PYTHON 資料科學期末專題

[提示]

1. 用Pandas讀入時有時需要注意encoding參數

    1.1利用dataframe 去 Creating new feature "Active_case"; Active_case = Confirmed - Deaths – Recovered

    1.2 需要注意異常值與缺失值的處理, 注意資料區間, 評估值區間差異過大的問題

    1.3 資料的分類, 以期可以分別繪製比對圖形; 繪製出交互作用的地圖

[基本目標]

1. 繪製出全球的感染與康復人數, 以國家別區分
2. 利用地理資訊圖表繪製出全球的感染與康復人數

參考範例

同學可以參照範例執行, 有問題隨時提問

Table of contents

In [1]:

```
# 載入所需要的套件
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import plotly.graph_objects as go
import plotly.express as px
pd.set_option('display.max_rows', None)
from plotly.subplots import make_subplots
import seaborn as sns
import datetime
```

In [2]:

```
import warnings
warnings.simplefilter("ignore")
```

In [3]:

```
#pyecharts table
from pyecharts.components import Table
from pyecharts.options import ComponentTitleOpts
from pyecharts.commons.utils import JsCode

#pyecharts pie chart
from pyecharts import options as opts
from pyecharts.globals import ThemeType

from pyecharts.charts import Pie
from pyecharts.charts import Line
from pyecharts.charts import Scatter
from pyecharts.charts import Bar
from pyecharts.charts import Map
from pyecharts.charts import Timeline, Grid
```

進行資料讀入, 檢視與前處理

In [4]:

```
# 利用 PANDAS 讀入DATA
data = pd.read_csv('covid_19_data.csv')
```

In [5]:

```
# 檢視資料集
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 156292 entries, 0 to 156291
Data columns (total 8 columns):
 #   Column           Non-Null Count   Dtype  
---  ------           --------------   -----  
 0   SNo              156292 non-null  int64  
 1   ObservationDate  156292 non-null  object 
 2   Province/State   111979 non-null  object 
 3   Country/Region   156292 non-null  object 
 4   Last Update      156292 non-null  object 
 5   Confirmed        156292 non-null  float64
 6   Deaths           156292 non-null  float64
 7   Recovered        156292 non-null  float64
dtypes: float64(3), int64(1), object(4)
memory usage: 9.5+ MB
```

In [6]:

```
# 缺失值處理
#Percentage of NAN Values, 計算缺失值在資料內的比例
NAN = [(c, data[c].isna().mean()*100) for c in data]
NAN = pd.DataFrame(NAN, columns=["column_name", "percentage"])
NAN
```

Out[6]:

	column_name	percentage
0	SNo	0.000000
1	ObservationDate	0.000000
2	Province/State	28.352699
3	Country/Region	0.000000
4	Last Update	0.000000
5	Confirmed	0.000000
6	Deaths	0.000000
7	Recovered	0.000000

In [7]:

```
# 34 % of Province/State are unknown, 所以我們要把 NAN 的值替換成 Unknown.
data["Province/State"]= data["Province/State"].fillna('Unknown')
```

In [8]:

```
#查看資料
data.head()
```

Out[8]:

	SNo	ObservationDate	Province/State	Country/Region	Last Update	Confirmed	Deaths	Recovered
0	1	01/22/2020	Anhui	Mainland China	1/22/2020 17:00	1.0	0.0	0.0
1	2	01/22/2020	Beijing	Mainland China	1/22/2020 17:00	14.0	0.0	0.0
2	3	01/22/2020	Chongqing	Mainland China	1/22/2020 17:00	6.0	0.0	0.0
3	4	01/22/2020	Fujian	Mainland China	1/22/2020 17:00	1.0	0.0	0.0
4	5	01/22/2020	Gansu	Mainland China	1/22/2020 17:00	0.0	0.0	0.0

In [9]:

```
# 資料型別轉換，利於 ARRAY
# 把下列資料換成 INT 整數型別 : "Confirmed","Deaths" and "Recovered"

data[["Confirmed","Deaths","Recovered"]] =data[["Confirmed","Deaths","Recovered"]].astype(int)
```

In [10]:

```
# 為了識別簡易，把 "Mainland China" 用 "China" 替代
data['Country/Region'] = data['Country/Region'].replace('Mainland China', 'China')
```

In [11]:

```
# 原始資料集並無感染的欄位，利用資料處理新增一項目 "Active_case"
# Active_case = Confirmed - Deaths - Recovered
data['Active_case'] = data['Confirmed'] - data['Deaths'] - data['Recovered']
data.head()
```

Out[11]:

	SNo	ObservationDate	Province/State	Country/Region	Last Update	Confirmed	Deaths	Recovered	Active_case
0	1	01/22/2020	Anhui	China	1/22/2020 17:00	1	0	0	1
1	2	01/22/2020	Beijing	China	1/22/2020 17:00	14	0	0	14
2	3	01/22/2020	Chongqing	China	1/22/2020 17:00	6	0	0	6
3	4	01/22/2020	Fujian	China	1/22/2020 17:00	1	0	0	1
4	5	01/22/2020	Gansu	China	1/22/2020 17:00	0	0	0	0

In [12]:

```
# 更新INDEX
Data = data[data['ObservationDate'] == max(data['ObservationDate'])].reset_index()
data.head()
```

Out[12]:

	SNo	ObservationDate	Province/State	Country/Region	Last Update	Confirmed	Deaths	Recovered	Active_case
0	1	01/22/2020	Anhui	China	1/22/2020 17:00	1	0	0	1
1	2	01/22/2020	Beijing	China	1/22/2020 17:00	14	0	0	14
2	3	01/22/2020	Chongqing	China	1/22/2020 17:00	6	0	0	6
3	4	01/22/2020	Fujian	China	1/22/2020 17:00	1	0	0	1
4	5	01/22/2020	Gansu	China	1/22/2020 17:00	0	0	0	0

開始使用資料視覺化顯示結果

Covid-19 - In the world

In [13]:

```
# 整理資料
Data_world = Data.groupby(["ObservationDate"])[["Confirmed","Active_case","Recovered","Deaths"]].sum().reset_index()
Data_world.head()
```

Out[13]:

	ObservationDate	Confirmed	Active_case	Recovered	Deaths
0	11/15/2020	54370186	18097899	34955148	1317139

In [14]:

```
table = Table()
headers = ["Last Update","Confirmed","Active cases","Recovered","Deaths"]
table.add(headers, [Data_world.loc[0,["ObservationDate","Confirmed","Active_case","Recovered","Deaths"]]])
table.set_global_opts(
    title_opts=ComponentTitleOpts(title="Coronavirus in the word : ")
)
table.render_notebook()
```

Out[14]:

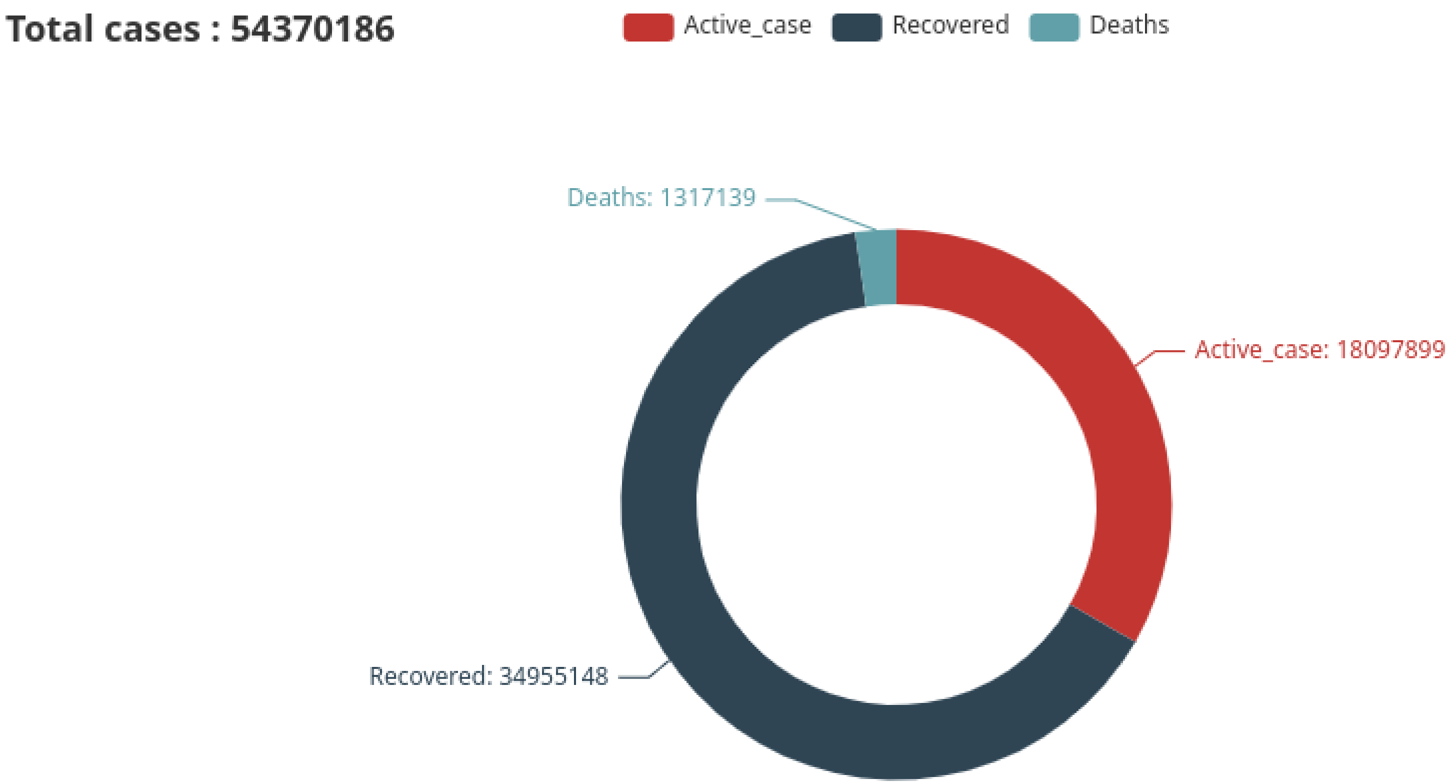
Coronavirus in the word :

Last Update	Confirmed	Active cases	Recovered	Deaths
11/15/2020	54370186	18097899	34955148	1317139

In [15]:

```
labels = ["Active cases","Recovered","Deaths"]
idxies = ["Active_case","Recovered","Deaths"]
data_pair = [ [z[0],int(z[1])] for z in zip(idxies,
                                            Data_world.loc[0, ["Active_case","Recovered","Deaths"]].values.tolist())
]
(
    Pie()
    .add("Cases", data_pair, radius=["40%", "55%"])
    .set_global_opts(title_opts=opts.TitleOpts(title='Total cases : '+str(Data_world["Confirmed"][0])))
    .set_series_opts(label_opts=opts.LabelOpts(formatter="{b}: {c}"))
    .render_notebook()
)
```

Out[15]:



In [16]:

```
# Evolution of coronavirus over time
data_over_time= data.groupby(["ObservationDate"])[["Confirmed","Active_case","Recovered","Deaths"]].sum().reset_index().sort_values("ObservationDate",ascending=True).reset_index(drop=True)
data_over_time.head()
```

Out[16]:

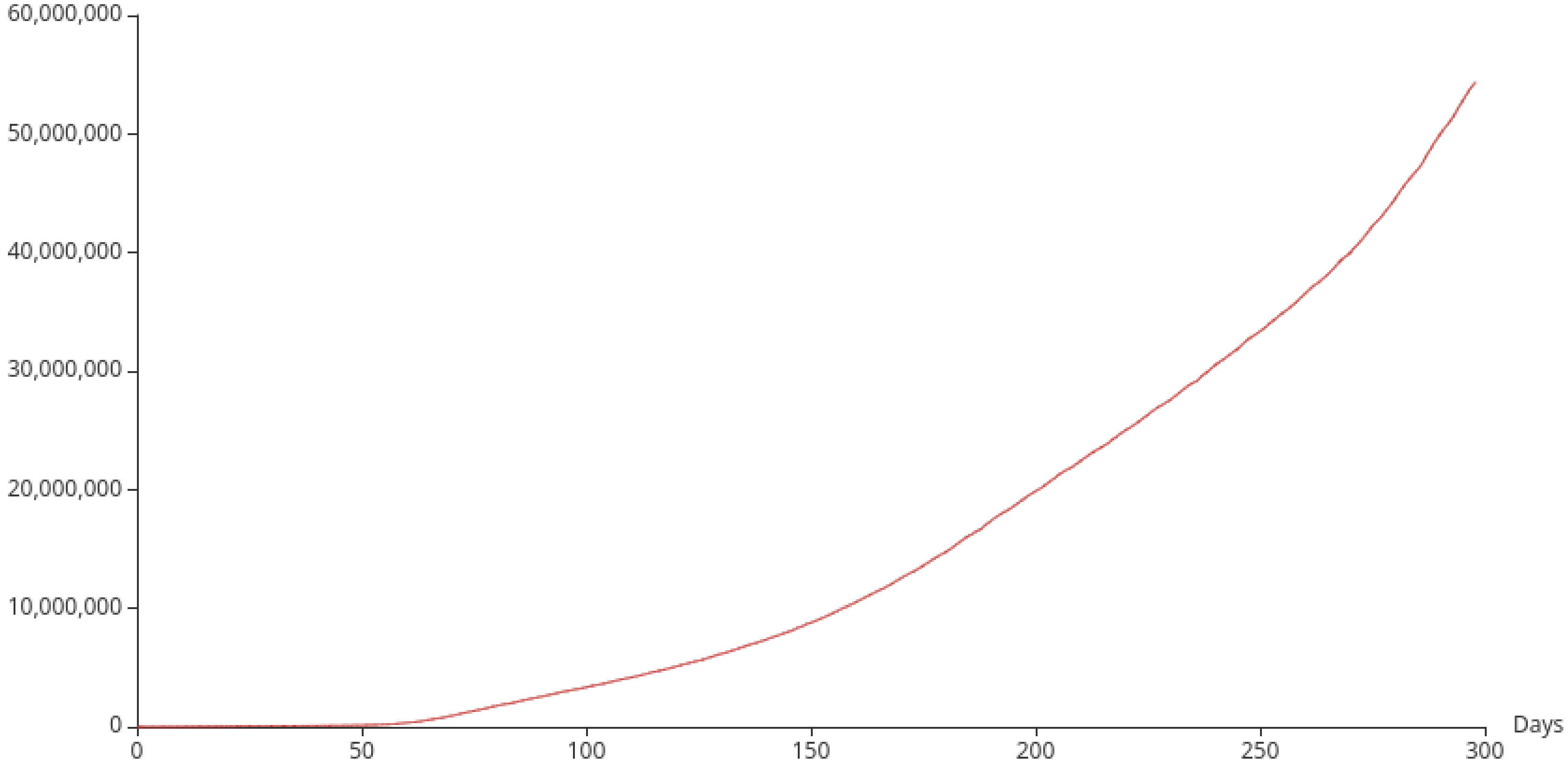
	ObservationDate	Confirmed	Active_case	Recovered	Deaths
0	01/22/2020	555	510	28	17
1	01/23/2020	653	605	30	18
2	01/24/2020	941	879	36	26
3	01/25/2020	1438	1357	39	42
4	01/26/2020	2118	2010	52	56

In [17]:

```
(
    Line()
    .set_global_opts(
        title_opts=opts.TitleOpts(title="Evolution of Confirmed cases over time in the word"),
        xaxis_opts=opts.AxisOpts(type_='value', name="Days"),
    )
    .add_xaxis(data_over_time.index.tolist())
    .add_yaxis(
        series_name="",
        y_axis=data_over_time['Confirmed'].tolist(),
        is_symbol_show=False,
        is_smooth=True,
        label_opts=opts.LabelOpts(is_show=False),
    )
    .render_notebook()
)
```

```
Out[17]:
```

Evolution of Confirmed cases over time in the word

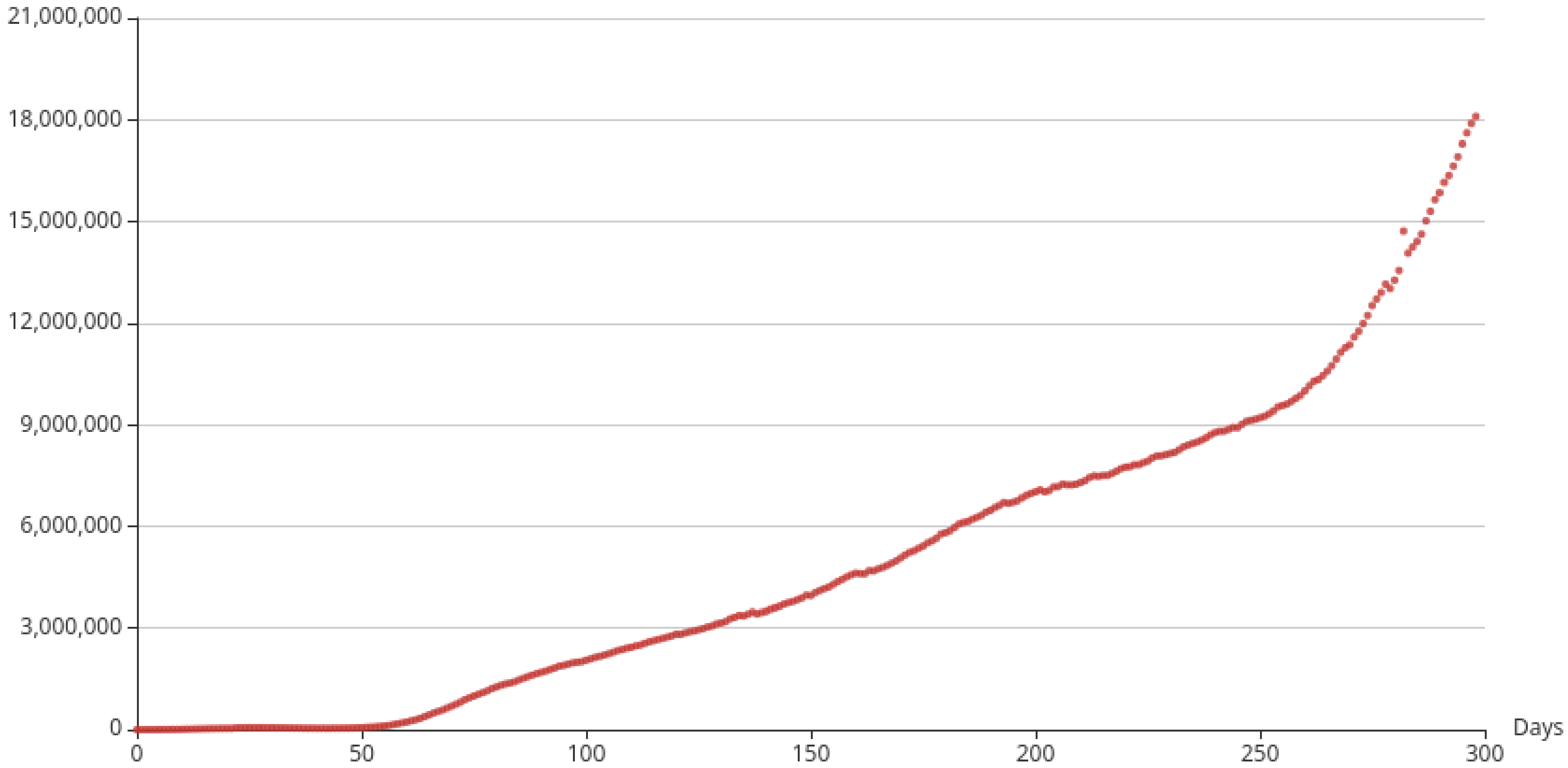


In [18]:

```
(
    Scatter()
    .set_global_opts(
        title_opts=opts.TitleOpts(title="Evolution of Active cases over time in the word"),
        xaxis_opts=opts.AxisOpts(type_='value', name="Days"),
        yaxis_opts=opts.AxisOpts(
            type_="value",
            axistick_opts=opts.AxisTickOpts(is_show=True),
            splitline_opts=opts.SplitLineOpts(is_show=True),
        ),
        tooltip_opts=opts.TooltipOpts(is_show=False),
    )
    .add_xaxis(data_over_time.index.tolist())
    .add_yaxis(
        series_name="",
        y_axis=data_over_time['Active_case'].tolist(),
        symbol_size=4,
        label_opts=opts.LabelOpts(is_show=False),

    )
    .render_notebook()
)
```

```
Out[18]:
```

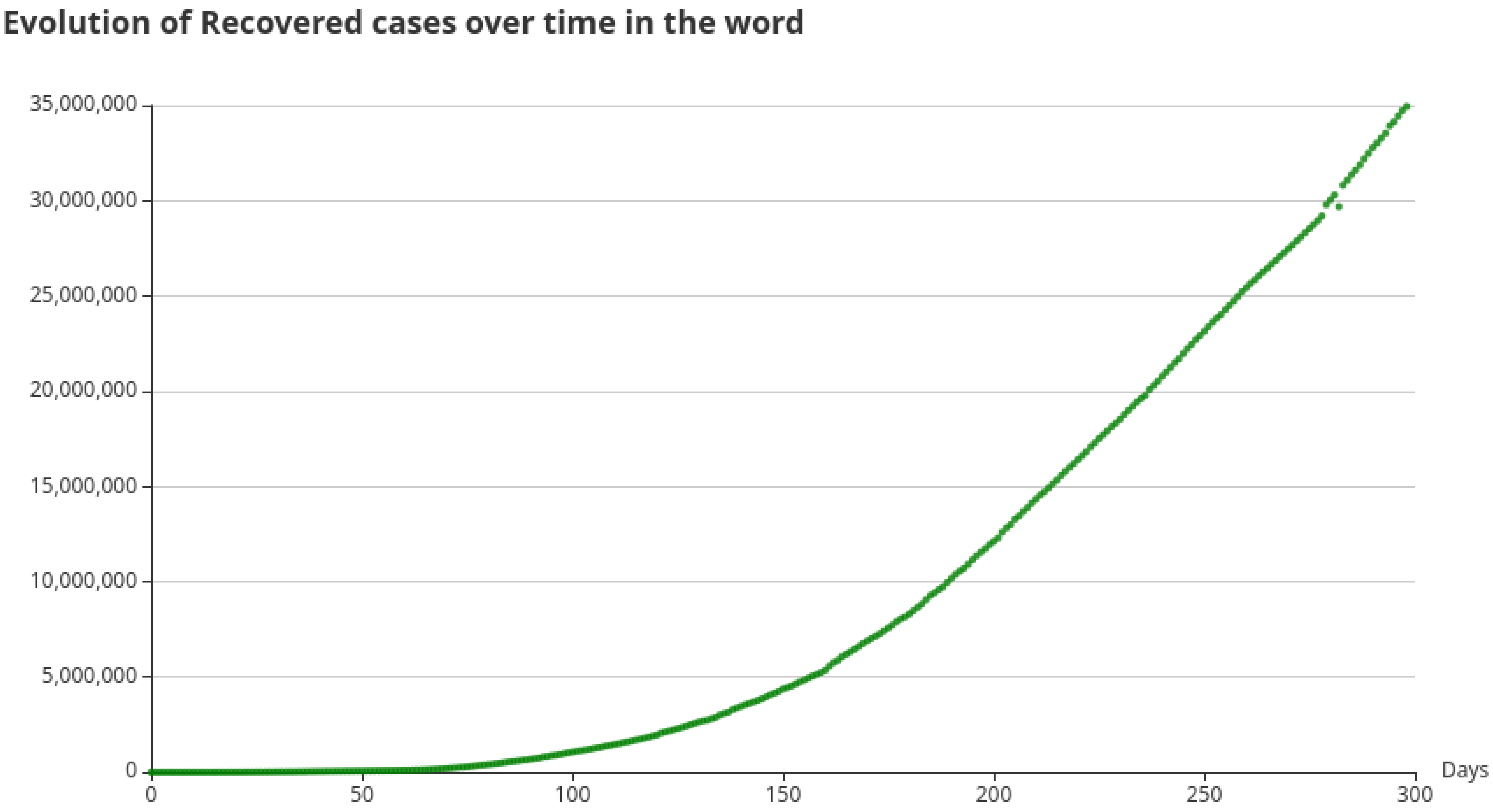
Evolution of Active cases over time in the word



In [19]:

```
(
    Scatter()
    .set_global_opts(
        title_opts=opts.TitleOpts(title="Evolution of Recovered cases over time in the word"),
        xaxis_opts=opts.AxisOpts(type_='value', name="Days"),
        yaxis_opts=opts.AxisOpts(
            type_="value",
            axistick_opts=opts.AxisTickOpts(is_show=True),
            splitline_opts=opts.SplitLineOpts(is_show=True),
        ),
        tooltip_opts=opts.TooltipOpts(is_show=False),
    )
    .add_xaxis(data_over_time.index.tolist())
    .add_yaxis(
        series_name="",
        y_axis=data_over_time['Recovered'].tolist(),
        symbol_size=4,
        label_opts=opts.LabelOpts(is_show=False),
        itemstyle_opts=opts.ItemStyleOpts(color="green"),
    )
    .render_notebook()
)
```

```
Out[19]:
```

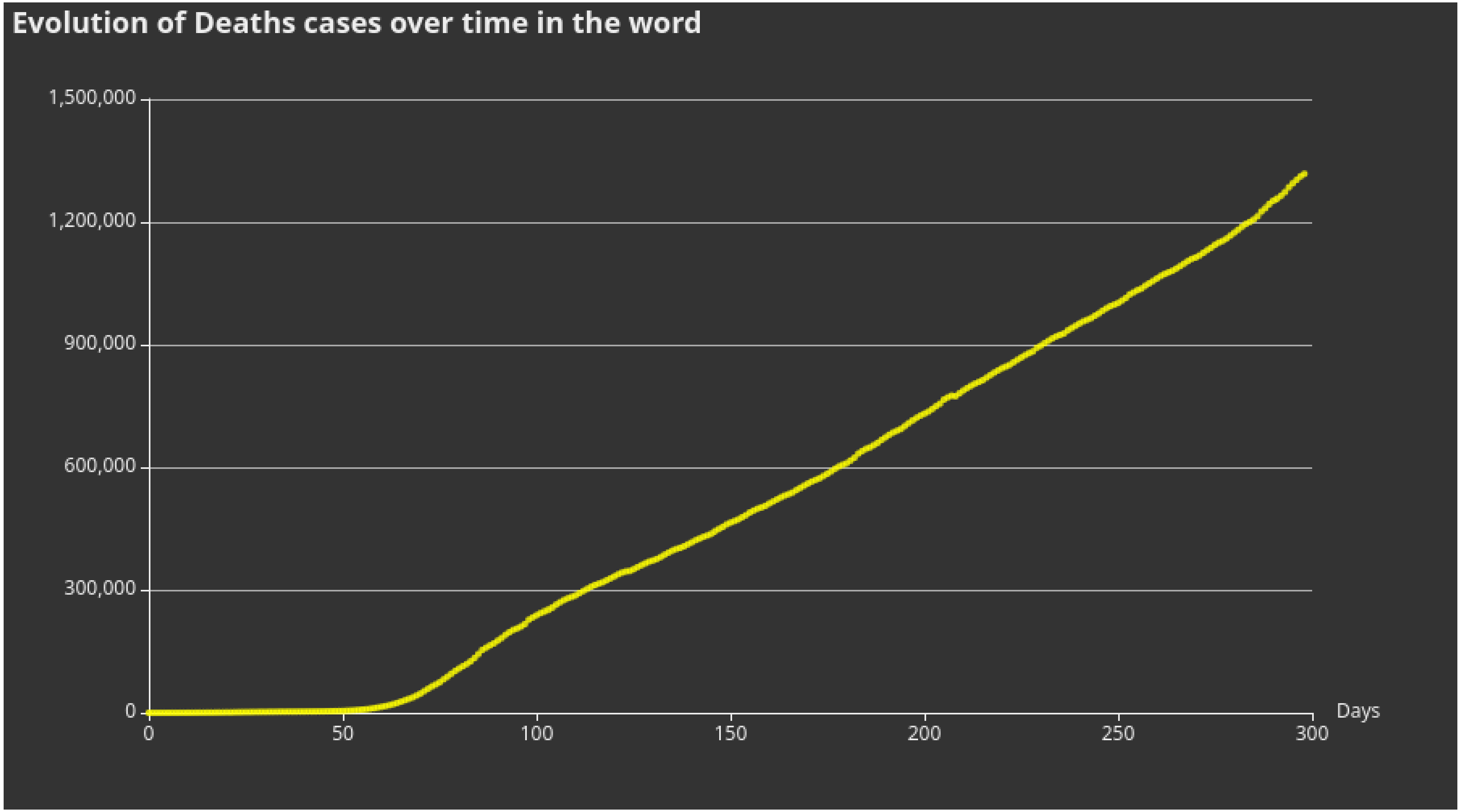


Evolution of Recovered cases over time in the word

In [20]:

```
(
    Scatter(init_opts=opts.InitOpts(theme=ThemeType.DARK))
    .set_global_opts(
        title_opts=opts.TitleOpts(title="Evolution of Deaths cases over time in the word"),
        xaxis_opts=opts.AxisOpts(type_='value', name="Days"),
        yaxis_opts=opts.AxisOpts(
            type_="value",
            axistick_opts=opts.AxisTickOpts(is_show=True),
            splitline_opts=opts.SplitLineOpts(is_show=True),
        ),
        tooltip_opts=opts.TooltipOpts(is_show=False),
    )
    .add_xaxis(data_over_time.index.tolist())
    .add_yaxis(
        series_name="",
        y_axis=data_over_time['Deaths'].tolist(),
        symbol_size=4,
        label_opts=opts.LabelOpts(is_show=False),
        itemstyle_opts=opts.ItemStyleOpts(color="yellow"),
    )
    .render_notebook()
)
```

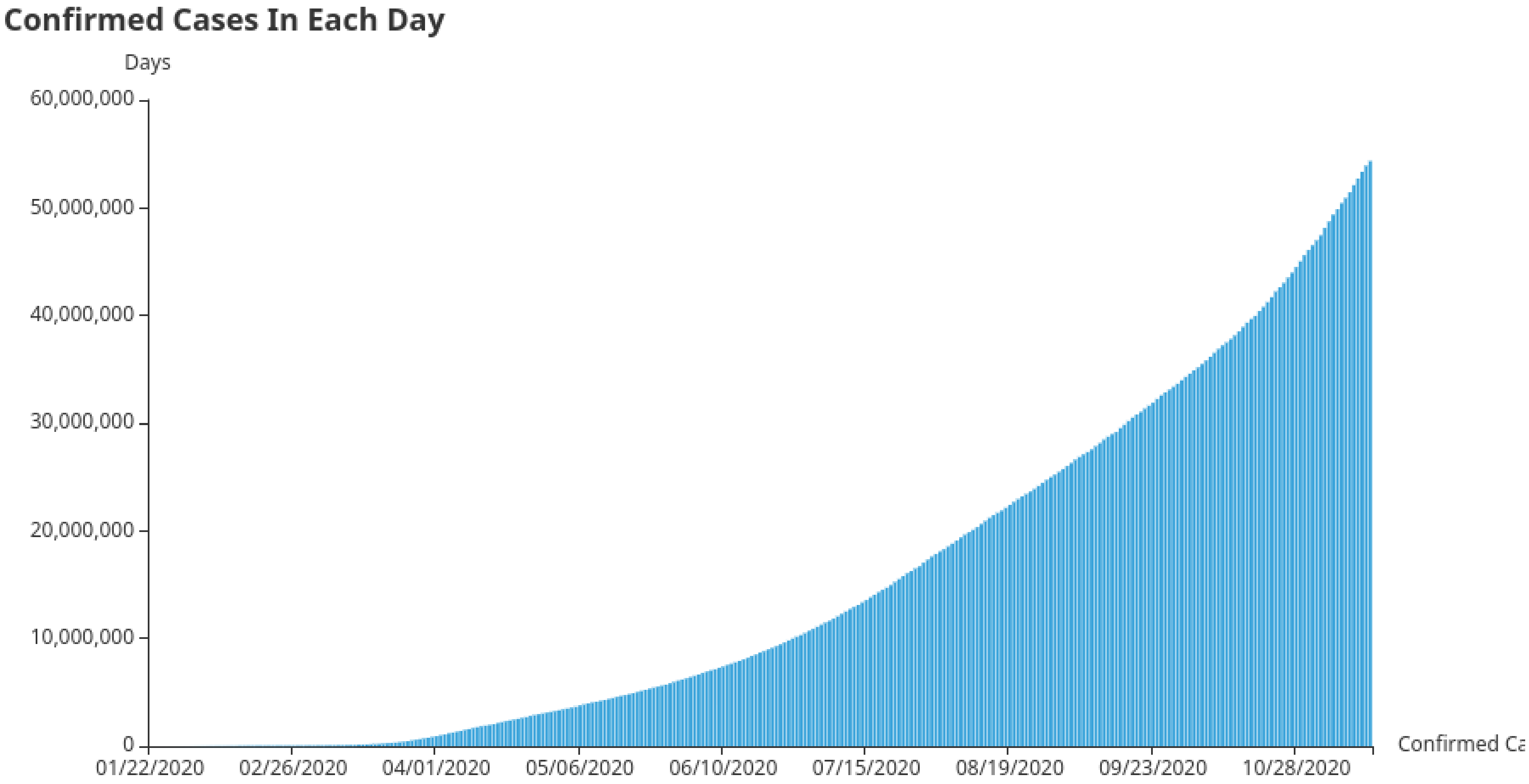
Out[20]:



Evolution of Deaths cases over time in the word
1,500,000
1,200,000
900,000
600,000
300,000
0
0
50
100
150
200
250
300
Days

```
In [21]: (
             Bar(init_opts=opts.InitOpts(theme=ThemeType.LIGHT))
             .set_global_opts(
                 title_opts=opts.TitleOpts(title="Confirmed Cases In Each Day"),
                 xaxis_opts=opts.AxisOpts(type_='category', name="Confirmed Cases"),
                 yaxis_opts=opts.AxisOpts(
                     type_="value", name="Days"
                 ),
                 tooltip_opts=opts.TooltipOpts(is_show=False),
             )
             .add_xaxis(data_over_time['ObservationDate'].values.tolist())
             .add_yaxis("", data_over_time['Confirmed'].values.tolist(), label_opts=opts.LabelOpts(is_show=False),)
             .render_notebook()
         )
```

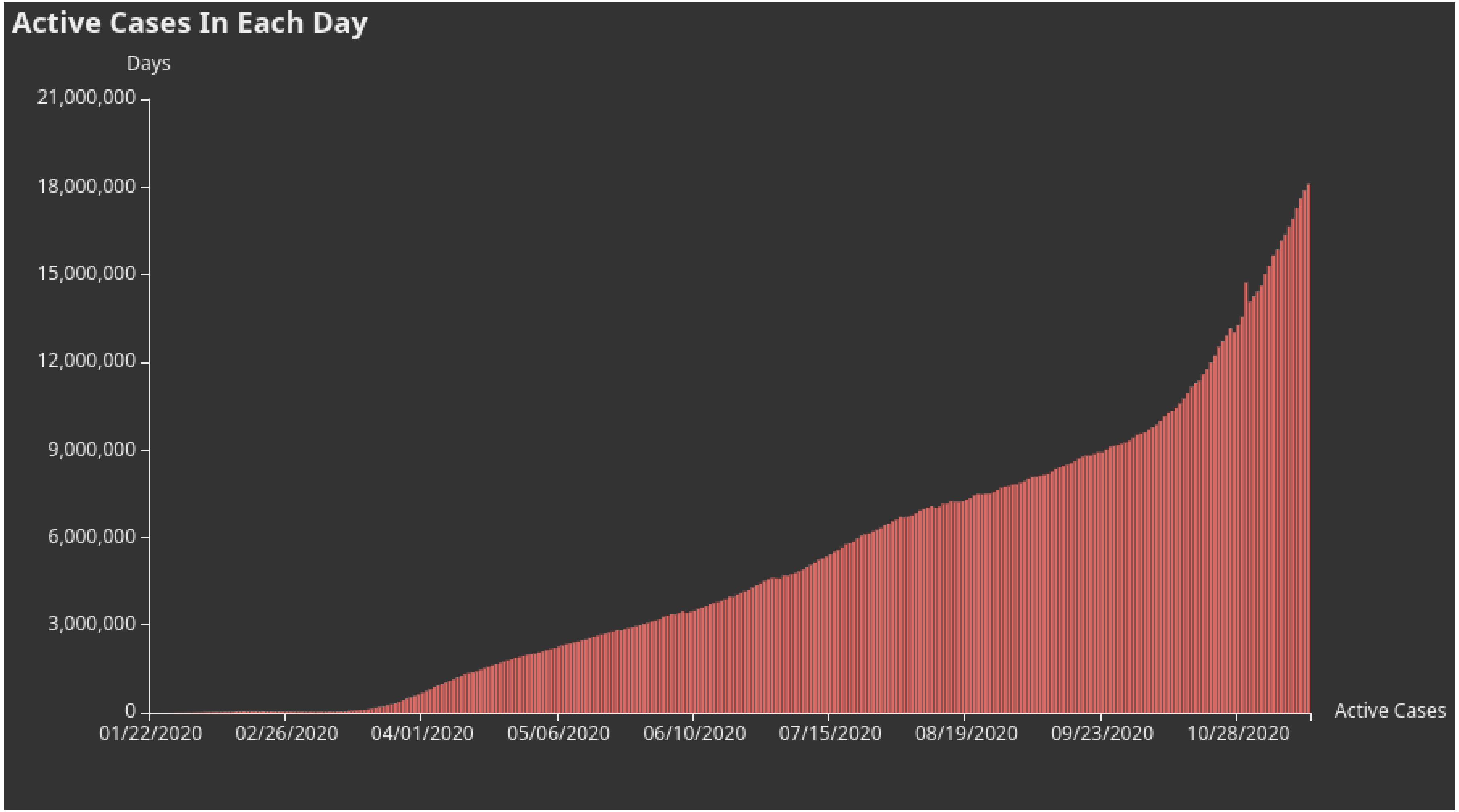
Out[21]:



In [22]:

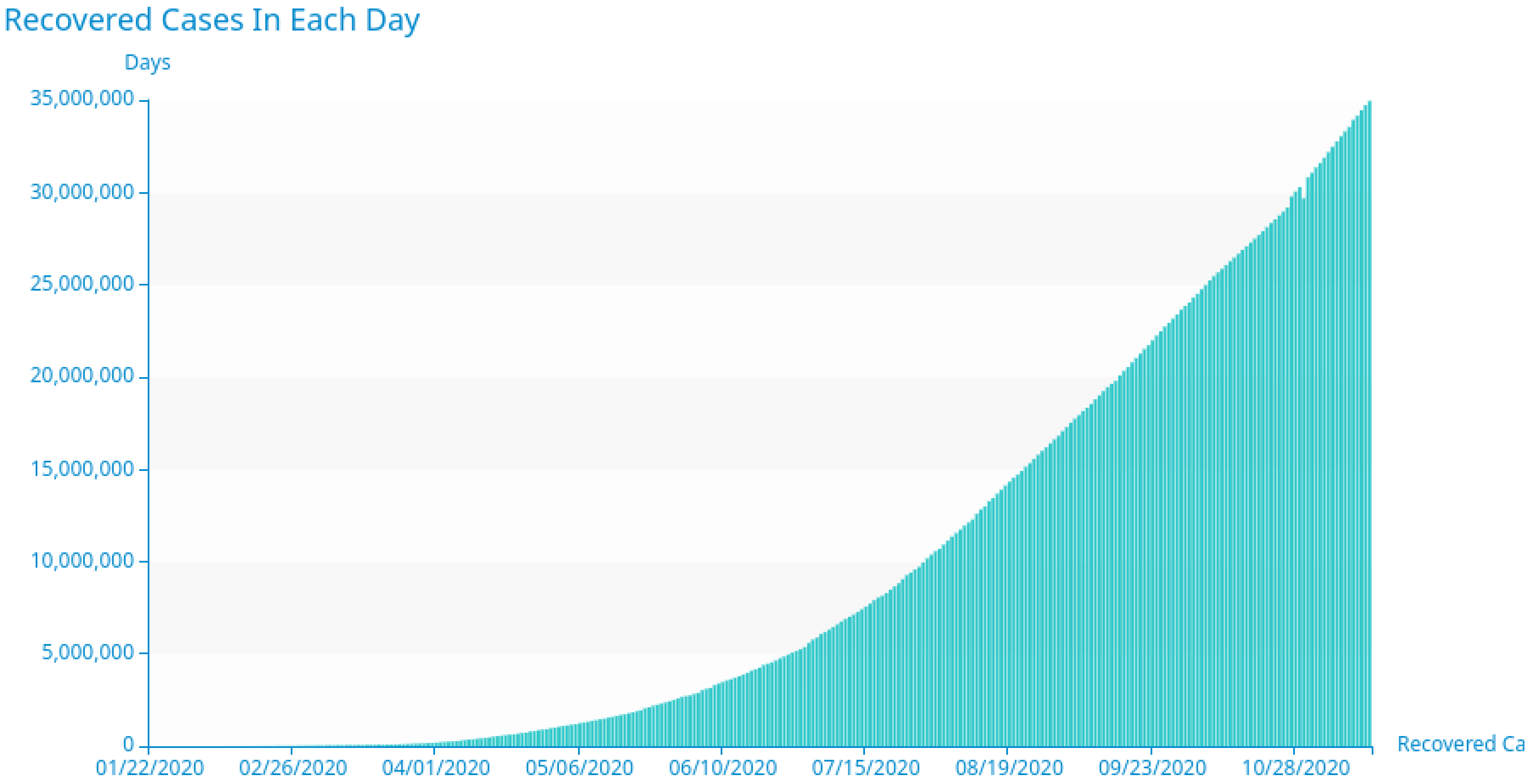
```
(
    Bar(init_opts=opts.InitOpts(theme=ThemeType.DARK))
    .set_global_opts(
        title_opts=opts.TitleOpts(title="Active Cases In Each Day"),
        xaxis_opts=opts.AxisOpts(type_='category', name="Active Cases"),
        yaxis_opts=opts.AxisOpts(
            type_="value", name="Days"
        ),
        tooltip_opts=opts.TooltipOpts(is_show=False),
    )
    .add_xaxis(data_over_time['ObservationDate'].values.tolist())
    .add_yaxis("", data_over_time['Active_case'].values.tolist(), label_opts=opts.LabelOpts(is_show=False),)
    .render_notebook()
)
```

Out[22]:



```
In [23]: (
             Bar(init_opts=opts.InitOpts(theme=ThemeType.MACARONS))
             .set_global_opts(
                 title_opts=opts.TitleOpts(title="Recovered Cases In Each Day"),
                 xaxis_opts=opts.AxisOpts(type_='category', name="Recovered Cases"),
                 yaxis_opts=opts.AxisOpts(
                     type_="value", name="Days"
                 ),
                 tooltip_opts=opts.TooltipOpts(is_show=False),
             )
             .add_xaxis(data_over_time['ObservationDate'].values.tolist())
             .add_yaxis("", data_over_time['Recovered'].values.tolist(), label_opts=opts.LabelOpts(is_show=False),)
             .render_notebook()
         )
```

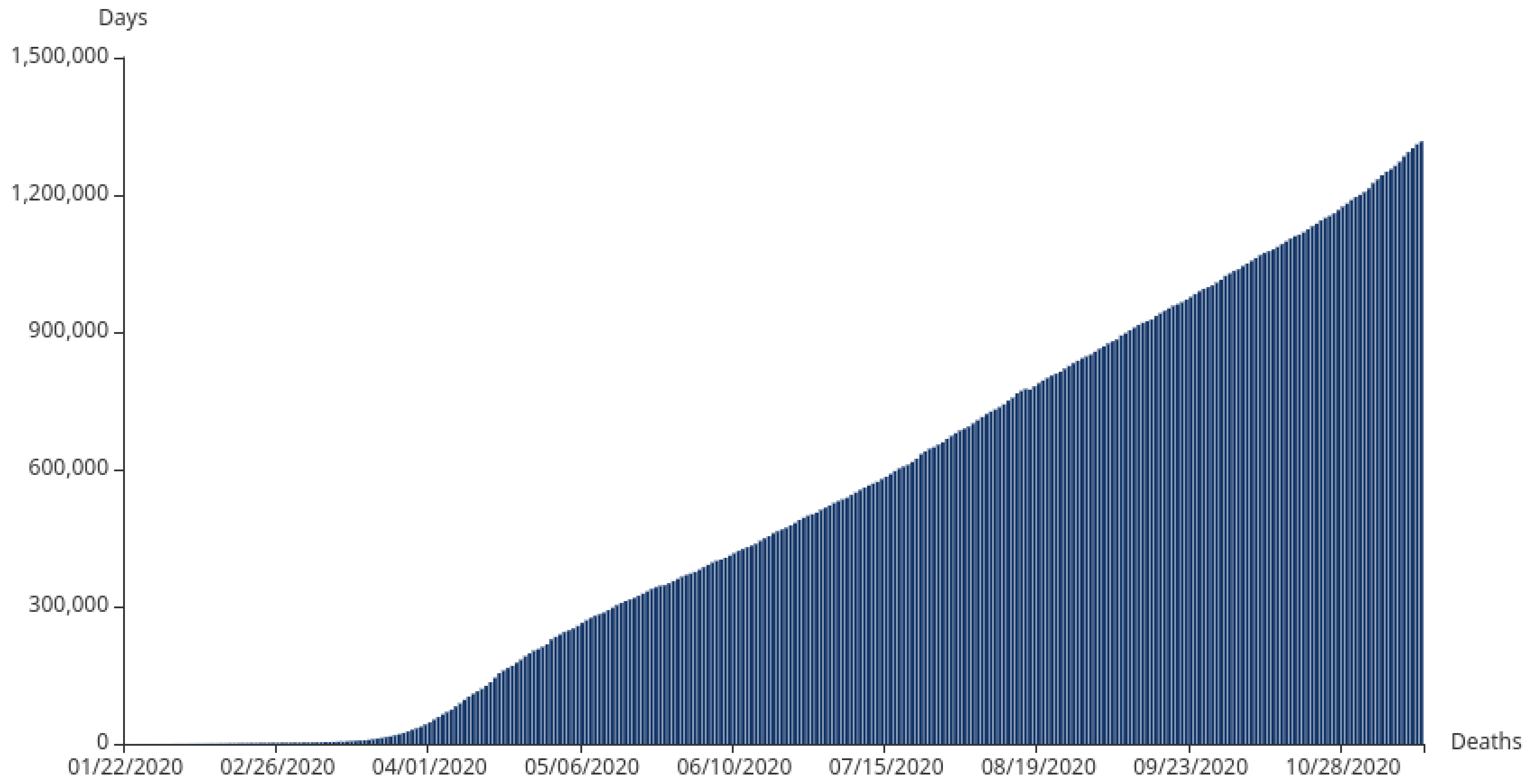
Out[23]:



```
In [24]: (
             Bar()
             .set_global_opts(
                 title_opts=opts.TitleOpts(title="Deaths In Each Day"),
                 xaxis_opts=opts.AxisOpts(type_='category', name="Deaths"),
                 yaxis_opts=opts.AxisOpts(
                     type_="value", name="Days"
                 ),
                 tooltip_opts=opts.TooltipOpts(is_show=False),
             )
             .add_xaxis(data_over_time['ObservationDate'].values.tolist())
             .add_yaxis("", data_over_time['Deaths'].values.tolist(),
                        label_opts=opts.LabelOpts(is_show=False),itemstyle_opts=opts.ItemStyleOpts(color="rgb(13,48,100)"))
             .render_notebook()
         )
```

Out[24]:

Deaths In Each Day



Confirmed cases in each Country

In [25]:

```
Data_per_country = Data.groupby(["Country/Region"])["Confirmed","Active_case","Recovered","Deaths"].sum().reset_index().sort_values("Confirmed",ascending=False).reset_index(drop=True)
Data_per_country.head()
```

Out[25]:

	Country/Region	Confirmed	Active_case	Recovered	Deaths
0	US	11036935	6615837	4174884	246214
1	India	8845127	465478	8249579	130070
2	Brazil	5863093	417843	5279452	165798
3	France	1915713	1733352	139760	42601
4	Russia	1910149	447699	1429565	32885

In [26]:

```
table = Table()
headers = ['Country','Confirmed Cases']
table.add(headers, [[country,confirmed] for country,confirmed in zip(Data_per_country['Country/Region'].tolist(),Data_per_country['Confirmed'].tolist())])
table.set_global_opts(
    title_opts=ComponentTitleOpts(title="Confirmed Cases In Each Country")
)
table.render_notebook()
```

Out[26]:

Confirmed Cases In Each Country

Country	Confirmed Cases
US	11036935
India	8845127
Brazil	5863093
France	1915713
Russia	1910149
Spain	1458591
UK	1372884
Argentina	1310491
Colombia	1198746
Italy	1178529
Mexico	1006522
Peru	934899
Germany	802946
Iran	762068
South Africa	751024
Poland	712972
Ukraine	551533
Belgium	535939
Chile	531273
Iraq	519152

Country	Confirmed Cases
Indonesia	467113
Czech Republic	460116
Netherlands	454961
Bangladesh	432333
Turkey	414278
Philippines	407838
Romania	360281
Pakistan	359032
Saudi Arabia	353255
Israel	323741
Canada	299422
Morocco	293177
Switzerland	257135
Portugal	217301
Nepal	209776
Austria	203956
Ecuador	180295
Sweden	177355
United Arab Emirates	150345
Panama	146653
Jordan	143678

Country	Confirmed Cases
Bolivia	143246
Hungary	140961
Kuwait	136840
Qatar	135785
Dominican Republic	133724
Costa Rica	123223
Kazakhstan	121051
Oman	120389
Japan	118611
Armenia	117337
Guatemala	114885
Belarus	114185
Egypt	110767
Lebanon	105430
Honduras	103102
Ethiopia	102720
Bulgaria	98251
Venezuela	97352
Moldova	89279
Slovakia	86767
China	86346

Country	Confirmed Cases
Bahrain	84703
Serbia	84568
Croatia	84206
Tunisia	80404
Georgia	79678
Azerbaijan	75688
Greece	74205
Libya	73602
Bosnia and Herzegovina	71956
Paraguay	71574
Kenya	70245
Uzbekistan	70243
Burma	68994
Ireland	67903
Algeria	67679
Kyrgyzstan	66983
Nigeria	65148
West Bank and Gaza	63031
Denmark	62652
Singapore	58119
Slovenia	55042

Country	Confirmed Cases
Ghana	50123
Malaysia	47417
North Macedonia	47050
Afghanistan	43240
El Salvador	36358
Lithuania	34758
Kosovo	29117
South Korea	28769
Norway	28434
Albania	27830
Australia	27749
Montenegro	27177
Luxembourg	25931
Cameroon	22692
Ivory Coast	20976
Finland	19315
Madagascar	17310
Sri Lanka	17287
Zambia	17123
Uganda	16020
Senegal	15793

Country	Confirmed Cases
Sudan	14626
Mozambique	14448
Namibia	13508
Angola	13451
Guinea	12611
Maldives	12204
Congo (Kinshasa)	11760
Tajikistan	11610
Latvia	10547
Jamaica	9884
Cabo Verde	9822
Haiti	9168
Gabon	9062
Zimbabwe	8829
Botswana	8225
Malta	8034
Mauritania	7952
Estonia	7637
Cuba	7590
Bahamas	7186
Cyprus	7178

Country	Confirmed Cases
Syria	6684
Eswatini	6095
Trinidad and Tobago	6080
Malawi	5965
Andorra	5872
Nicaragua	5661
Djibouti	5649
Congo (Brazzaville)	5515
Hong Kong	5458
Rwanda	5455
Suriname	5274
Iceland	5189
Equatorial Guinea	5104
Central African Republic	4900
Belize	4861
Guyana	4823
Somalia	4301
Uruguay	4030
Mali	3900
Thailand	3875
Gambia	3702

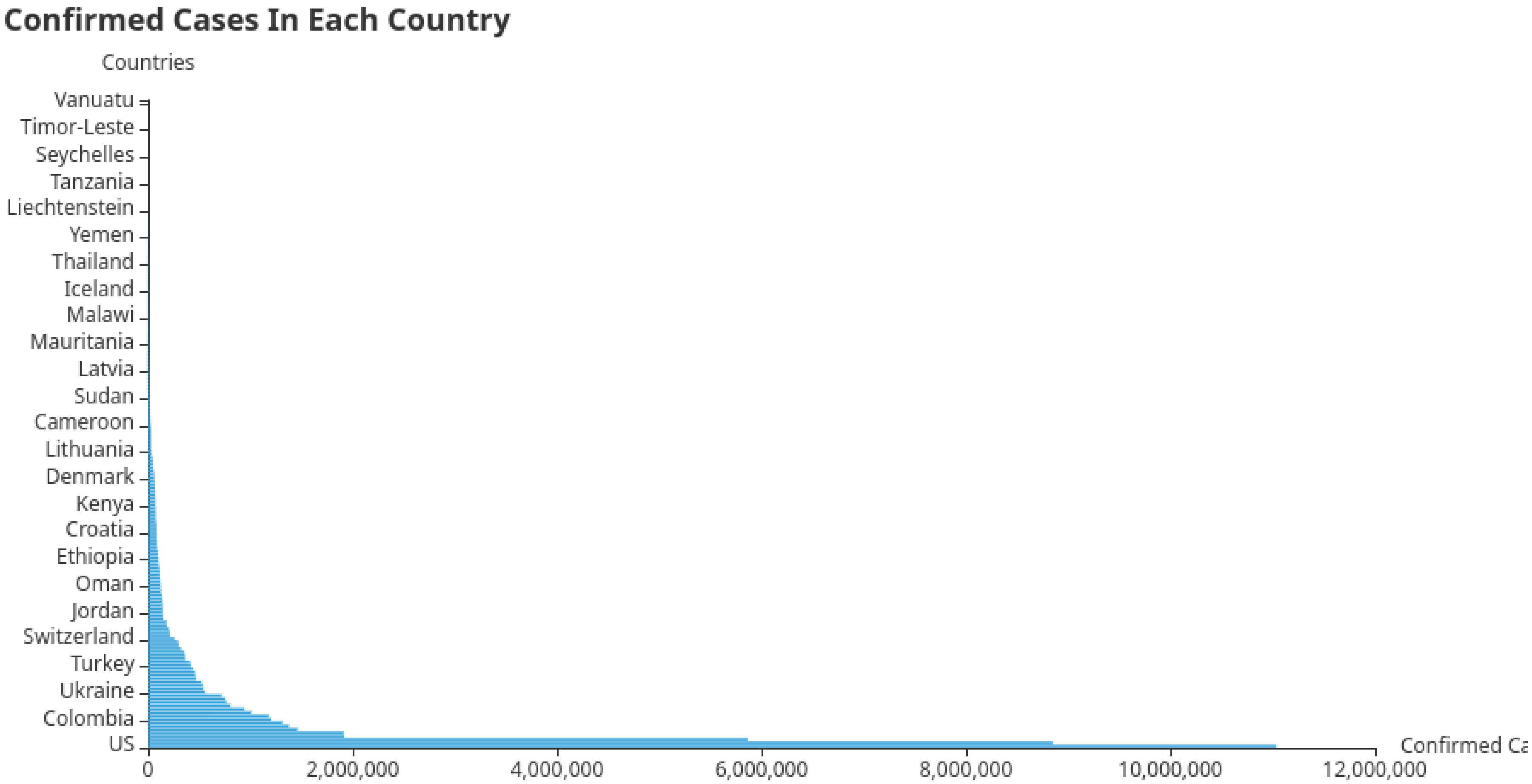
Country	Confirmed Cases
South Sudan	3003
Benin	2844
Togo	2675
Burkina Faso	2641
Guinea-Bissau	2419
Sierra Leone	2391
Yemen	2072
Lesotho	2041
New Zealand	2001
Chad	1597
Liberia	1507
Niger	1307
Vietnam	1281
San Marino	1253
Liechtenstein	985
Sao Tome and Principe	965
Diamond Princess	712
Burundi	628
Papua New Guinea	602
Taiwan	602
Comoros	579

Country	Confirmed Cases
Monaco	548
Tanzania	509
Eritrea	493
Mauritius	478
Mongolia	431
Bhutan	375
Cambodia	302
Barbados	250
Saint Lucia	162
Seychelles	160
Brunei	148
Antigua and Barbuda	134
Saint Vincent and the Grenadines	78
Dominica	68
Macau	46
Fiji	35
Grenada	32
Timor-Leste	30
Holy See	27
Laos	24
Saint Kitts and Nevis	19

Country	Confirmed Cases
Solomon Islands	16
Western Sahara	10
MS Zaandam	9
Marshall Islands	1
Vanuatu	1

```
In [27]: (
    Bar(init_opts=opts.InitOpts(theme=ThemeType.LIGHT))
    .add_xaxis(Data_per_country['Country/Region'].values.tolist())
    .add_yaxis("", Data_per_country['Confirmed'].values.tolist(), label_opts=opts.LabelOpts(is_show=False),)
    .reversal_axis()
    .set_global_opts(
        title_opts=opts.TitleOpts(title="Confirmed Cases In Each Country"),
        tooltip_opts=opts.TooltipOpts(is_show=False),
        xaxis_opts=opts.AxisOpts(type_='value', name="Confirmed Cases"),
        yaxis_opts=opts.AxisOpts(type_="category", name="Countries"),
    )
    .render_notebook()
)
```

Out[27]:

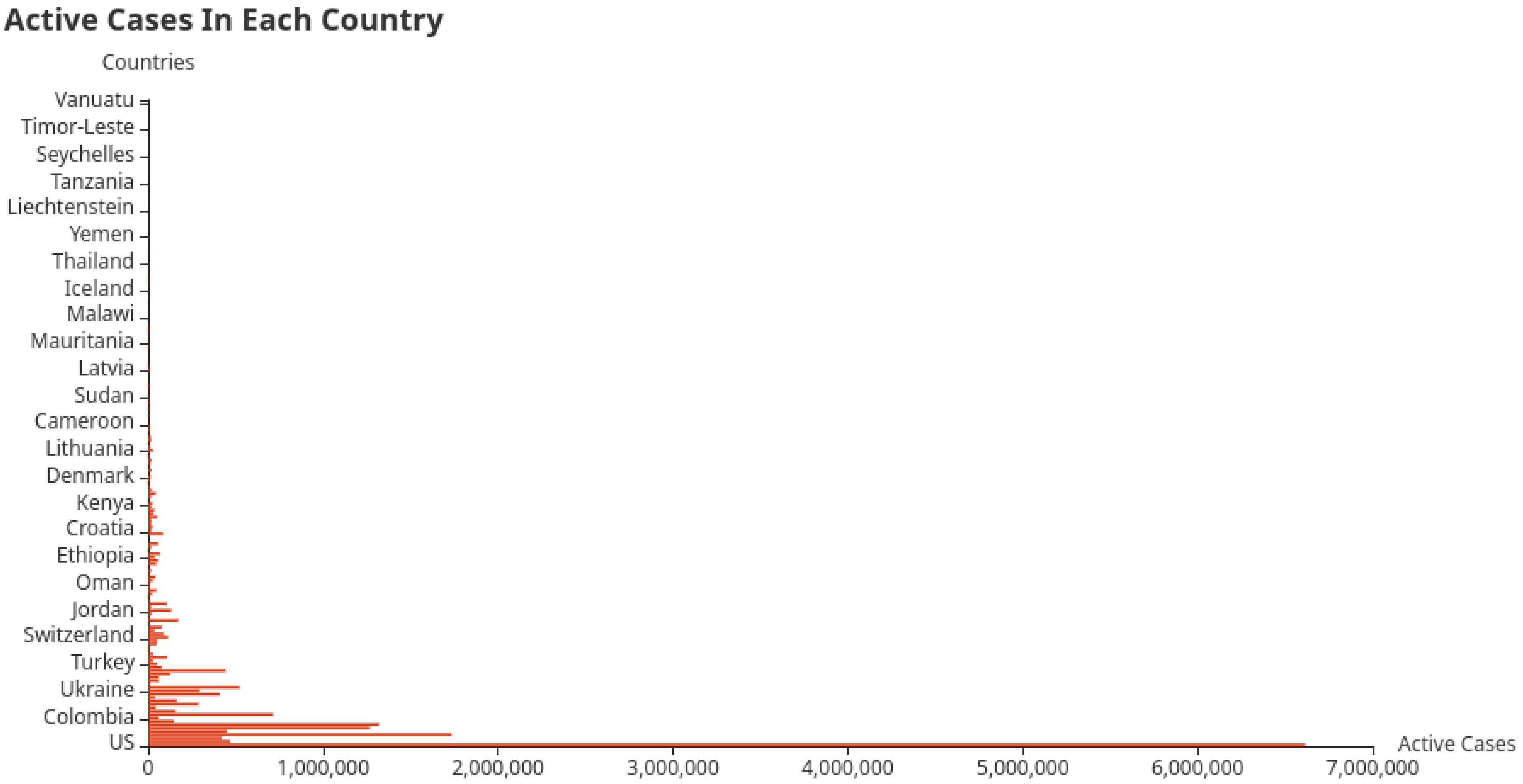


Confirmed Cases In Each Country

In [28]:

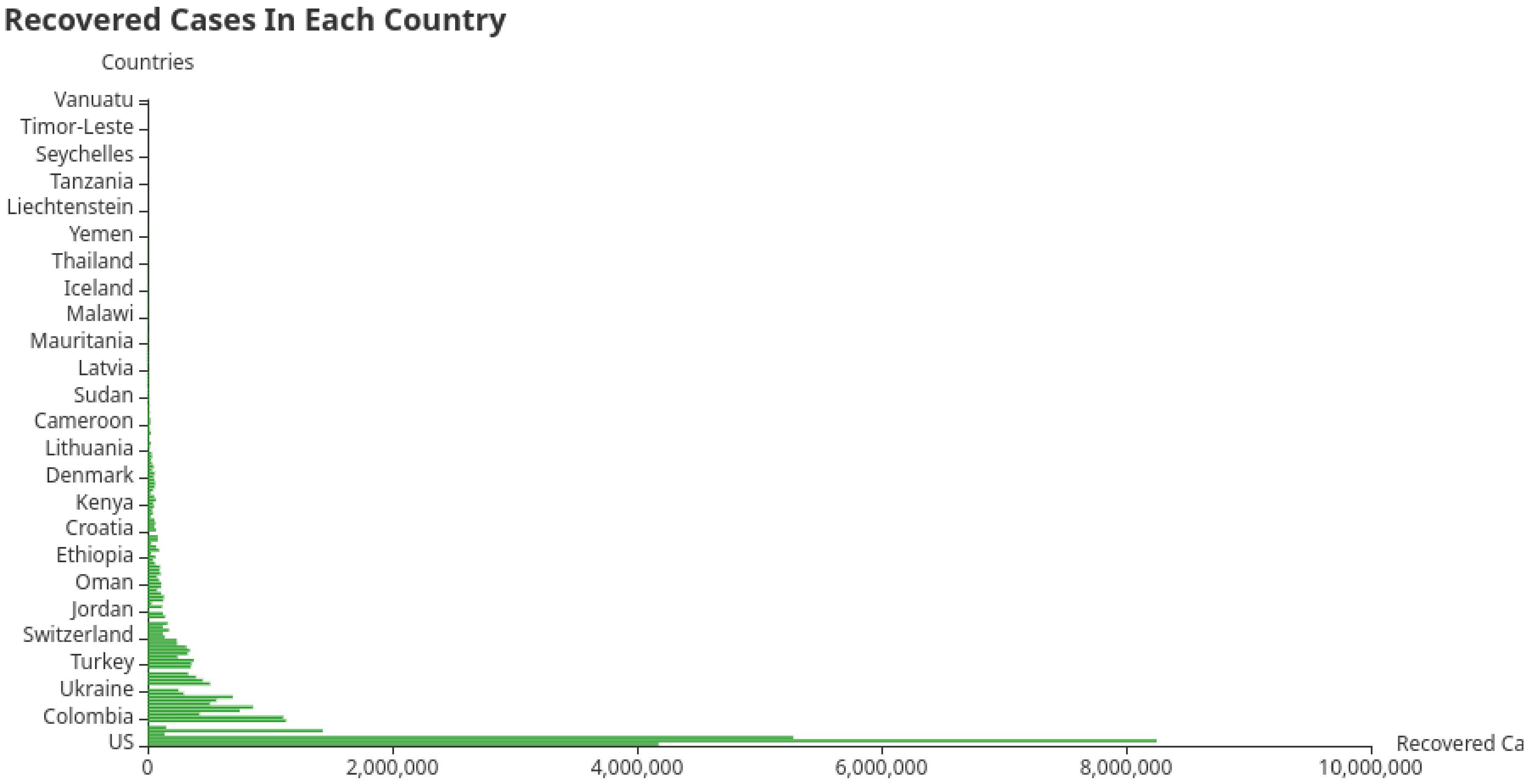
```
(
    Bar(init_opts=opts.InitOpts(theme=ThemeType.LIGHT))
    .add_xaxis(Data_per_country['Country/Region'].values.tolist())
    .add_yaxis("", Data_per_country['Active_case'].values.tolist(),
               label_opts=opts.LabelOpts(is_show=False),
               itemstyle_opts=opts.ItemStyleOpts(color="#DC3912"))
    .reversal_axis()
    .set_global_opts(
        title_opts=opts.TitleOpts(title="Active Cases In Each Country"),
        tooltip_opts=opts.TooltipOpts(is_show=False),
        xaxis_opts=opts.AxisOpts(type_='value', name="Active Cases"),
        yaxis_opts=opts.AxisOpts(type_="category", name="Countries"),
    )
    .render_notebook()
)
```

Out[28]:



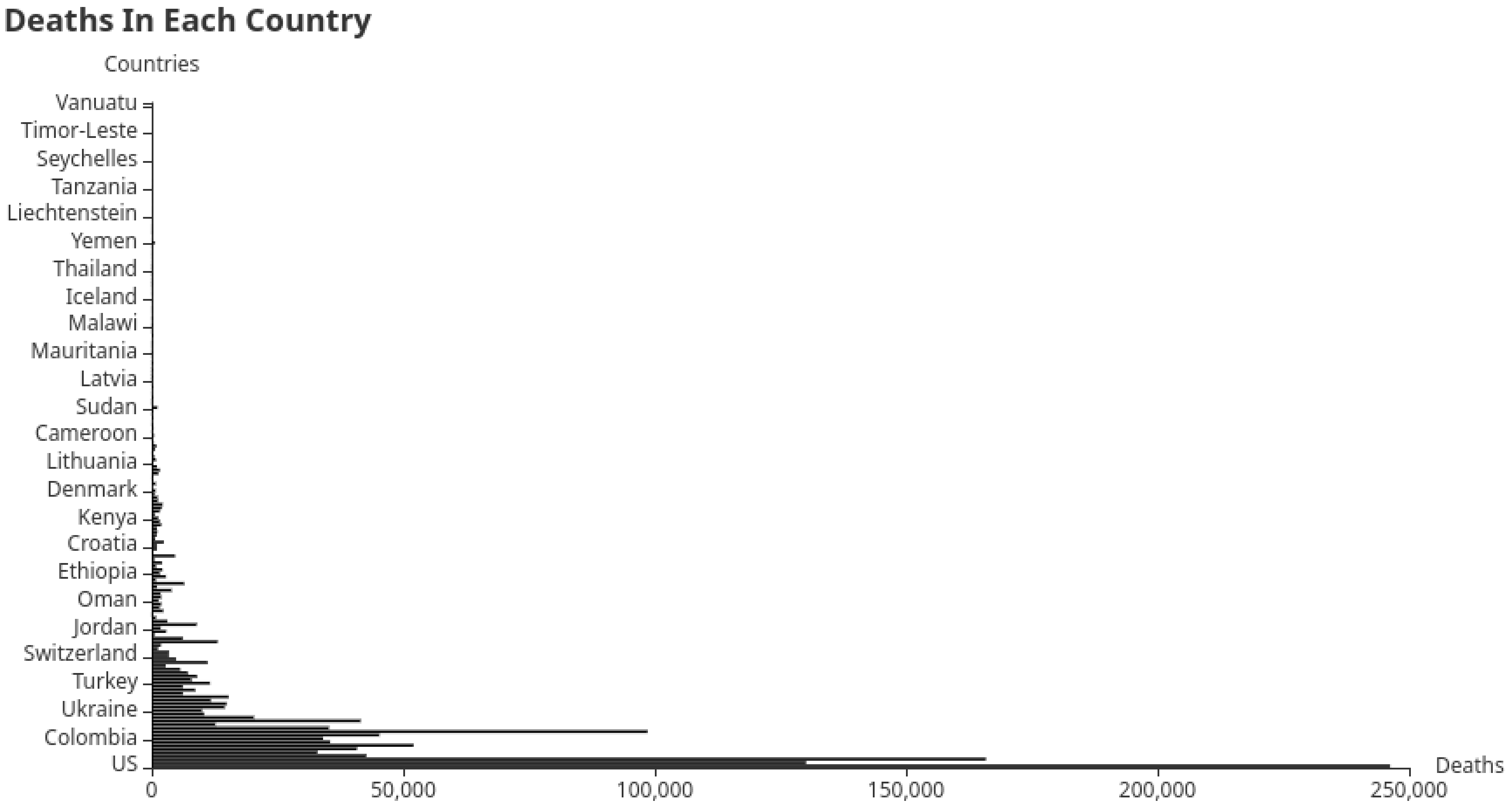
```
In [29]: (
             Bar(init_opts=opts.InitOpts(theme=ThemeType.LIGHT))
             .add_xaxis(Data_per_country['Country/Region'].values.tolist())
             .add_yaxis("", Data_per_country['Recovered'].values.tolist(),
                        label_opts=opts.LabelOpts(is_show=False),
                        itemstyle_opts=opts.ItemStyleOpts(color="#2CA02C"))
             .reversal_axis()
             .set_global_opts(
                 title_opts=opts.TitleOpts(title="Recovered Cases In Each Country"),
                 tooltip_opts=opts.TooltipOpts(is_show=False),
                 xaxis_opts=opts.AxisOpts(type_='value', name="Recovered Cases"),
                 yaxis_opts=opts.AxisOpts(type_="category", name="Countries"),
             )
             .render_notebook()
         )
```

Out[29]:



```
In [30]: (
             Bar(init_opts=opts.InitOpts(theme=ThemeType.LIGHT))
             .add_xaxis(Data_per_country['Country/Region'].values.tolist())
             .add_yaxis("", Data_per_country['Deaths'].values.tolist(),
                        label_opts=opts.LabelOpts(is_show=False),
                        itemstyle_opts=opts.ItemStyleOpts(color="black"))
             .reversal_axis()
             .set_global_opts(
                 title_opts=opts.TitleOpts(title="Deaths In Each Country"),
                 tooltip_opts=opts.TooltipOpts(is_show=False),
                 xaxis_opts=opts.AxisOpts(type_='value', name="Deaths"),
                 yaxis_opts=opts.AxisOpts(type_="category", name="Countries"),
             )
             .render_notebook()
         )
```

Out[30]:



In [31]:

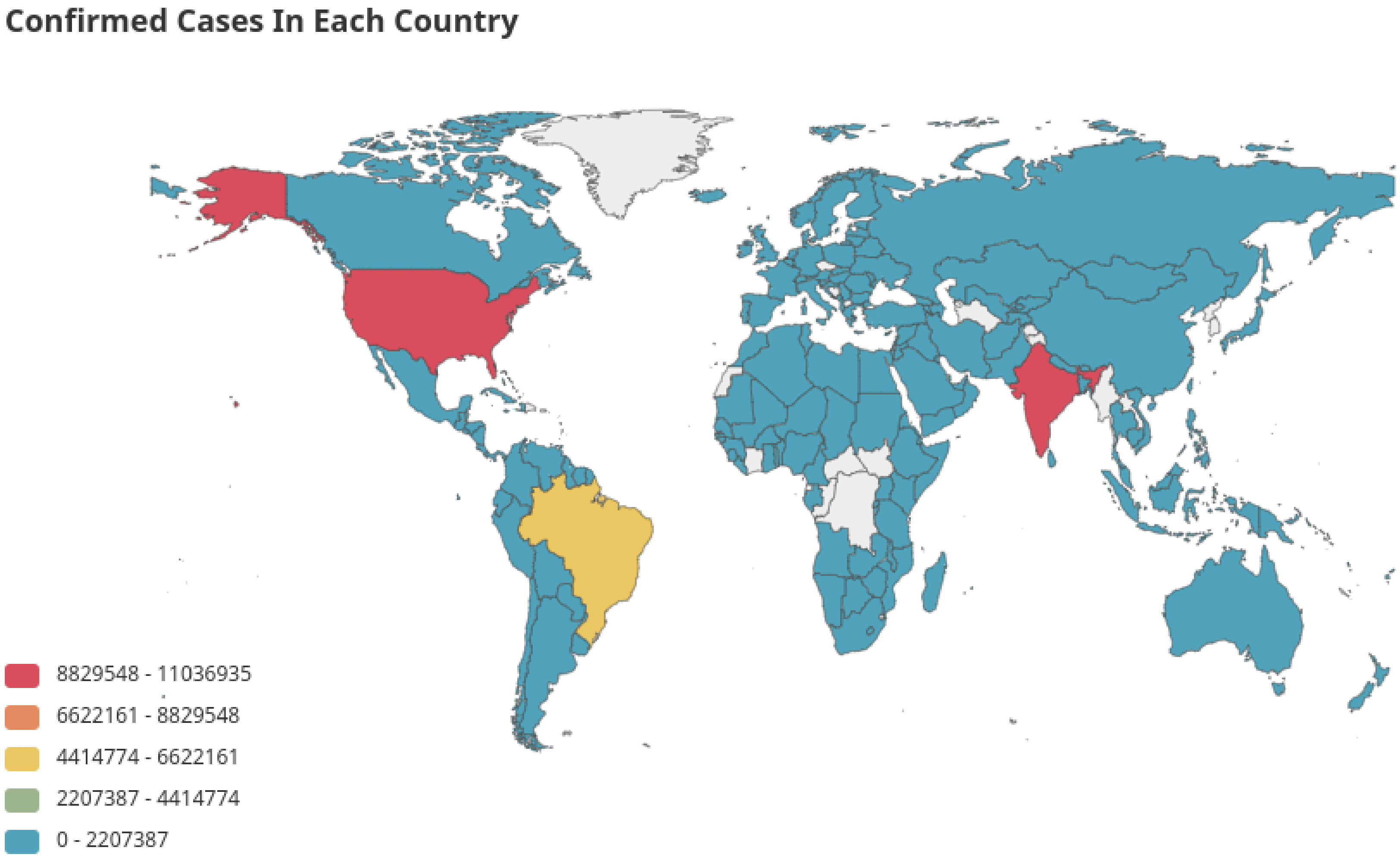
```
#pyechart map
Data_per_country['Country/Region'] = Data_per_country['Country/Region'].replace('US', 'United States')
Data_per_country['Country/Region'] = Data_per_country['Country/Region'].replace('UK', 'United Kingdom')
Data_per_country.head()
```

Out[31]:

	Country/Region	Confirmed	Active_case	Recovered	Deaths
0	United States	11036935	6615837	4174884	246214
1	India	8845127	465478	8249579	130070
2	Brazil	5863093	417843	5279452	165798
3	France	1915713	1733352	139760	42601
4	Russia	1910149	447699	1429565	32885

```
In [32]: (
             Map()
             .add("", [list(z) for z in zip(Data_per_country['Country/Region'].values.tolist(),
                                            Data_per_country['Confirmed'].values.tolist())], "world",
                 label_opts=None,is_map_symbol_show=False)
             .set_global_opts(title_opts=opts.TitleOpts(title="Confirmed Cases In Each Country"),
                              visualmap_opts=opts.VisualMapOpts(max_=int(Data_per_country['Confirmed'].max()), is_piecewise=T
         rue))
             .render_notebook()
         )
```

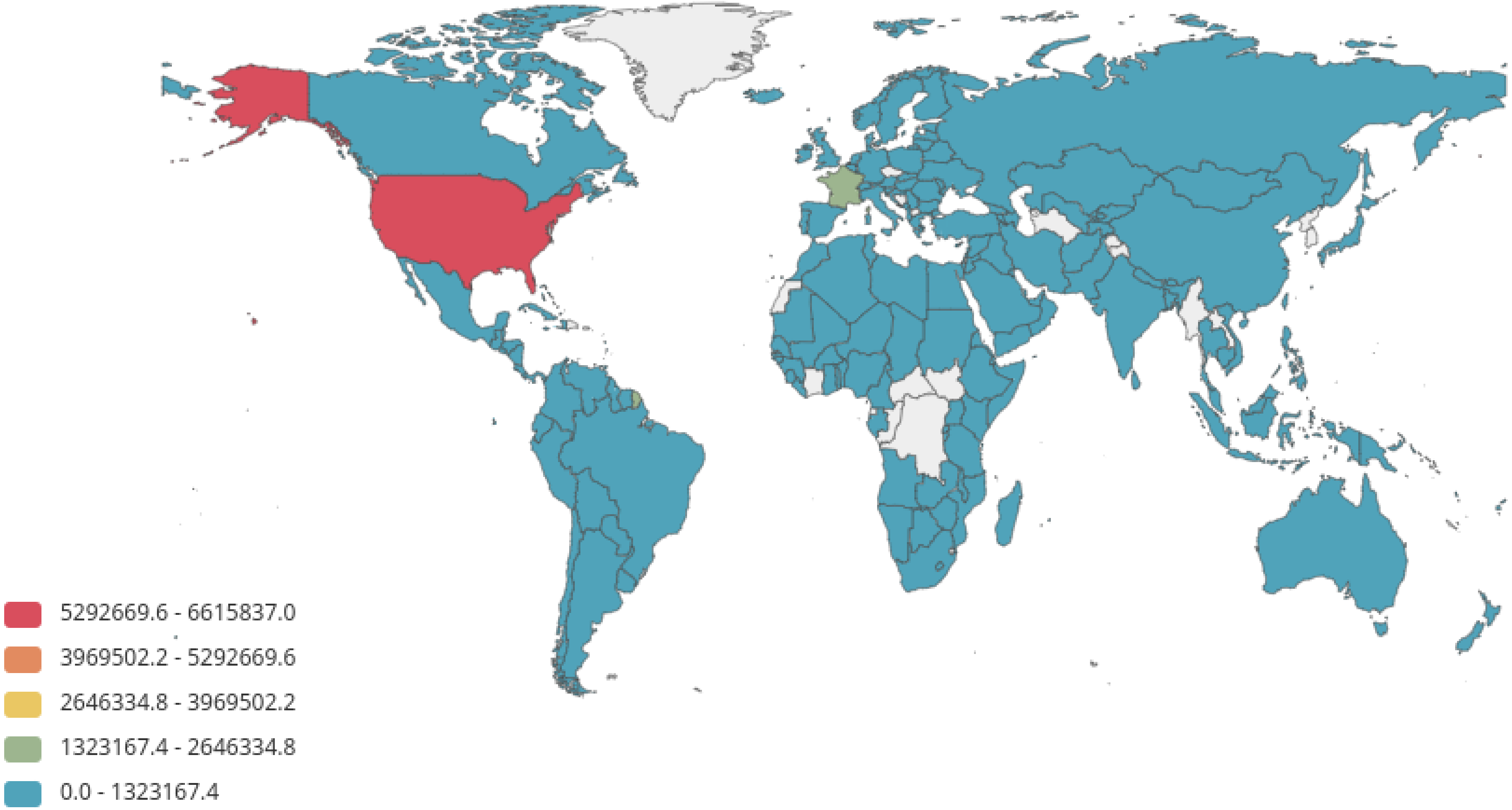
Out[32]:



In [33]:
```
(
    Map()
    .add("", [list(z) for z in zip(Data_per_country['Country/Region'].values.tolist(),
                                  Data_per_country['Active_case'].values.tolist())], "world",
        label_opts=None,is_map_symbol_show=False)
    .set_global_opts(title_opts=opts.TitleOpts(title="Active Cases In Each Country"),
                     visualmap_opts=opts.VisualMapOpts(max_=int(Data_per_country['Active_case'].max()), is_piecewise
=True))
    .render_notebook()
)
```

Out[33]:

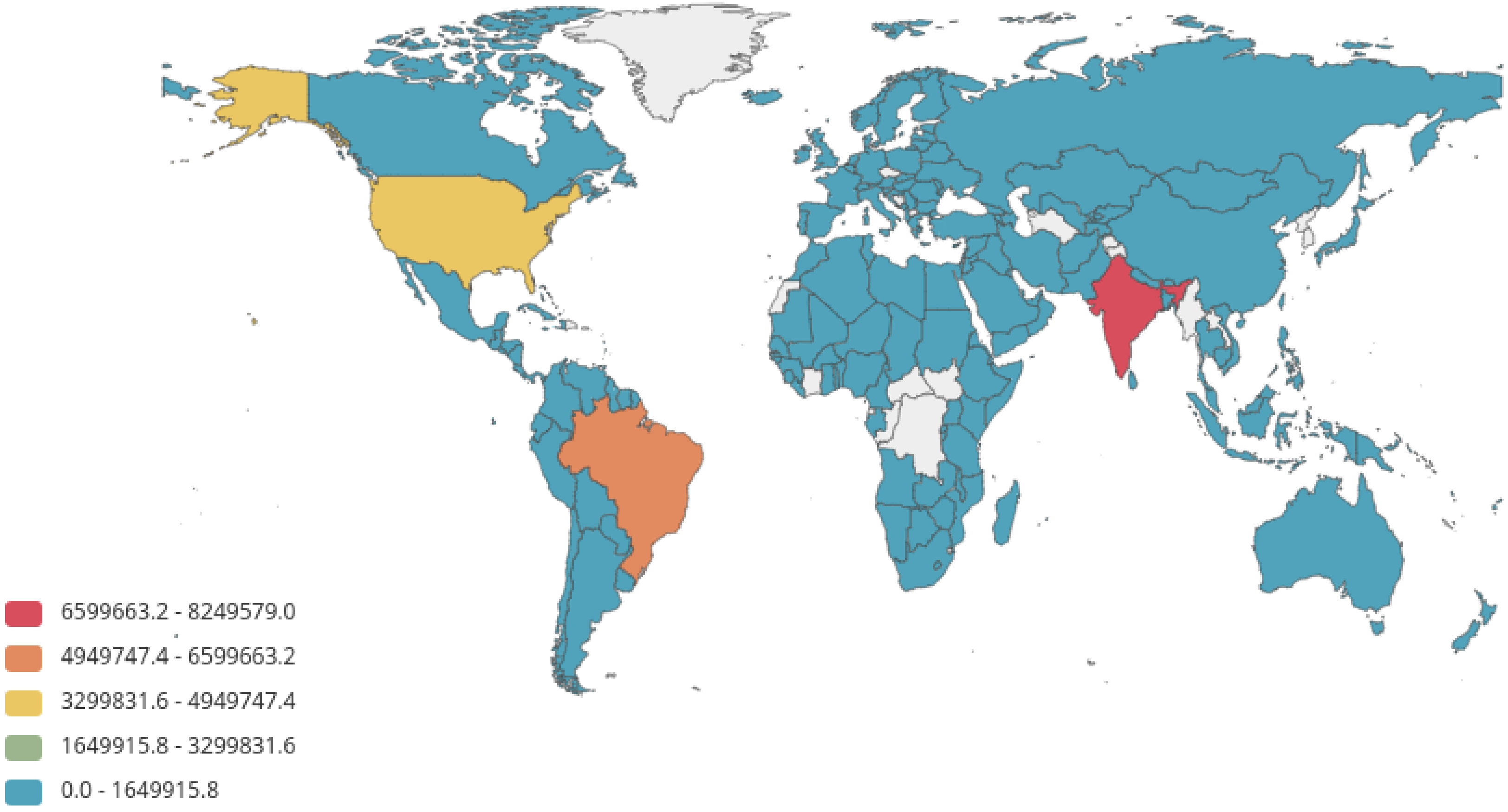
Active Cases In Each Country



```
In [34]: (
             Map()
             .add("", [list(z) for z in zip(Data_per_country['Country/Region'].values.tolist(),
                                          Data_per_country['Recovered'].values.tolist())], "world",
                 label_opts=None,is_map_symbol_show=False)
             .set_global_opts(title_opts=opts.TitleOpts(title="Recovered Cases In Each Country"),
                              visualmap_opts=opts.VisualMapOpts(max_=int(Data_per_country['Recovered'].max()), is_piecewise=T
         rue))
             .render_notebook()
         )
```

Out[34]:

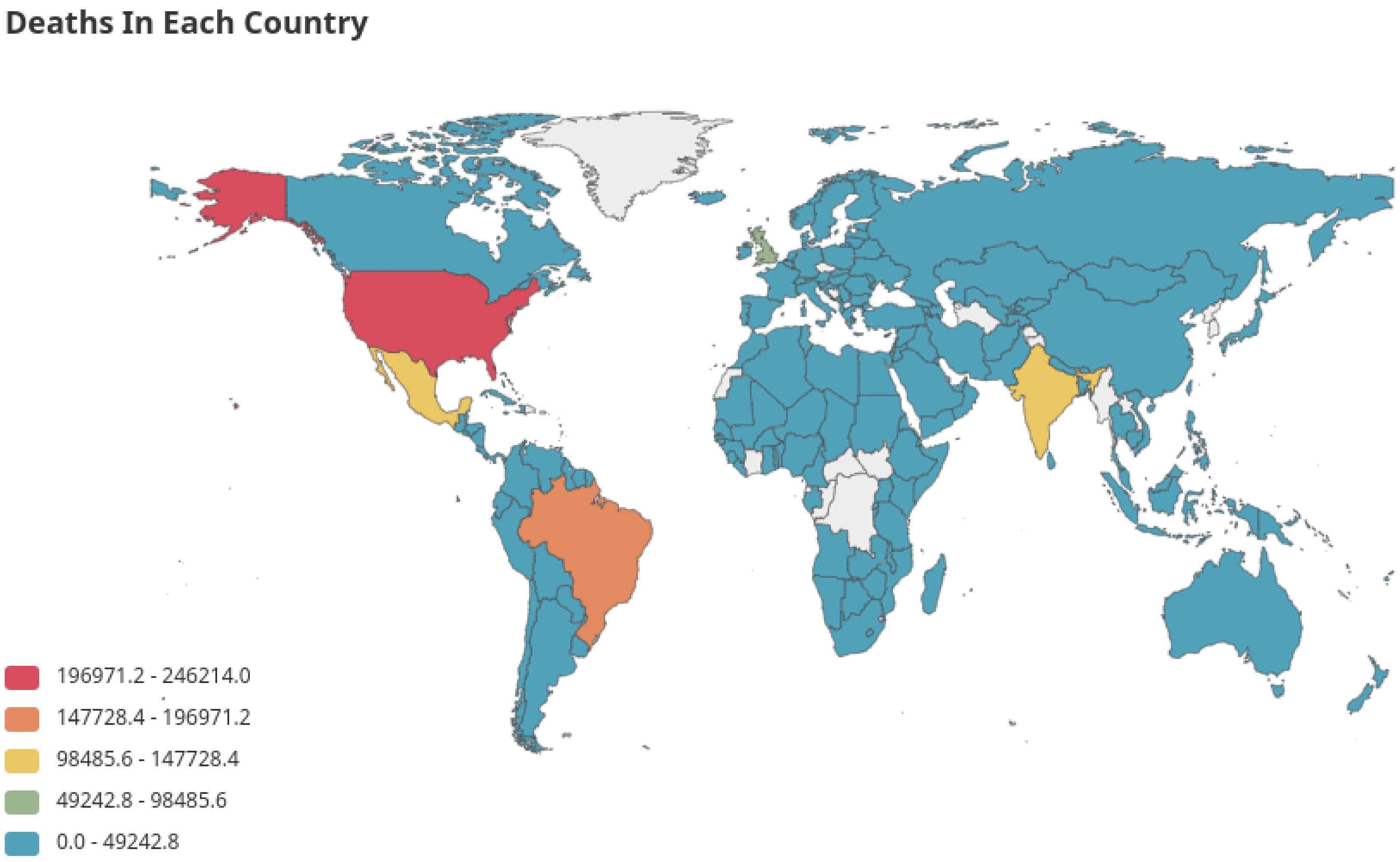
Recovered Cases In Each Country



In [35]:

```
(
    Map()
    .add("", [list(z) for z in zip(Data_per_country['Country/Region'].values.tolist(),
                                   Data_per_country['Deaths'].values.tolist())], "world",
        label_opts=None,is_map_symbol_show=False)
    .set_global_opts(title_opts=opts.TitleOpts(title="Deaths In Each Country"),
                     visualmap_opts=opts.VisualMapOpts(max_=int(Data_per_country['Deaths'].max()), is_piecewise=True
))
    .render_notebook()
)
```

Out[35]:



Evolution of coronavirus over time.

In [36]:

```
data_per_country = data.groupby(["Country/Region","ObservationDate"])[["Confirmed","Active_case","Recovered","Deaths"]].sum().reset_index().sort_values("ObservationDate",ascending=True).reset_index(drop=True)
data_per_country.head()
```

Out[36]:

	Country/Region	ObservationDate	Confirmed	Active_case	Recovered	Deaths
0	Japan	01/22/2020	2	2	0	0
1	US	01/22/2020	1	1	0	0
2	Macau	01/22/2020	1	1	0	0
3	China	01/22/2020	547	502	28	17
4	Hong Kong	01/22/2020	0	0	0	0

In [37]:

```
#pyechart map
data_per_country['Country/Region'] = data_per_country['Country/Region'].replace('US', 'United States')
data_per_country['Country/Region'] = data_per_country['Country/Region'].replace('UK', 'United Kingdom')
data_per_country.head()
```

Out[37]:

	Country/Region	ObservationDate	Confirmed	Active_case	Recovered	Deaths
0	Japan	01/22/2020	2	2	0	0
1	United States	01/22/2020	1	1	0	0
2	Macau	01/22/2020	1	1	0	0
3	China	01/22/2020	547	502	28	17
4	Hong Kong	01/22/2020	0	0	0	0

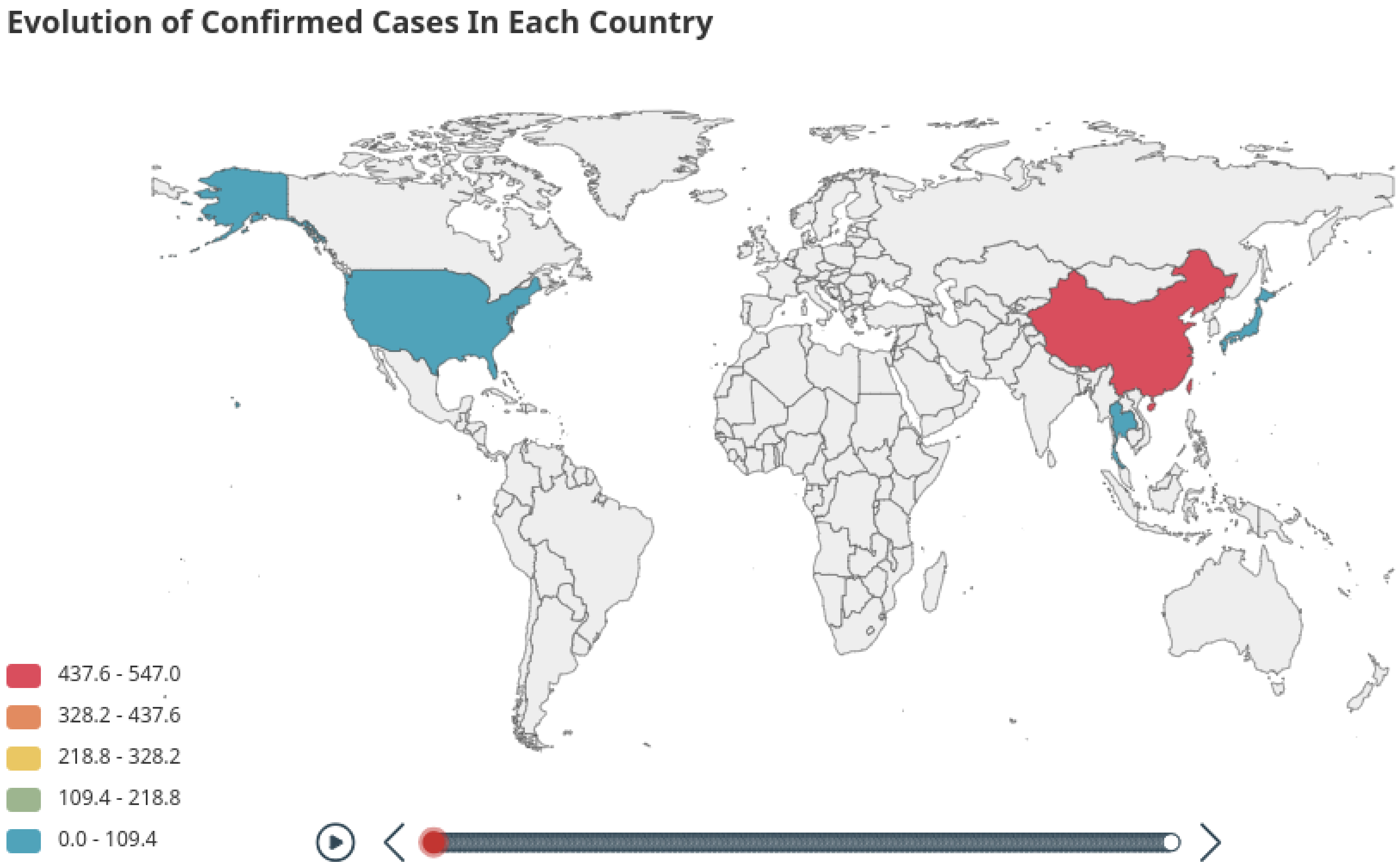
In [38]:

```python
def get_chart(date: str, case: str):
    ob_data = data_per_country[data_per_country['ObservationDate']==date]
    min_data, max_data = (ob_data[case].min(),ob_data[case].max())
    #print(min_data,max_data)
    map_chart = (
        Map()
        .add(
            series_name="",
            data_pair=[list(z) for z in zip(ob_data['Country/Region'].values.tolist(),
                                          ob_data[case].values.tolist())],
            label_opts=opts.LabelOpts(is_show=False),
            is_map_symbol_show=False,
            maptype='world'
        )
        .set_global_opts(
            title_opts=opts.TitleOpts(
                title="Evolution of {0} Cases In Each Country".format(case),
            ),
            visualmap_opts=opts.VisualMapOpts(is_calculable=True, min_=min_data, max_=int(max_data), is_piecewise=True)
        )
    )

    grid_chart = (
        Grid()
        .add(map_chart, grid_opts=opts.GridOpts())
    )

    return grid_chart
```

In [39]:

```
time_list = data_per_country['ObservationDate'].unique().tolist()
timeline = Timeline(
    init_opts=opts.InitOpts()
)
for d in time_list:
    g = get_chart(d,'Confirmed')
    timeline.add(g, time_point=str(d))

timeline.add_schema(
    orient="horizontal",
    is_auto_play=False,
    is_inverse=False,
    label_opts=opts.LabelOpts(is_show=True, color="#fff"),
)
timeline.render_notebook()
```

Out[39]:

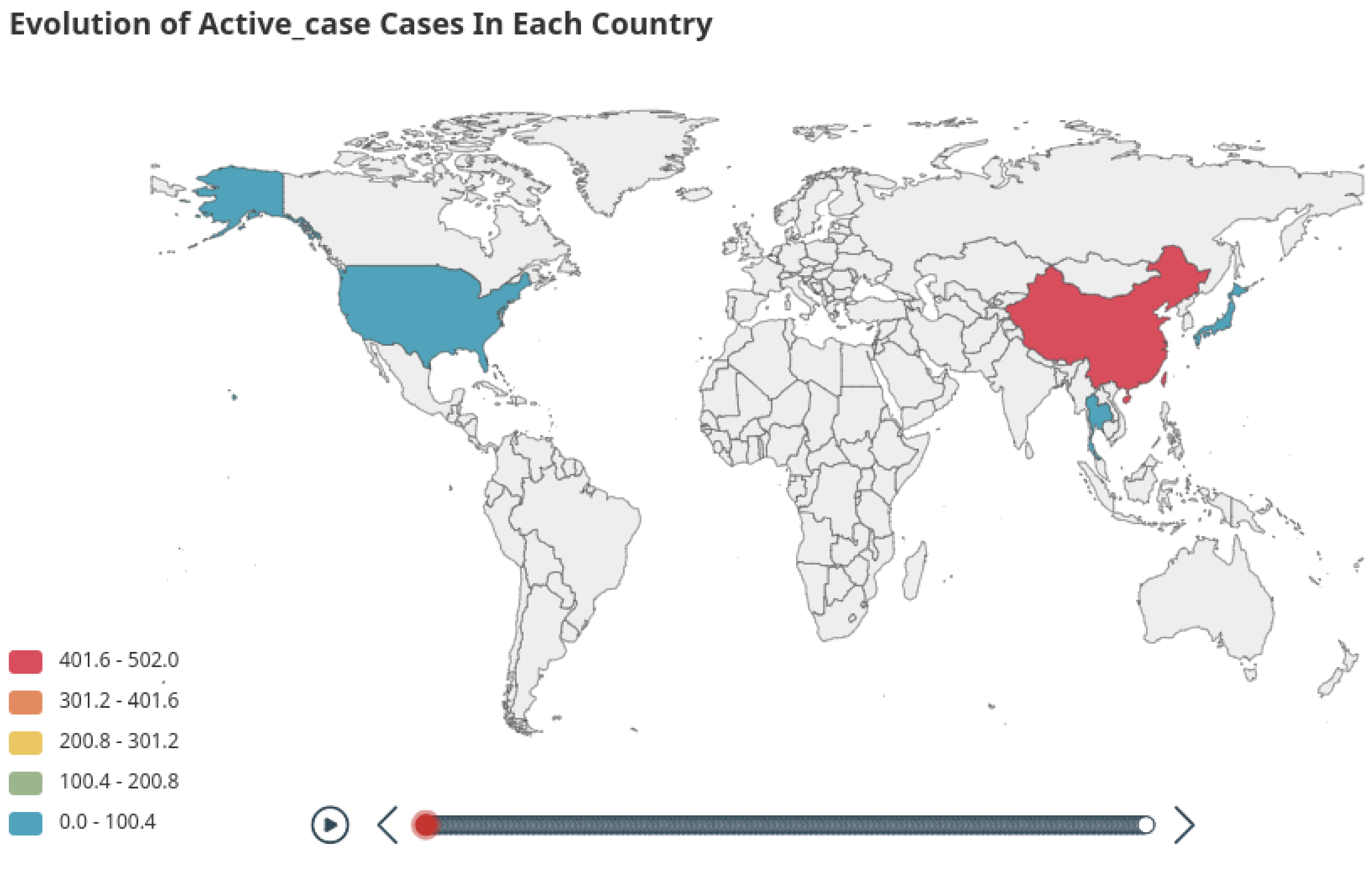


Evolution of Confirmed Cases In Each Country

In [40]:

```
timeline = Timeline(
    init_opts=opts.InitOpts()
)
for d in time_list:
    g = get_chart(d,'Active_case')
    timeline.add(g, time_point=str(d))

timeline.add_schema(
    orient="horizontal",
    is_auto_play=False,
    is_inverse=False,
    label_opts=opts.LabelOpts(is_show=True, color="#fff"),
)
timeline.render_notebook()
```

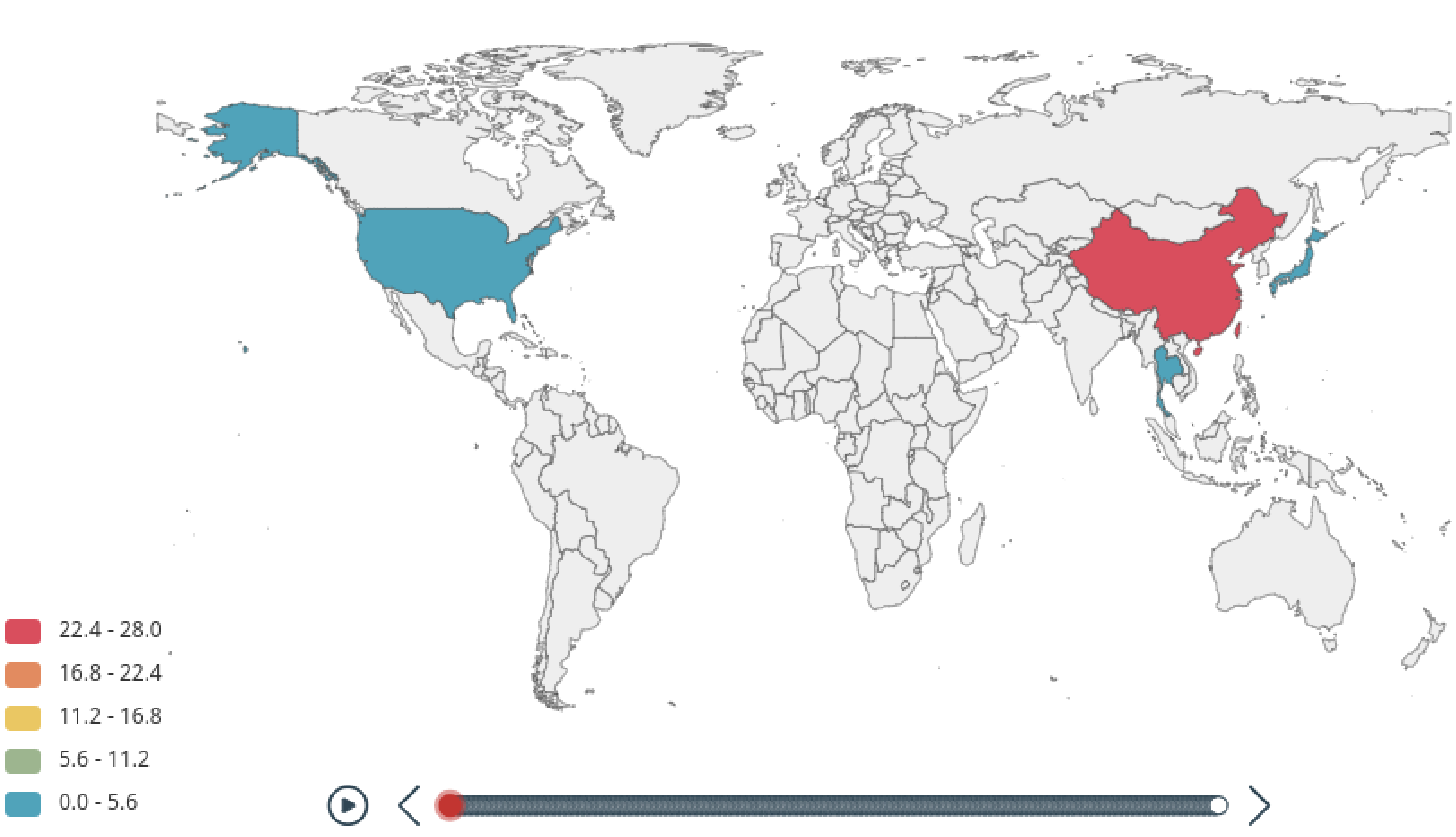
Out[40]:



In [41]:

```
timeline = Timeline(
    init_opts=opts.InitOpts()
)
for d in time_list:
    g = get_chart(d,'Recovered')
    timeline.add(g, time_point=str(d))

timeline.add_schema(
    orient="horizontal",
    is_auto_play=False,
    is_inverse=False,
    label_opts=opts.LabelOpts(is_show=True, color="#fff"),
)
timeline.render_notebook()
```

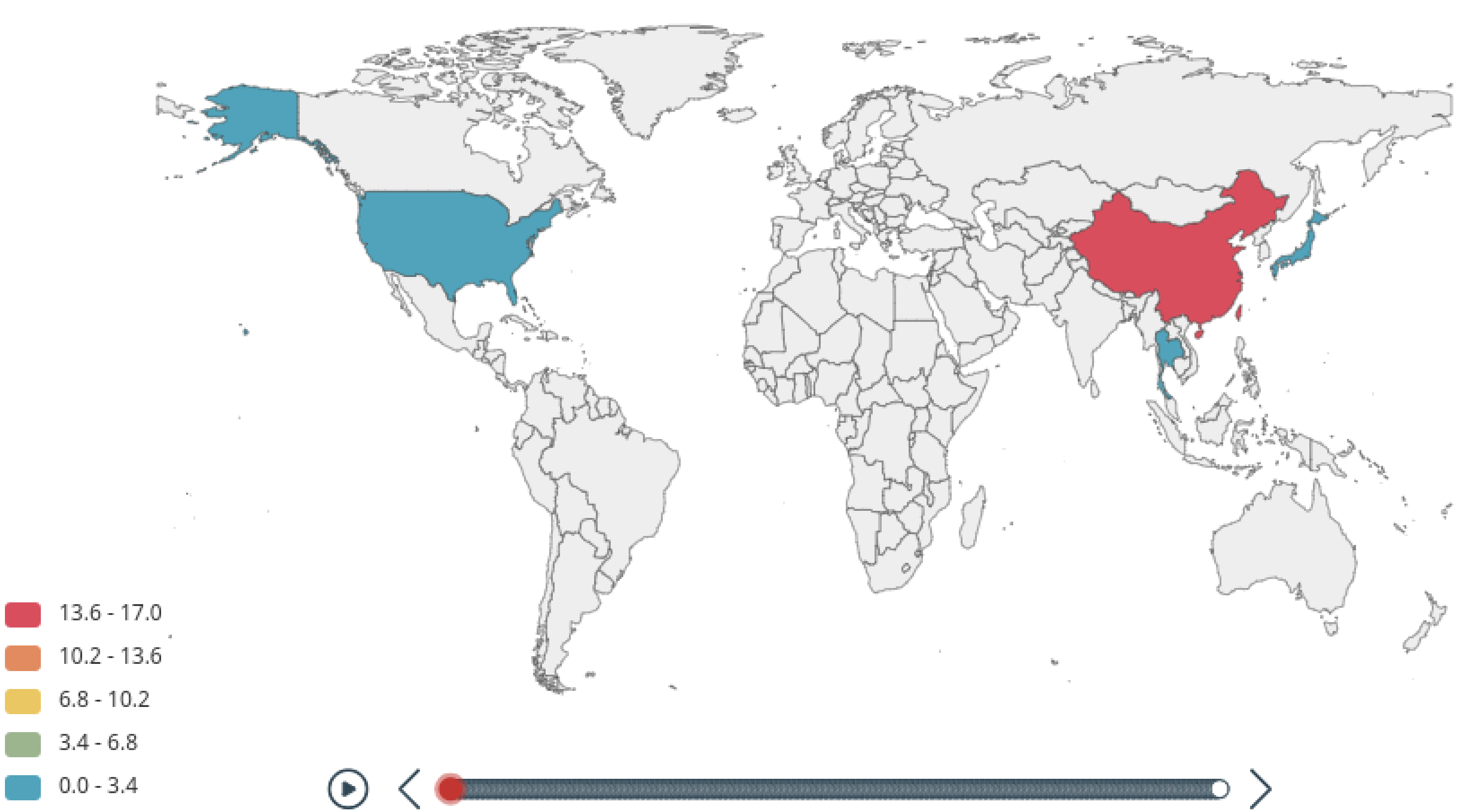
Out[41]:



Evolution of Recovered Cases In Each Country

```
In [42]: timeline = Timeline(
             init_opts=opts.InitOpts()
         )
         for d in time_list:
             g = get_chart(d,'Deaths')
             timeline.add(g, time_point=str(d))

         timeline.add_schema(
             orient="horizontal",
             is_auto_play=False,
             is_inverse=False,
             label_opts=opts.LabelOpts(is_show=True, color="#fff"),
         )
         timeline.render_notebook()
```

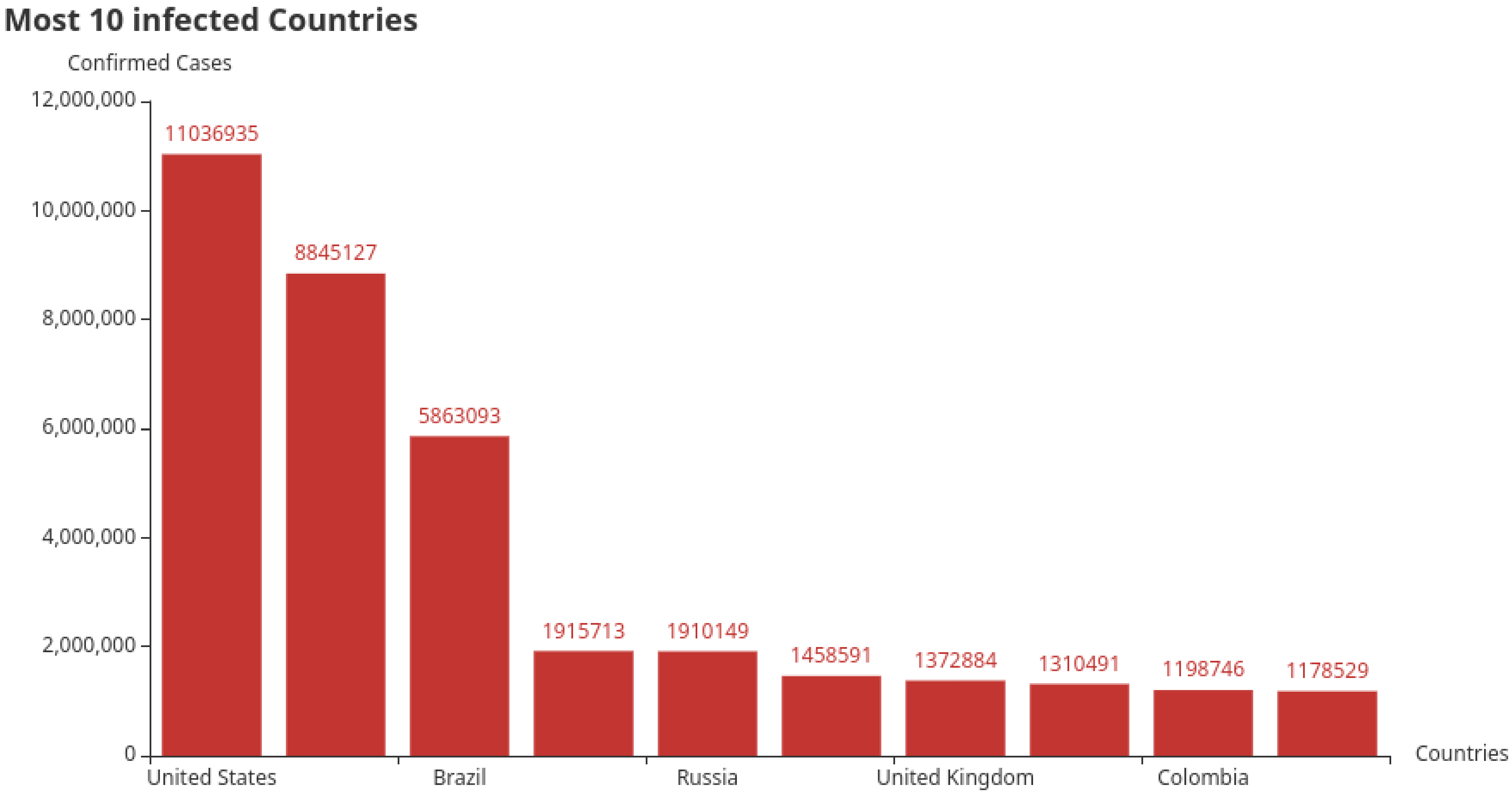
Out[42]: **Evolution of Deaths Cases In Each Country**



In [43]:

```
(
    Bar(init_opts=opts.InitOpts())
    .set_global_opts(
        title_opts=opts.TitleOpts(title="Most 10 infected Countries"),
        xaxis_opts=opts.AxisOpts(type_='category', name="Countries"),
        yaxis_opts=opts.AxisOpts(
            type_="value", name="Confirmed Cases"
        ),
        tooltip_opts=opts.TooltipOpts(is_show=True),
    )
    .add_xaxis(Data_per_country['Country/Region'][0:10].values.tolist())
    .add_yaxis("", Data_per_country['Confirmed'][0:10].values.tolist(), label_opts=opts.LabelOpts(is_show=True),)
    .render_notebook()
)
```

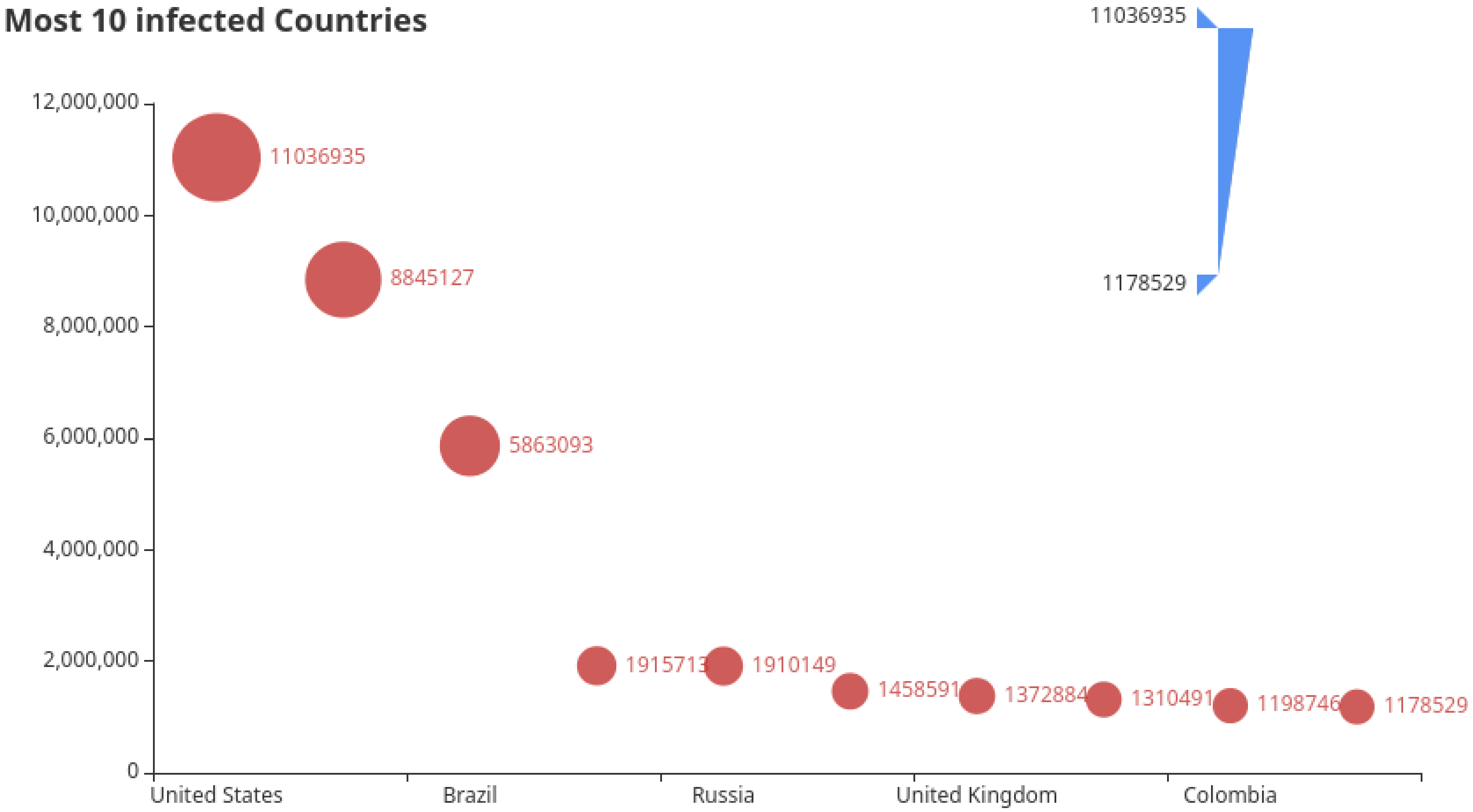
Out[43]:



In [44]:

```
(
    Scatter()
    .add_xaxis(Data_per_country['Country/Region'][0:10].values.tolist())
    .add_yaxis("", Data_per_country['Confirmed'][0:10].values.tolist())
    .set_global_opts(
        title_opts=opts.TitleOpts(title="Most 10 infected Countries"),
        visualmap_opts=opts.VisualMapOpts(type_="size",
                                          min_=int(Data_per_country['Confirmed'][0:10].min()),
                                          max_=int(Data_per_country['Confirmed'][0:10].max()),
                                          orient="vertical",pos_right="20%",pos_top=0),
    )
    .render_notebook()
)
```

Out[44]:



- **Lets start with good news : Recorvered cases in each Country**

In [45]:

```
# *計算恢復數 * *Recorvered cases in each Country *
Recovered_per_country = Data.groupby(["Country/Region"])["Recovered"].sum().reset_index().sort_values("Recovered",ascending=False).reset_index(drop=True)
Recovered_per_country.head()
```

Out[45]:

	Country/Region	Recovered
0	India	8249579
1	Brazil	5279452
2	US	4174884
3	Russia	1429565
4	Argentina	1129102

In [46]:

```
case_sum = Recovered_per_country["Recovered"].sum()
Recovered_per_country["Recovered_per"] = Recovered_per_country["Recovered"]/case_sum
Recovered_per_country.head()
```

Out[46]:

	Country/Region	Recovered	Recovered_per
0	India	8249579	0.236005
1	Brazil	5279452	0.151035
2	US	4174884	0.119435
3	Russia	1429565	0.040897
4	Argentina	1129102	0.032301

In [47]:

```
table = Table()
headers = ['Country','Recovered Cases']
table.add(headers, [[country,confirmed] for country,confirmed in zip(Data_per_country['Country/Region'].tolist()
                                                                       ,Data_per_country['Recovered'].tolist())])
table.set_global_opts(
    title_opts=ComponentTitleOpts(title="Recovered Cases In Each Country")
)
table.render_notebook()
```

Out[47]:

Recovered Cases In Each Country

Country	Recovered Cases
United States	4174884
India	8249579
Brazil	5279452
France	139760
Russia	1429565
Spain	150376
United Kingdom	3121
Argentina	1129102
Colombia	1104956
Italy	420810
Mexico	750190
Peru	858388
Germany	505714
Iran	558818
South Africa	693467
Poland	294783
Ukraine	251864
Belgium	0
Chile	506700
Iraq	447039

Country	Recovered Cases
Indonesia	391991
Czech Republic	327306
Netherlands	6448
Bangladesh	349542
Turkey	353663
Philippines	374329
Romania	243832
Pakistan	323824
Saudi Arabia	340304
Israel	313380
Canada	240434
Morocco	238598
Switzerland	141000
Portugal	125066
Nepal	173430
Austria	125674
Ecuador	160639
Sweden	0
United Arab Emirates	143252
Panama	125370
Jordan	10067

Country	Recovered Cases
Bolivia	117137
Hungary	31599
Kuwait	127585
Qatar	132784
Dominican Republic	109981
Costa Rica	75341
Kazakhstan	109944
Oman	111096
Japan	91954
Armenia	74984
Guatemala	104438
Belarus	95843
Egypt	101046
Lebanon	60416
Honduras	44778
Ethiopia	63866
Bulgaria	29375
Venezuela	92373
Moldova	70812
Slovakia	30519
China	81332

Country	Recovered Cases
Bahrain	82469
Serbia	0
Croatia	66231
Tunisia	54362
Georgia	62061
Azerbaijan	55150
Greece	23074
Libya	44133
Bosnia and Herzegovina	35606
Paraguay	51375
Kenya	45766
Uzbekistan	67526
Burma	52351
Ireland	23364
Algeria	44633
Kyrgyzstan	57760
Nigeria	61073
West Bank and Gaza	54346
Denmark	48259
Singapore	58029
Slovenia	34864

Country	Recovered Cases
Ghana	48328
Malaysia	34785
North Macedonia	27364
Afghanistan	35092
El Salvador	32721
Lithuania	7753
Kosovo	17305
South Korea	25759
Norway	11863
Albania	12889
Australia	25468
Montenegro	17387
Luxembourg	15428
Cameroon	21510
Ivory Coast	20647
Finland	14000
Madagascar	16592
Sri Lanka	11495
Zambia	16137
Uganda	8038
Senegal	15416

Country	Recovered Cases
Sudan	9571
Mozambique	12505
Namibia	12680
Angola	6444
Guinea	11151
Maldives	11289
Congo (Kinshasa)	11086
Tajikistan	10973
Latvia	1515
Jamaica	5248
Cabo Verde	9220
Haiti	7701
Gabon	8939
Zimbabwe	8096
Botswana	5559
Malta	5765
Mauritania	7524
Estonia	4741
Cuba	7060
Bahamas	5323
Cyprus	2021

Country	Recovered Cases
Syria	2714
Eswatini	5779
Trinidad and Tobago	5402
Malawi	5399
Andorra	4747
Nicaragua	4225
Djibouti	5525
Congo (Brazzaville)	3887
Hong Kong	5194
Rwanda	4995
Suriname	5148
Iceland	4813
Equatorial Guinea	4973
Central African Republic	1924
Belize	2598
Guyana	3777
Somalia	3330
Uruguay	3233
Mali	2949
Thailand	3721
Gambia	3571

Country	Recovered Cases
South Sudan	1290
Benin	2515
Togo	1920
Burkina Faso	2445
Guinea-Bissau	2255
Sierra Leone	1820
Yemen	1394
Lesotho	1056
New Zealand	1918
Chad	1443
Liberia	1314
Niger	1148
Vietnam	1103
San Marino	888
Liechtenstein	682
Sao Tome and Principe	917
Diamond Princess	659
Burundi	549
Papua New Guinea	586
Taiwan	536
Comoros	549

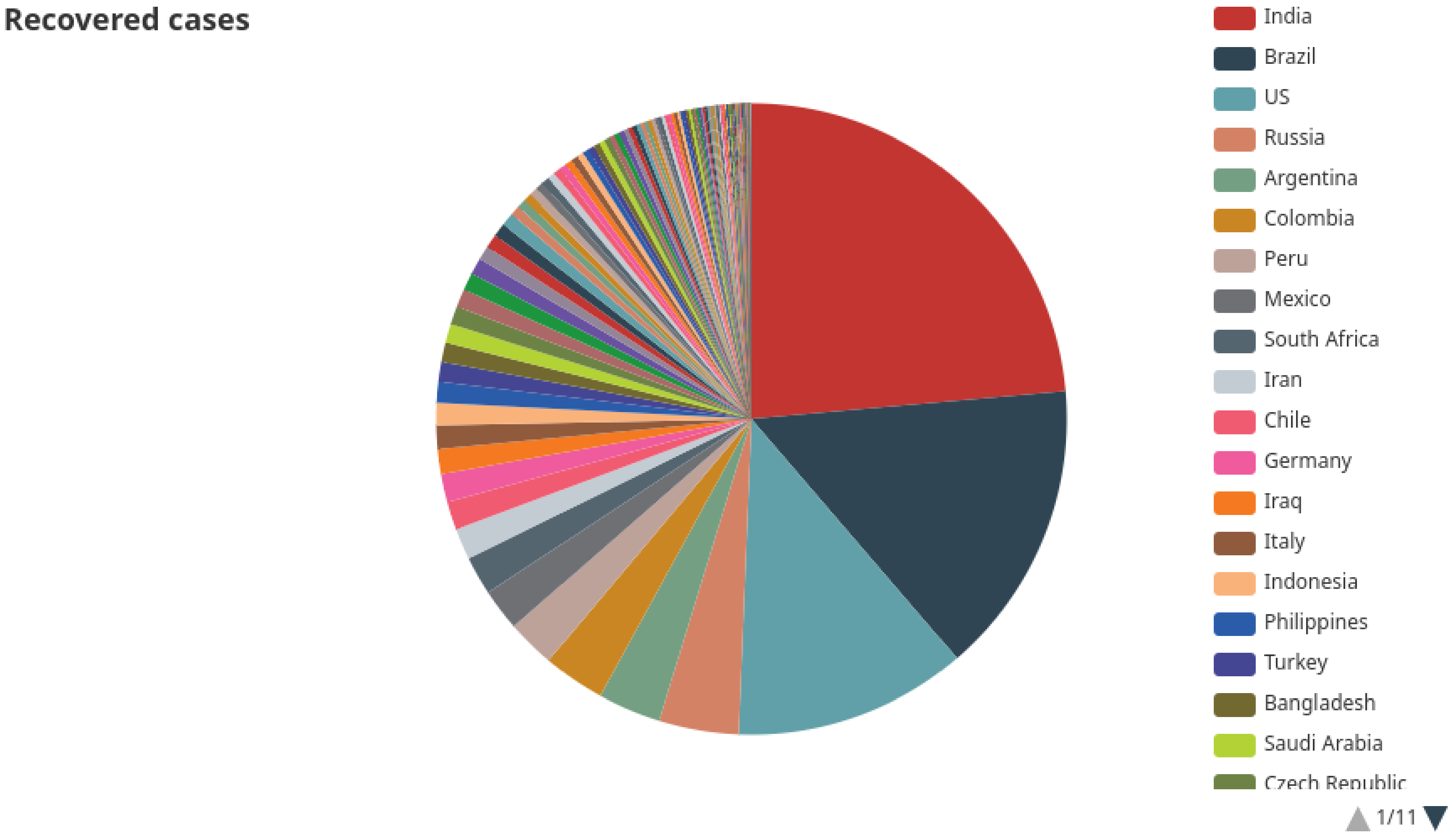
Country	Recovered Cases
Monaco	450
Tanzania	183
Eritrea	444
Mauritius	431
Mongolia	328
Bhutan	355
Cambodia	289
Barbados	238
Saint Lucia	46
Seychelles	157
Brunei	145
Antigua and Barbuda	127
Saint Vincent and the Grenadines	77
Dominica	41
Macau	46
Fiji	32
Grenada	27
Timor-Leste	30
Holy See	15
Laos	23
Saint Kitts and Nevis	19

Country	Recovered Cases
Solomon Islands	5
Western Sahara	8
MS Zaandam	0
Marshall Islands	1
Vanuatu	0

In [48]:

```
(
    Pie()
    .add("Cases", [list(z) for z in zip(Recovered_per_country['Country/Region'].values.tolist(),
                                      Recovered_per_country['Recovered'].values.tolist())])
    .set_global_opts(title_opts=opts.TitleOpts(title='Recovered cases')
                    ,legend_opts=opts.LegendOpts(type_="scroll", orient="vertical"
                                                 , pos_left="80%", is_show=True,
                                                ))
    .set_series_opts(label_opts=opts.LabelOpts(is_show=False,
                                              formatter=JsCode(
                "function(x){ if(x.value>0.1){ return x.name + ':' + Number(x.value);} else {return '';} }"
            )))
    .render_notebook()
)
```

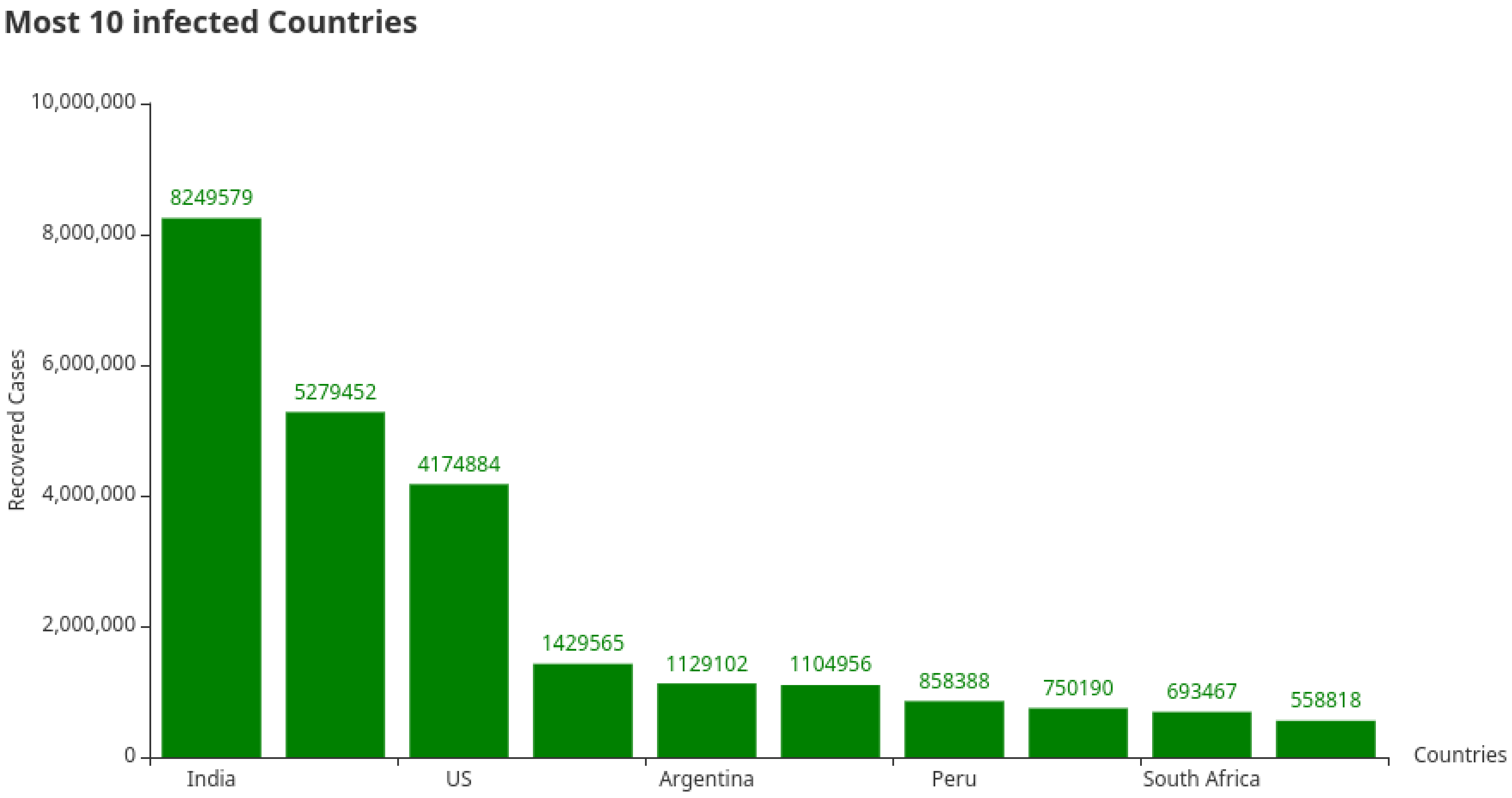
Out[48]:



In [49]:

```
(
    Bar(init_opts=opts.InitOpts())
    .set_global_opts(
        title_opts=opts.TitleOpts(title="Most 10 infected Countries"),
        xaxis_opts=opts.AxisOpts(type_='category', name="Countries",boundary_gap=["5%","5%"]),
        yaxis_opts=opts.AxisOpts(
            type_="value", name="Recovered Cases",
            name_rotate=90,name_location="middle",name_gap=70
        ),
        tooltip_opts=opts.TooltipOpts(is_show=True),
    )
    .add_xaxis(Recovered_per_country['Country/Region'][0:10].values.tolist())
    .add_yaxis("", Recovered_per_country['Recovered'][0:10].values.tolist(),
               itemstyle_opts=opts.ItemStyleOpts(color="green"),
               label_opts=opts.LabelOpts(is_show=True),)
    .render_notebook()
)
```

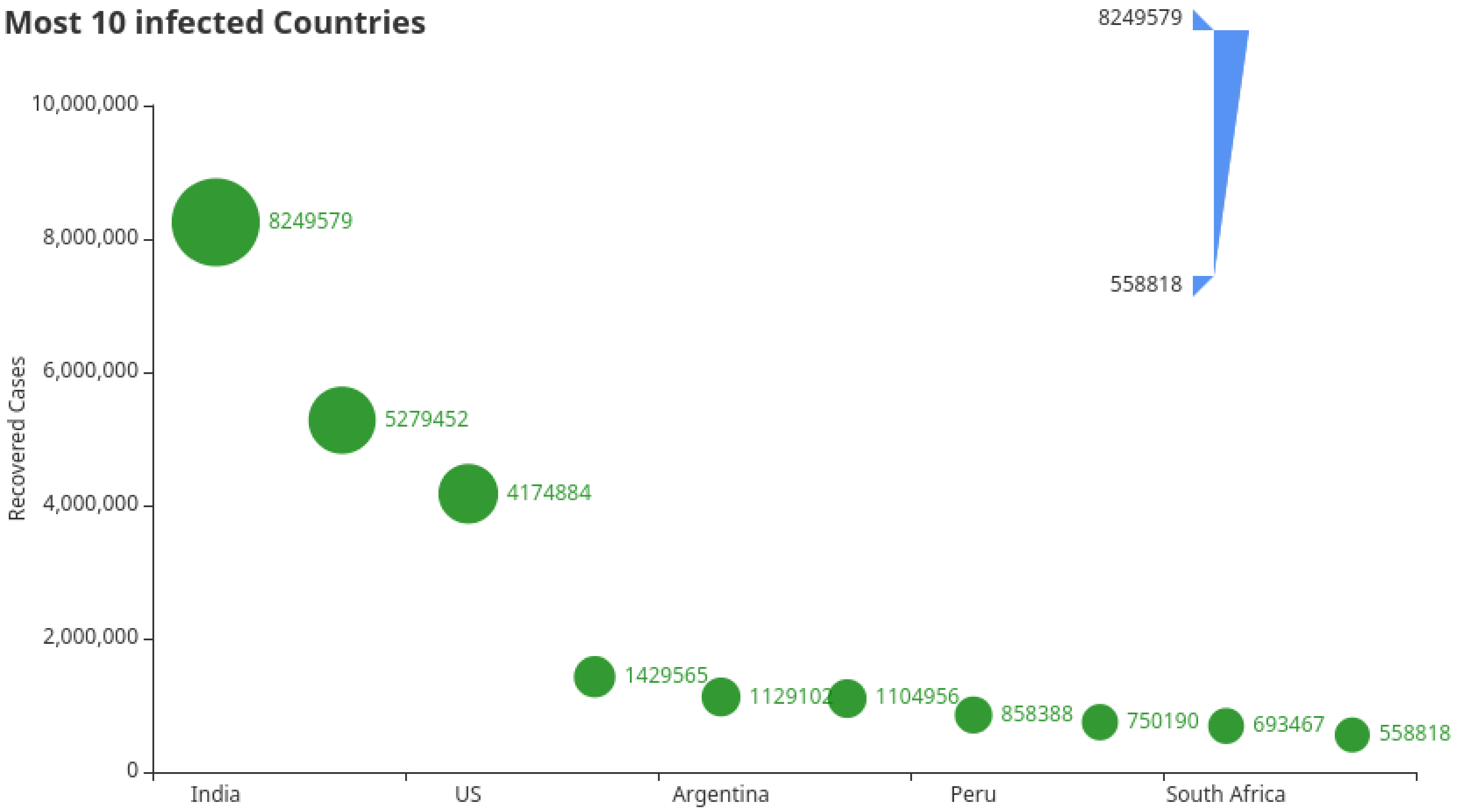
```
Out[49]:
```


Most 10 infected Countries
Recovered Cases
10,000,000
8,000,000
6,000,000
4,000,000
2,000,000
0
8249579
5279452
4174884
1429565
1129102
1104956
858388
750190
693467
558818
India
US
Argentina
Peru
South Africa
Countries

In [50]:

```
(
    Scatter()
    .add_xaxis(Recovered_per_country['Country/Region'][0:10].values.tolist())
    .add_yaxis("", Recovered_per_country['Recovered'][0:10].values.tolist(),
              itemstyle_opts=opts.ItemStyleOpts(color="green"))
    .set_global_opts(
        title_opts=opts.TitleOpts(title="Most 10 infected Countries"),
        visualmap_opts=opts.VisualMapOpts(type_="size",
                                          min_=int(Recovered_per_country['Recovered'][0:10].min()),
                                          max_=int(Recovered_per_country['Recovered'][0:10].max()),
                                          orient="vertical",pos_right="20%",pos_top=0),
        yaxis_opts=opts.AxisOpts(
            type_="value", name="Recovered Cases",
            name_rotate=90,name_location="middle",name_gap=70
        ),
    )
    .render_notebook()
)
```

Out[50]:



Most 10 infected Countries

Active cases in each Country

In [51]:

```
Active_per_country = Data.groupby(["Country/Region"])["Active_case"].sum().reset_index().sort_values("Active_case",a
scending=False).reset_index(drop=True)
Active_per_country.head()
```

Out[51]:

	Country/Region	Active_case
0	US	6615837
1	France	1733352
2	UK	1317737
3	Spain	1267446
4	Italy	712490

```
In [52]: table = Table()
         headers = ['Country','Active Cases']
         table.add(headers, [[country,confirmed] for country,confirmed in zip(Active_per_country['Country/Region'].tolist()
                                                                              ,Active_per_country['Active_case'].tolist())])
         table.set_global_opts(
             title_opts=ComponentTitleOpts(title="Active Cases In Each Country")
         )
         table.render_notebook()
```

Out[52]:

Active Cases In Each Country

Country	Active Cases
US	6615837
France	1733352
UK	1317737
Spain	1267446
Italy	712490
Belgium	521518
India	465478
Russia	447699
Netherlands	439954
Brazil	417843
Poland	407841
Ukraine	289765
Germany	284659
Sweden	171191
Iran	161757
Mexico	157790
Argentina	145953
Jordan	131839
Czech Republic	126602
Switzerland	112766

Country	Active Cases
Romania	107523
Hungary	106265
Portugal	88854
Serbia	83559
Bangladesh	76597
Austria	76453
Bulgaria	66746
Iraq	60443
Indonesia	59911
Colombia	59759
Slovakia	55738
Honduras	55501
Greece	50025
Morocco	49800
Turkey	49108
Canada	47987
Costa Rica	46336
Lebanon	44197
Ireland	42560
Peru	41334
Armenia	40590

Country	Active Cases
South Africa	37316
Ethiopia	37285
Nepal	35125
Bosnia and Herzegovina	34461
Libya	28452
Pakistan	28048
Lithuania	26728
Philippines	25677
Japan	24783
Tunisia	23697
Kenya	23210
Dominican Republic	21458
Algeria	20892
Azerbaijan	19571
Slovenia	19381
Paraguay	18612
Panama	18410
North Macedonia	18381
Belarus	17296
Bolivia	17260
Croatia	16926

Country	Active Cases
Georgia	16914
Moldova	16448
Norway	16277
Burma	15066
Albania	14318
Denmark	13633
Malaysia	12323
Kosovo	10990
Luxembourg	10284
Chile	9754
Montenegro	9399
Kazakhstan	9208
Latvia	8909
Kuwait	8417
West Bank and Gaza	8120
Kyrgyzstan	8020
Oman	7955
Uganda	7837
Israel	7629
Saudi Arabia	7294
Angola	6685

Country	Active Cases
Ecuador	6648
United Arab Emirates	6563
Afghanistan	6531
Guatemala	6515
Sri Lanka	5734
Cyprus	5118
Finland	4946
Jamaica	4405
Venezuela	4128
Sudan	3939
Syria	3625
Egypt	3268
Central African Republic	2913
Nigeria	2912
Estonia	2815
Qatar	2767
Botswana	2639
El Salvador	2593
South Korea	2516
Belize	2173
Malta	2172

Country	Active Cases
Uzbekistan	2119
Bahrain	1900
Mozambique	1830
Bahamas	1707
South Sudan	1654
Congo (Brazzaville)	1536
Ghana	1473
Guinea	1385
Australia	1374
Nicaragua	1278
Haiti	1235
Andorra	1049
Lesotho	941
Guyana	907
Maldives	872
Somalia	864
Mali	810
Cameroon	749
Uruguay	732
Togo	694
Namibia	690

Country	Active Cases
Zambia	633
Trinidad and Tobago	566
Tajikistan	552
Cabo Verde	499
Sierra Leone	497
Zimbabwe	476
Madagascar	468
Rwanda	415
Cuba	399
Malawi	381
China	380
Congo (Kinshasa)	355
Iceland	351
San Marino	323
Tanzania	305
Liechtenstein	298
Benin	286
Mauritania	263
Ivory Coast	201
Eswatini	197
Hong Kong	156

Country	Active Cases
Vietnam	143
Burkina Faso	128
Guinea-Bissau	121
Saint Lucia	114
Liberia	111
Mongolia	103
Monaco	96
Thailand	94
Niger	89
Burundi	78
Yemen	73
Gabon	65
Djibouti	63
Singapore	62
Taiwan	59
New Zealand	58
Chad	53
Senegal	49
Eritrea	49
Equatorial Guinea	46
Diamond Princess	40

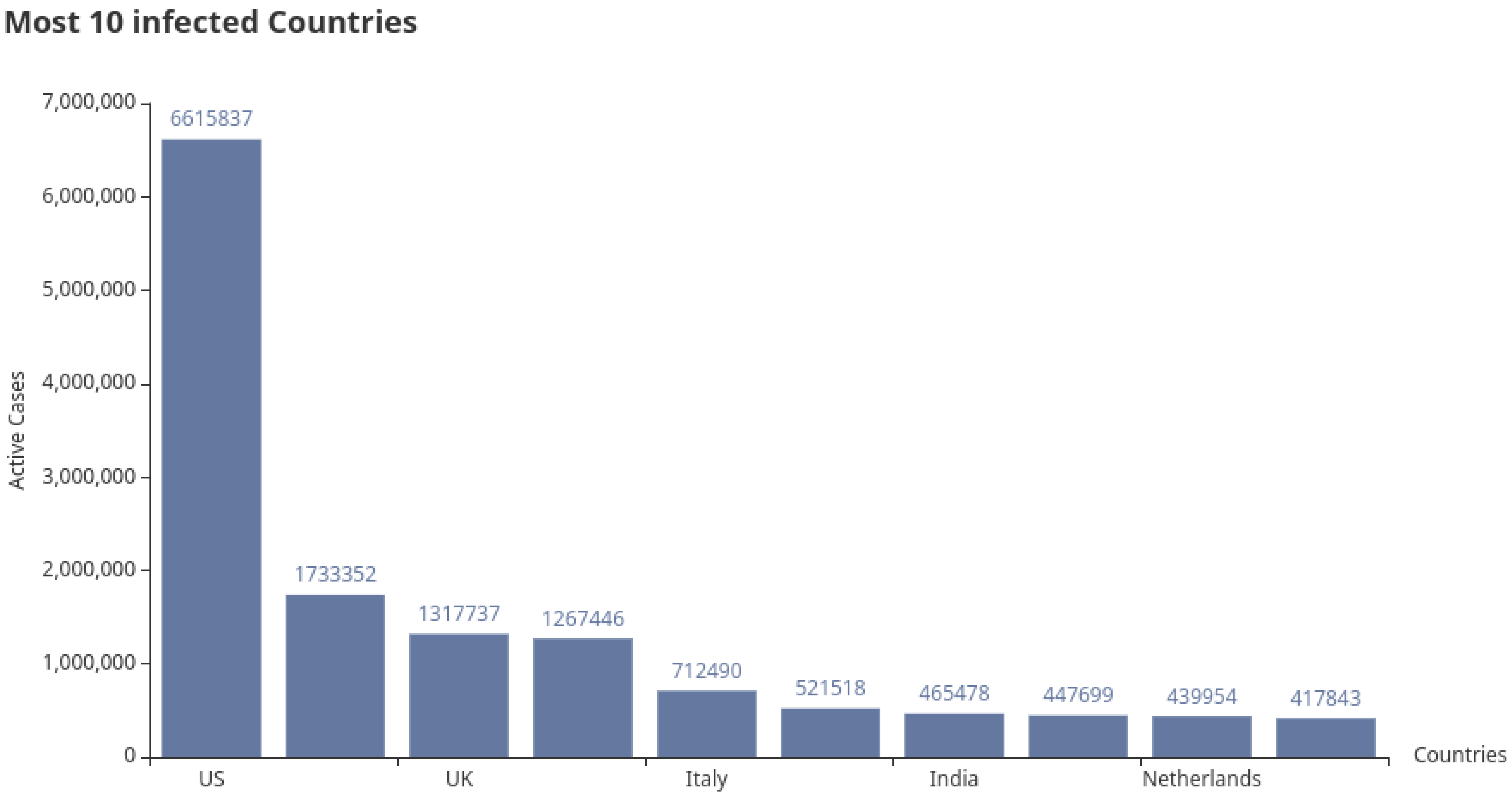
Country	Active Cases
Mauritius	37
Sao Tome and Principe	32
Dominica	27
Comoros	23
Bhutan	20
Cambodia	13
Suriname	12
Holy See	12
Solomon Islands	11
Gambia	9
Papua New Guinea	9
MS Zaandam	7
Grenada	5
Barbados	5
Seychelles	3
Antigua and Barbuda	3
Fiji	1
Western Sahara	1
Laos	1
Saint Vincent and the Grenadines	1
Vanuatu	1

Country	Active Cases
Saint Kitts and Nevis	0
Timor-Leste	0
Macau	0
Brunei	0
Marshall Islands	0

In [53]:

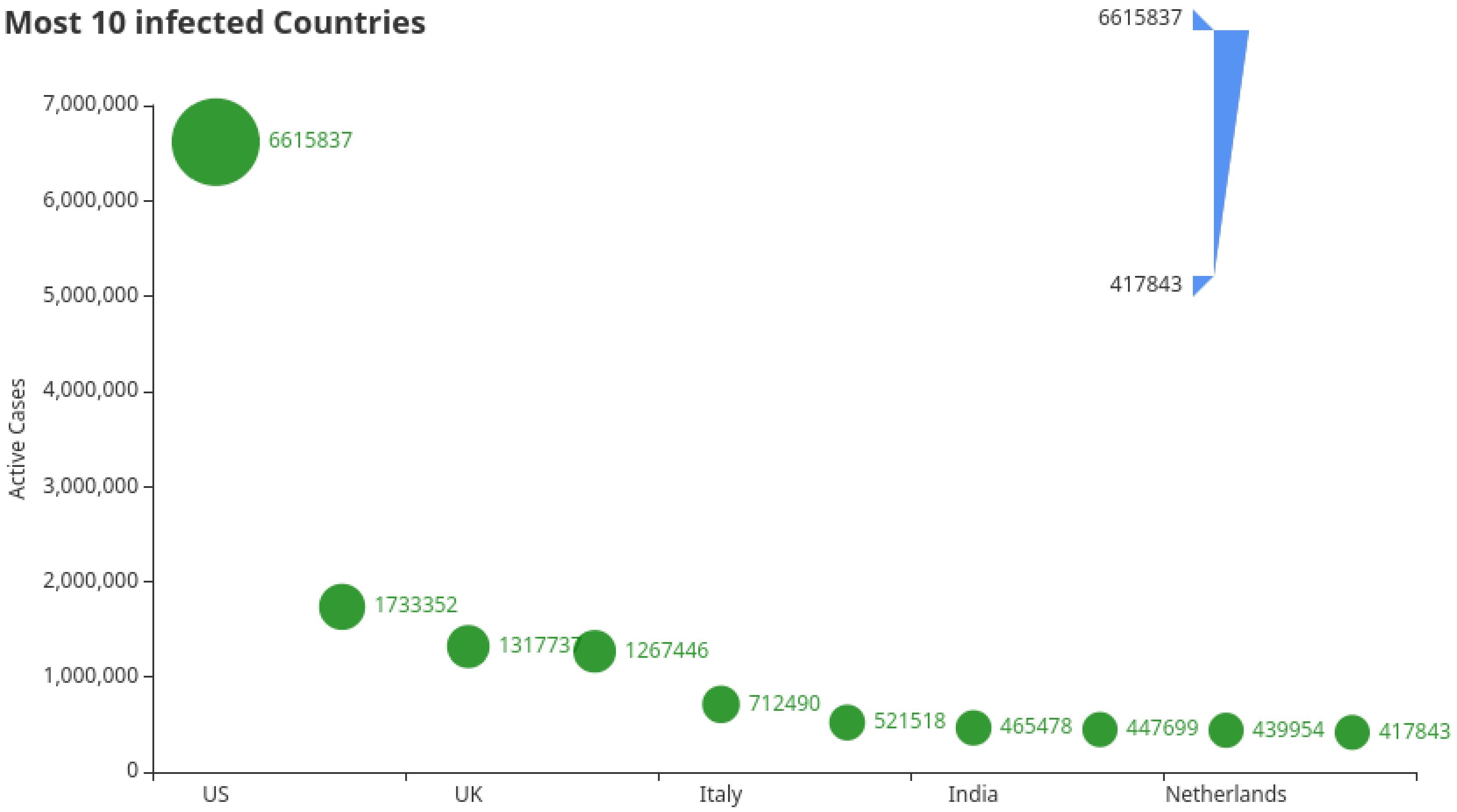
```
(
    Bar(init_opts=opts.InitOpts())
    .set_global_opts(
        title_opts=opts.TitleOpts(title="Most 10 infected Countries"),
        xaxis_opts=opts.AxisOpts(type_='category', name="Countries",boundary_gap=["5%","5%"]),
        yaxis_opts=opts.AxisOpts(
            type_="value", name="Active Cases",
            name_rotate=90,name_location="middle",name_gap=70
        ),
        tooltip_opts=opts.TooltipOpts(is_show=True),
    )
    .add_xaxis(Active_per_country['Country/Region'][0:10].values.tolist())
    .add_yaxis("", Active_per_country['Active_case'][0:10].values.tolist(),
               itemstyle_opts=opts.ItemStyleOpts(color="rgb(100,120,160)"),
               label_opts=opts.LabelOpts(is_show=True),)
    .render_notebook()
)
```

Out[53]:


Most 10 infected Countries
Active Cases
7,000,000
6,000,000
5,000,000
4,000,000
3,000,000
2,000,000
1,000,000
0
6615837
1733352
1317737
1267446
712490
521518
465478
447699
439954
417843
US
UK
Italy
India
Netherlands
Countries

```
In [54]: (
             Scatter()
             .add_xaxis(Active_per_country['Country/Region'][0:10].values.tolist())
             .add_yaxis("", Active_per_country['Active_case'][0:10].values.tolist(),
                       itemstyle_opts=opts.ItemStyleOpts(color="green"))
             .set_global_opts(
                 title_opts=opts.TitleOpts(title="Most 10 infected Countries"),
                 visualmap_opts=opts.VisualMapOpts(type_="size",
                                                   min_=int(Active_per_country['Active_case'][0:10].min()),
                                                   max_=int(Active_per_country['Active_case'][0:10].max()),
                                                   orient="vertical",pos_right="20%",pos_top=0),
                 yaxis_opts=opts.AxisOpts(
                     type_="value", name="Active Cases",
                     name_rotate=90,name_location="middle",name_gap=70
                 ),
             )
             .render_notebook()
         )
```

Out[54]:



Deaths cases in each Country

In [55]:

```
Deaths_per_country = Data.groupby(["Country/Region"])["Deaths"].sum().reset_index().sort_values("Deaths",ascending=False).reset_index(drop=True)
Deaths_per_country.head()
```

Out[55]:

	Country/Region	Deaths
0	US	246214
1	Brazil	165798
2	India	130070
3	Mexico	98542
4	UK	52026

In [56]:

```
table = Table()
headers = ['Country','Deaths']
table.add(headers, [[country,confirmed] for country,confirmed in zip(Deaths_per_country['Country/Region'].tolist()
                                                                     ,Deaths_per_country['Deaths'].tolist())])
table.set_global_opts(
    title_opts=ComponentTitleOpts(title="Deaths In Each Country")
)
table.render_notebook()
```

Out[56]:

Deaths In Each Country

Country	Deaths
US	246214
Brazil	165798
India	130070
Mexico	98542
UK	52026
Italy	45229
France	42601
Iran	41493
Spain	40769
Argentina	35436
Peru	35177
Colombia	34031
Russia	32885
South Africa	20241
Indonesia	15211
Chile	14819
Belgium	14421
Ecuador	13008
Germany	12573
Iraq	11670

Country	Deaths
Turkey	11507
Canada	11001
Poland	10348
Ukraine	9904
Romania	8926
Bolivia	8849
Netherlands	8559
Philippines	7832
Pakistan	7160
Egypt	6453
Czech Republic	6208
Bangladesh	6194
Sweden	6164
Saudi Arabia	5657
Morocco	4779
China	4634
Guatemala	3932
Portugal	3381
Switzerland	3369
Hungary	3097
Panama	2873

Country	Deaths
Honduras	2823
Israel	2732
Tunisia	2345
Dominican Republic	2285
Algeria	2154
Bulgaria	2130
Moldova	2019
Ireland	1979
Kazakhstan	1899
Bosnia and Herzegovina	1889
Japan	1874
Austria	1829
Jordan	1772
Armenia	1763
Afghanistan	1617
Paraguay	1587
Burma	1577
Ethiopia	1569
Costa Rica	1546
Oman	1338
North Macedonia	1305

Country	Deaths
Kenya	1269
Nepal	1221
Kyrgyzstan	1203
Nigeria	1163
Sudan	1116
Greece	1106
Croatia	1049
Belarus	1046
El Salvador	1044
Libya	1017
Serbia	1009
Azerbaijan	967
Australia	907
Venezuela	851
Kuwait	838
Kosovo	822
Lebanon	817
Slovenia	797
Denmark	760
Georgia	703
Albania	623

Country	Deaths
Yemen	605
Uzbekistan	598
West Bank and Gaza	565
United Arab Emirates	530
Slovakia	510
South Korea	494
Cameroon	433
Montenegro	391
Finland	369
Zambia	353
Syria	345
Bahrain	334
Senegal	328
Angola	322
Ghana	322
Congo (Kinshasa)	319
Malaysia	309
Norway	294
Lithuania	277
Zimbabwe	257
Madagascar	250

Country	Deaths
Qatar	234
Haiti	232
Jamaica	231
Luxembourg	219
Malawi	185
Mauritania	165
Nicaragua	158
Bahamas	156
Uganda	145
Mali	141
Guyana	139
Namibia	138
Cuba	131
Ivory Coast	128
Latvia	123
Gambia	122
Eswatini	119
Suriname	114
Mozambique	113
Trinidad and Tobago	112
Hong Kong	108

Country	Deaths
Somalia	107
Cabo Verde	103
Chad	101
Malta	97
Congo (Brazzaville)	92
Belize	90
Equatorial Guinea	85
Tajikistan	85
Liberia	82
Estonia	81
Andorra	76
Guinea	75
Sierra Leone	74
Niger	70
Burkina Faso	68
Uruguay	65
Central African Republic	63
Togo	61
Djibouti	61
Thailand	60
South Sudan	59

Country	Deaths
Gabon	58
Sri Lanka	58
Rwanda	45
Lesotho	44
Benin	43
Guinea-Bissau	43
Maldives	43
San Marino	42
Cyprus	39
Vietnam	35
Singapore	28
Botswana	27
New Zealand	25
Iceland	25
Tanzania	21
Sao Tome and Principe	16
Diamond Princess	13
Mauritius	10
Comoros	7
Barbados	7
Taiwan	7

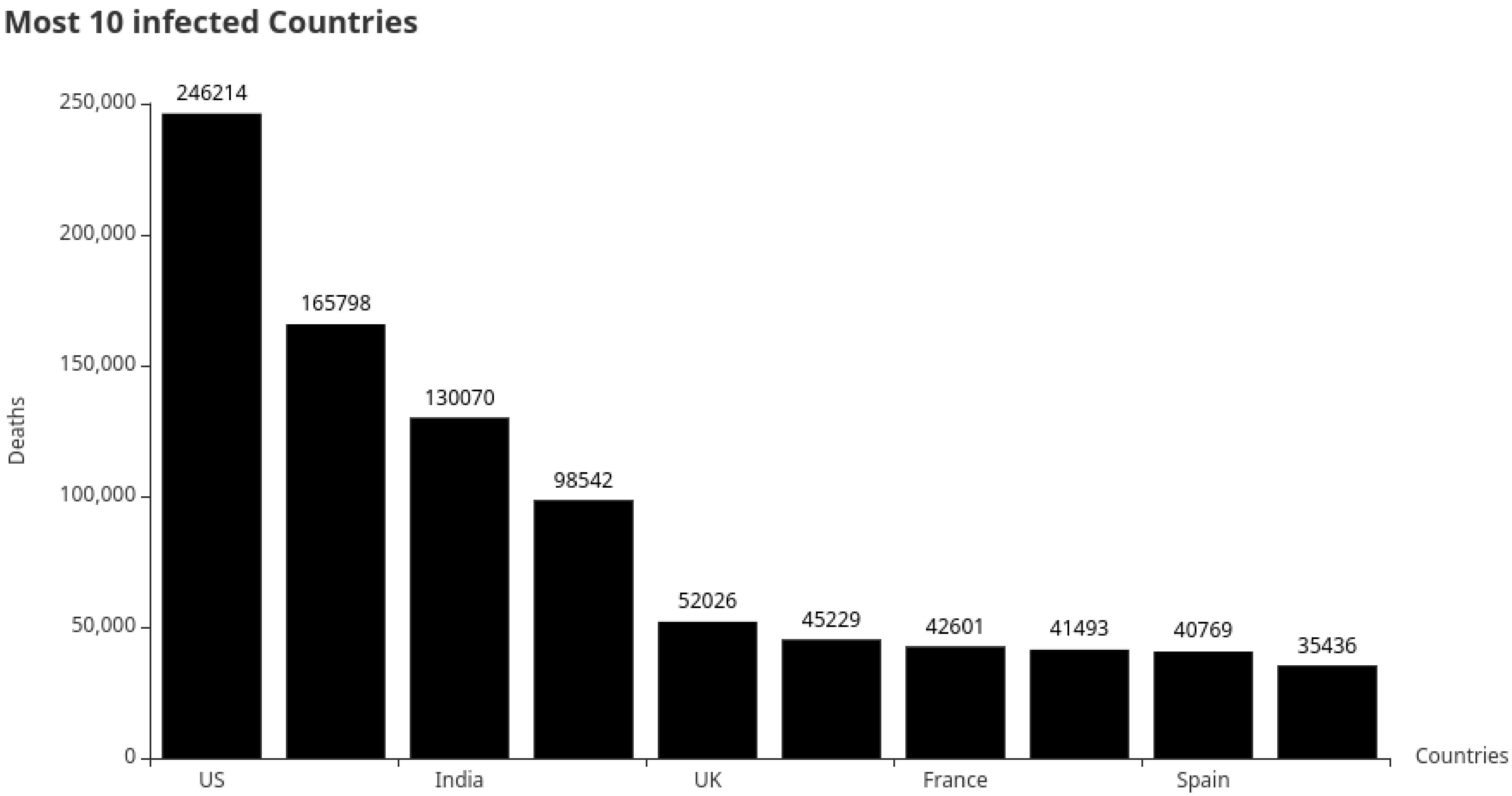
Country	Deaths
Papua New Guinea	7
Liechtenstein	5
Antigua and Barbuda	4
Brunei	3
Monaco	2
Fiji	2
Saint Lucia	2
MS Zaandam	2
Burundi	1
Western Sahara	1
Timor-Leste	0
Vanuatu	0
Mongolia	0
Seychelles	0
Macau	0
Holy See	0
Cambodia	0
Marshall Islands	0
Dominica	0
Saint Kitts and Nevis	0
Bhutan	0

Country	Deaths
Saint Vincent and the Grenadines	0
Grenada	0
Solomon Islands	0
Eritrea	0
Laos	0

In [57]:

```
(
    Bar(init_opts=opts.InitOpts())
    .set_global_opts(
        title_opts=opts.TitleOpts(title="Most 10 infected Countries"),
        xaxis_opts=opts.AxisOpts(type_='category', name="Countries",boundary_gap=["5%","5%"]),
        yaxis_opts=opts.AxisOpts(
            type_="value", name="Deaths",
            name_rotate=90,name_location="middle",name_gap=70
        ),
        tooltip_opts=opts.TooltipOpts(is_show=True),
    )
    .add_xaxis(Deaths_per_country['Country/Region'][0:10].values.tolist())
    .add_yaxis("", Deaths_per_country['Deaths'][0:10].values.tolist(),
               itemstyle_opts=opts.ItemStyleOpts(color="black"),
               label_opts=opts.LabelOpts(is_show=True),)
    .render_notebook()
)
```

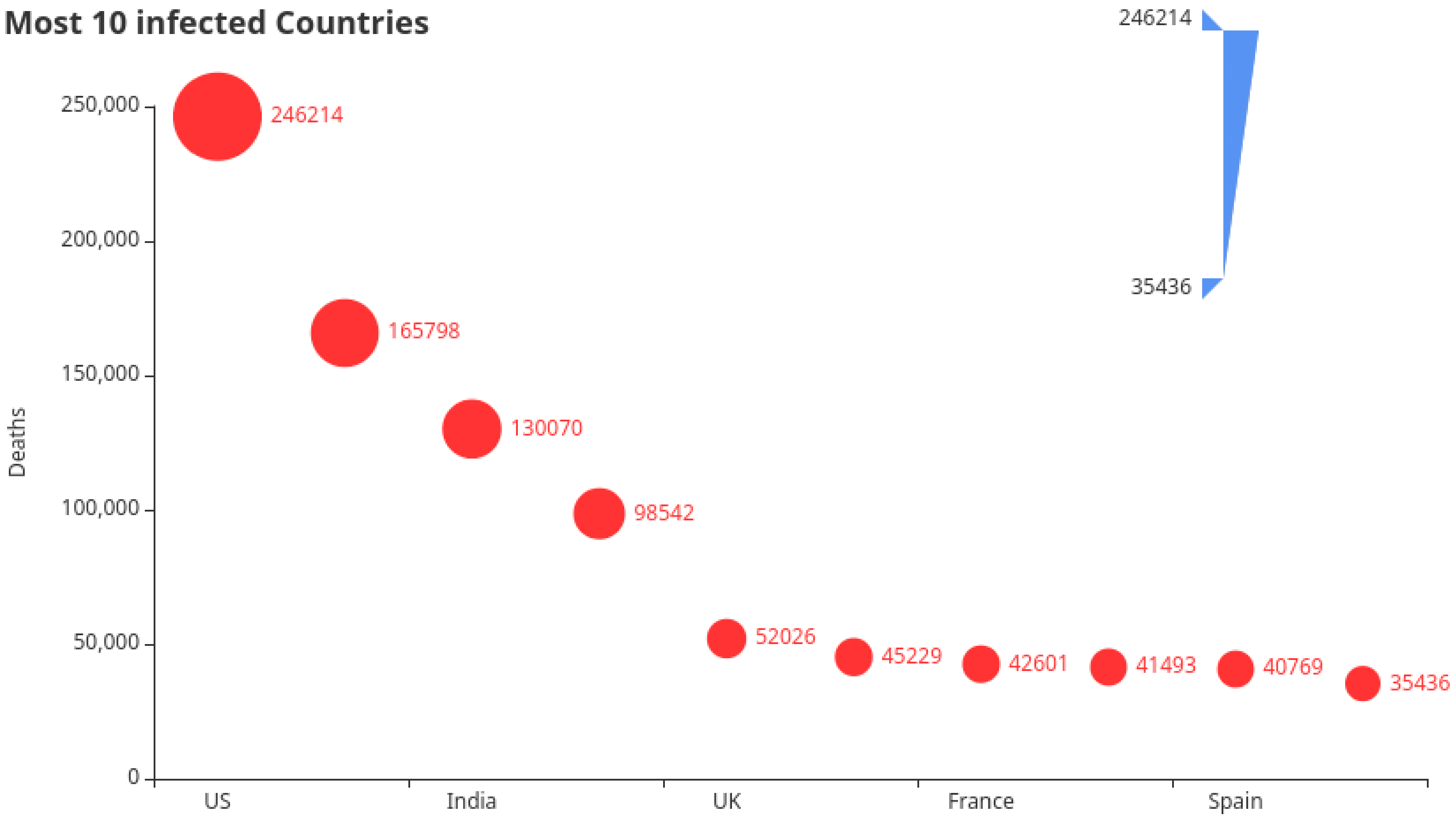
```
Out[57]:
```



Most 10 infected Countries
Deaths
250,000
200,000
150,000
100,000
50,000
0
246214
165798
130070
98542
52026
45229
42601
41493
40769
35436
US
India
UK
France
Spain
Countries

```
In [58]: (
             Scatter()
             .add_xaxis(Deaths_per_country['Country/Region'][0:10].values.tolist())
             .add_yaxis("", Deaths_per_country['Deaths'][0:10].values.tolist(),
                       itemstyle_opts=opts.ItemStyleOpts(color="red"))
             .set_global_opts(
                 title_opts=opts.TitleOpts(title="Most 10 infected Countries"),
                 visualmap_opts=opts.VisualMapOpts(type_="size",
                                                   min_=int(Deaths_per_country['Deaths'][0:10].min()),
                                                   max_=int(Deaths_per_country['Deaths'][0:10].max()),
                                                   orient="vertical",pos_right="20%",pos_top=0),
                 yaxis_opts=opts.AxisOpts(
                     type_="value", name="Deaths",
                     name_rotate=90,name_location="middle",name_gap=70
                 ),
             )
             .render_notebook()
         )
```

Out[58]:



Coronavirus in China

image.png

In [59]:
```
Data_China = data [(data['Country/Region'] == 'China') ].reset_index(drop=True)
Data_China.head()
```

Out[59]:

	SNo	ObservationDate	Province/State	Country/Region	Last Update	Confirmed	Deaths	Recovered	Active_case
0	1	01/22/2020	Anhui	China	1/22/2020 17:00	1	0	0	1
1	2	01/22/2020	Beijing	China	1/22/2020 17:00	14	0	0	14
2	3	01/22/2020	Chongqing	China	1/22/2020 17:00	6	0	0	6
3	4	01/22/2020	Fujian	China	1/22/2020 17:00	1	0	0	1
4	5	01/22/2020	Gansu	China	1/22/2020 17:00	0	0	0	0

Get last update in china

In [60]:
```
Data_china_last = Data_China[Data_China['ObservationDate'] == max(Data_China['ObservationDate'])].reset_index()
Data_china_last.head()
```

Out[60]:

	index	SNo	ObservationDate	Province/State	Country/Region	Last Update	Confirmed	Deaths	Recovered	Active_case
0	9236	155727	11/15/2020	Anhui	China	2020-11-16 05:25:57	992	6	985	1
1	9237	155761	11/15/2020	Beijing	China	2020-11-16 05:25:57	947	9	932	6
2	9238	155811	11/15/2020	Chongqing	China	2020-11-16 05:25:57	589	6	582	1
3	9239	155853	11/15/2020	Fujian	China	2020-11-16 05:25:57	461	1	426	34
4	9240	155858	11/15/2020	Gansu	China	2020-11-16 05:25:57	181	2	177	2

Confirmed cases in every Province/State in china

In [61]:

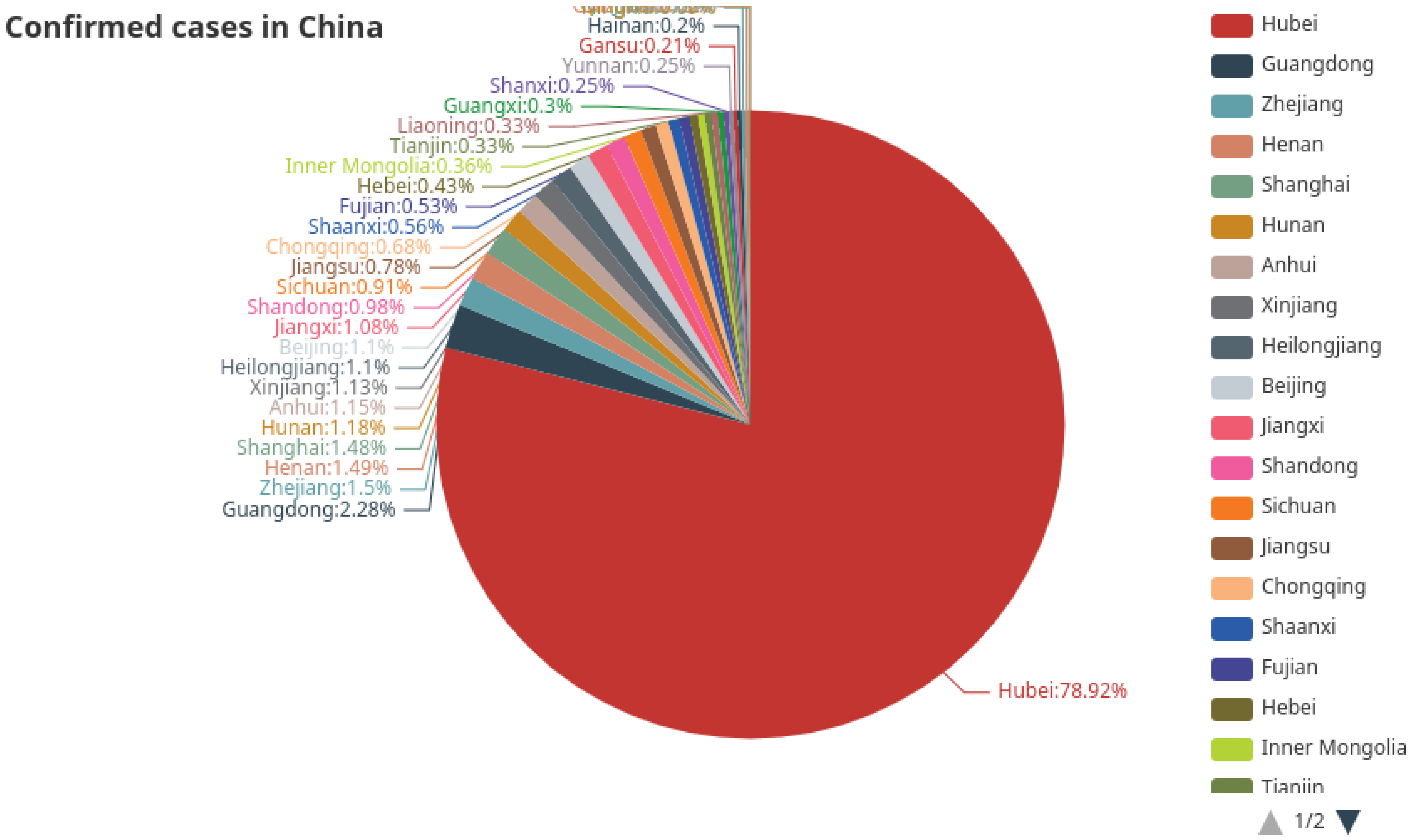
```
# 取得確診率
Data_china_per_state= Data_china_last.groupby(["Province/State"])["Confirmed","Active_case","Recovered","Deaths"].sum().reset_index().sort_values("Confirmed",ascending=False).reset_index(drop=True)
Data_china_per_state.head()
```

Out[61]:

	Province/State	Confirmed	Active_case	Recovered	Deaths
0	Hubei	68148	9	63627	4512
1	Guangdong	1968	37	1923	8
2	Zhejiang	1291	11	1279	1
3	Henan	1288	7	1259	22
4	Shanghai	1277	87	1183	7

```
In [62]: confirmed_all = Data_china_last["Confirmed"].sum()
         Data_china_per_state['Confirmed_per']=round(Data_china_per_state['Confirmