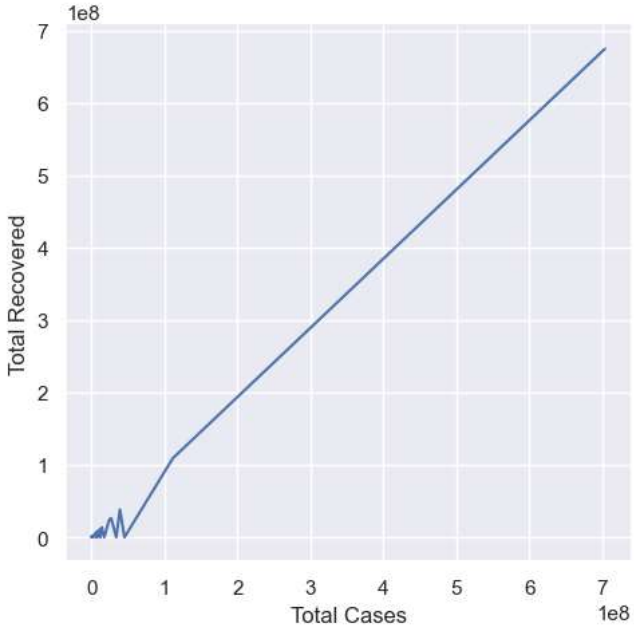
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 205 entries, 0 to 204
Data columns (total 15 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   Country             205 non-null    object
 1   Total Cases         205 non-null    object
 2   New Cases           205 non-null    float64
 3   Total Deaths        205 non-null    object
 4   New Deaths          205 non-null    float64
 5   Total Recovered     205 non-null    object
 6   New Recovered       205 non-null    float64
 7   Active Cases        205 non-null    object
 8   Serious Critical    205 non-null    object
 9   Total Cases/1M pop  205 non-null    object
 10  Death/1M pop        205 non-null    object
 11  Total Tests         205 non-null    object
 12  Tests/1M pop        205 non-null    object
 13  Population          205 non-null    object
 14  Continent           205 non-null    object
dtypes: float64(3), object(12)
memory usage: 24.1+ KB
```

In [39]:
```
for label in df.columns:
    if label !='Country' and label !='Continent':
```

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 205 entries, 0 to 204
Data columns (total 15 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   Country             205 non-null    object
 1   Total Cases         205 non-null    int64
 2   New Cases           205 non-null    float64
 3   Total Deaths        205 non-null    int64
 4   New Deaths          205 non-null    float64
 5   Total Recovered     205 non-null    int64
 6   New Recovered       205 non-null    float64
 7   Active Cases        205 non-null    int64
 8   Serious Critical    205 non-null    int64
 9   Total Cases/1M pop  205 non-null    float64
 10  Death/1M pop        205 non-null    float64
 11  Total Tests         205 non-null    int64
 12  Tests/1M pop        205 non-null    int64
 13  Population          205 non-null    int64
 14  Continent           205 non-null    object
dtypes: float64(5), int64(8), object(2)
memory usage: 24.1+ KB
```

In [41]:
```
sns.relplot(x='Total Cases', y='Total Recovered', data=df, kind='line')
```



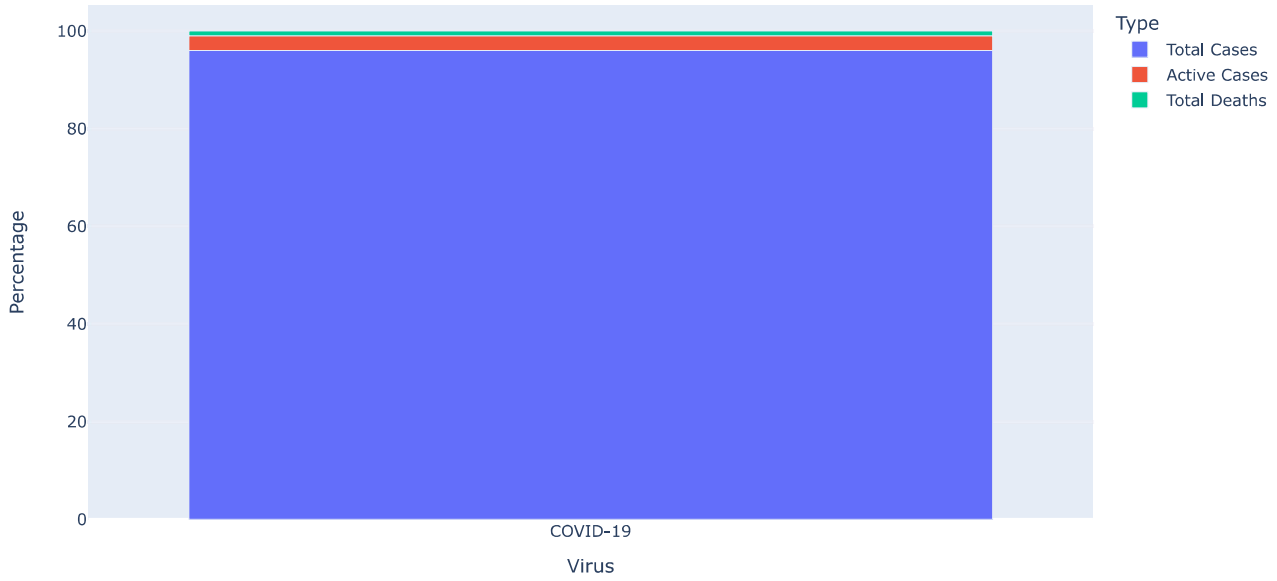
In [42]:
```
df["%inc Cases"] = df['New Cases']/df['Total Cases']*100
df["%inc Deaths"] = df['New Deaths']/df['Total Deaths']*100
```

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

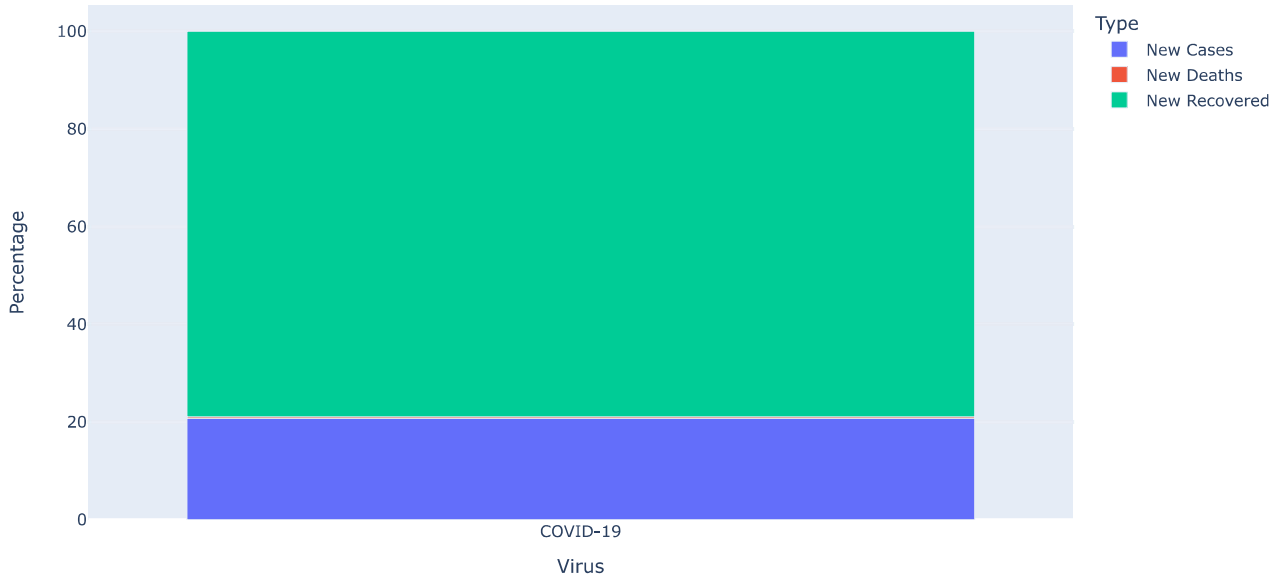
Out[43]:

	Country	Total Cases	New Cases	Total Deaths	New Deaths	Total Recovered	New Recovered	Active Cases	Serious Critical	Total Cases/1M pop	Death/1M pop	Total Tests	Tests/1M pop	Population	Continent
0	World	703659398	4709.0	6985937	74.0	674392205	17897.0	22281256	35667	90273.0	896.2	-1	-1	-1	Al
1	USA	111424589	2660.0	1199434	6.0	109118877	9283.0	1106278	1743	332804.0	3582.0	1186764898	3544642	334805269	North America
2	India	45028887	322.0	533477	1.0	0	0.0	0	0	32012.0	379.0	935879495	665334	1406631776	Asia
3	Germany	38820489	431.0	182541	33.0	38240600	-1.0	397348	0	462790.0	2176.0	122332384	1458359	83883596	Europe
4	Brazil	38452504	-1.0	709963	-1.0	36249161	-1.0	1493380	0	178555.0	3297.0	63776166	296146	215353593	South America

In [44]:
```
cases = df[['Total Cases','Active Cases','Total Deaths']].loc[0]
cases_df = pd.DataFrame(cases).reset_index()
cases_df.columns = ['Type','Total']
cases_df['Percentage'] = np.round(100*cases_df['Total']/np.sum(cases_df['Total']),2)
cases_df['Virus'] = ["COVID-19" for i in range(len(cases_df))]
fig = px.bar(cases_df, x='Virus', y='Percentage', color='Type', hover_data=['Total'])
```



In [45]:
```
cases = df[['New Cases','New Deaths','New Recovered']].loc[0]
cases_df = pd.DataFrame(cases).reset_index()
cases_df.columns = ['Type','Total']
cases_df['Percentage'] = np.round(100*cases_df['Total']/np.sum(cases_df['Total']),2)
cases_df['Virus'] = ["COVID-19" for i in range(len(cases_df))]
fig = px.bar(cases_df, x='Virus', y='Percentage', color='Type', hover_data=['Total'])
```



In [46]:
```
df1 = df.drop([len(df)-1])
country_df = df1.drop([0])
```

Out[46]:

	Country	Total Cases	New Cases	Total Deaths	New Deaths	Total Recovered	New Recovered	Active Cases	Serious Critical	Total Cases/1M pop	Death/1M pop	Total Tests	Tests/1M pop	Population	
1	USA	111424589	2660.0	1199434	6.0	109118877	9283.0	1106278	1743	332804.0	3582.0	1186764898	3544642	334805269	Nort
2	India	45028887	322.0	533477	1.0	0	0.0	0	0	32012.0	379.0	935879495	665334	1406631776	
3	Germany	38820489	431.0	182541	33.0	38240600	-1.0	397348	0	462790.0	2176.0	122332384	1458359	83883596	
4	Brazil	38452504	-1.0	709963	-1.0	36249161	-1.0	1493380	0	178555.0	3297.0	63776166	296146	215353593	Sout
5	Japan	33803572	-1.0	74694	-1.0	0	0.0	0	0	269169.0	595.0	100414883	799578	125584838	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
199	Tuvalu	2943	-1.0	1	-1.0	0	0.0	0	0	243909.0	83.0	-1	-1	12066	Australi
200	Saint Helena	2166	-1.0	-1	-1.0	2	-1.0	2164	-1	354211.0	-1.0	-1	-1	6115	
201	Montserrat	1403	-1.0	8	-1.0	1376	-1.0	19	-1	282578.0	1611.0	17762	3577442	4965	Nort
202	Niue	1059	-1.0	-1	-1.0	1056	-1.0	3	-1	652898.0	-1.0	-1	-1	1622	Australi
203	Tokelau	80	-1.0	-1	-1.0	-1	-1.0	80	-1	58055.0	-1.0	-1	-1	1378	Australi

203 rows × 18 columns

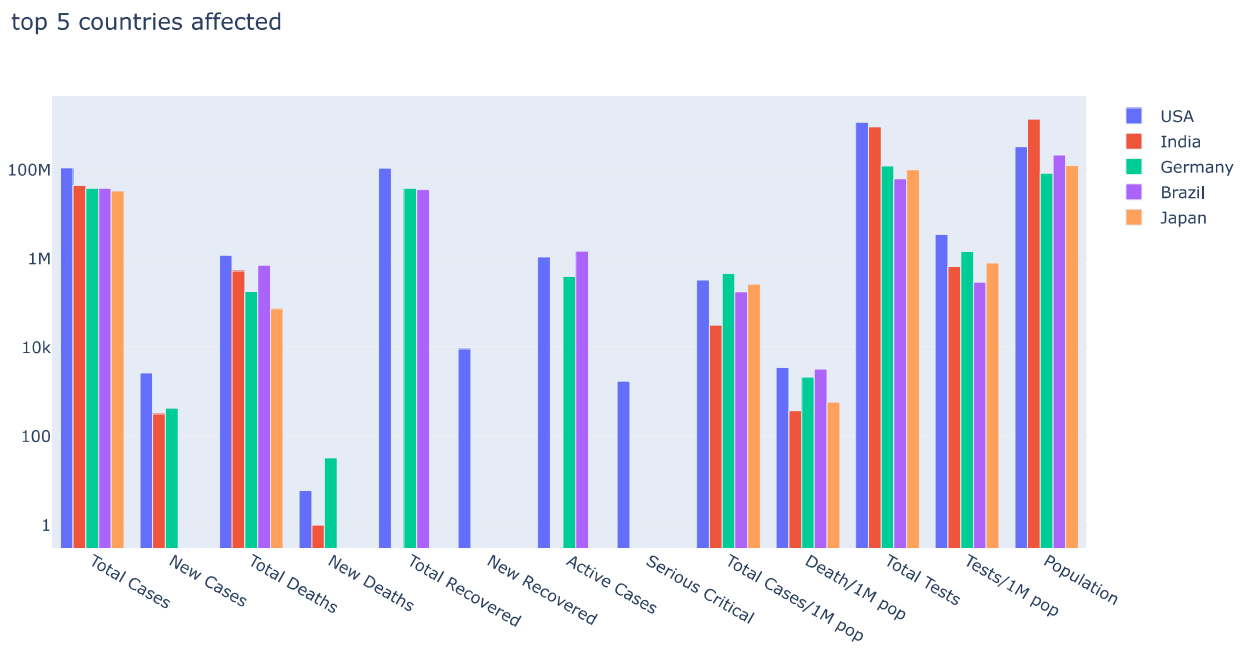
In [47]:
```
country_df.shape
```

Out[47]: (203, 18)

In [48]:
```
Top_Countries = 5
country = country_df.columns[1:14]

fig = go.Figure()
c = 0

for i in country_df.index:
    if c <Top_Countries:
        fig.add_trace(go.Bar(name=country_df['Country'][i],
                             x=country, y = country_df.loc[i][1:14]))
    else:
        break
    c +=1
fig.update_layout(title={"text":f'top {Top_Countries} countries affected'}, yaxis_type="log")
```



In [49]:
```
df['Continent'].value_counts()
```

Out[49]:
```
Africa                53
Asia                  42
Europe                42
North America         37
Australia/Oceania     19
South America         10
All                    2
Name: Continent, dtype: int64
```

In [50]:
```
continent_df = df.groupby('Continent').sum().drop('All')
continent_df = continent_df.reset_index()
```

Out[50]:

	Continent	Total Cases	New Cases	Total Deaths	New Deaths	Total Recovered	New Recovered	Active Cases	Serious Critical	Total Cases/1M pop	Death/1M pop	Total Tests	Tests/1M pop	Population
0	Africa	12727994	-53.0	253481	-53.0	9084723	-46.0	330916	357	1961359.0	16332.0	110416058	10774326	1356121214
1	Asia	168945577	422.0	1469083	-40.0	56200931	568.0	1411644	345	6043553.0	30919.0	2177719443	82189893	3116134094
2	Australia/Oceania	14814093	765.0	32677	12.0	2748000	-13.0	27962	2	4726665.0	9376.0	91668032	21284787	43410351
3	Europe	196334124	771.0	1752368	-2.0	185119180	6890.0	897709	429	16936961.0	115196.0	2303868918	179486756	617154618
4	North America	131450788	2624.0	1675058	-30.0	125732553	10098.0	1666052	1872	8563768.0	57939.0	1325230081	96278274	598014701
5	South America	54104183	-10.0	976816	-10.0	50549696	-8.0	1679044	622	1657390.0	21778.0	167578055	9486465	339883101

In [51]:
```
cases_list = ["Total Cases","Total Deaths","Total Recovered","New Cases","New Deaths"]
```

In [52]:
```
def continent_visaulization(v_list):
    for label in v_list:
        c_df = continent_df[['Continent', label]]
        c_df['Percentage'] = np.round(100*c_df[label]/np.sum(c_df[label]),2)
        c_df['Virus'] = ["COVID-19" for i in range(len(c_df))]
        fig = px.bar(c_df, x='Virus', y='Percentage', color='Continent', hover_data=[label])
        fig.update_layout(title={"text": f"{label}"})

        fig.show()
```

In [53]:
```
continent_visaulization(cases_list)
```

In [ ]: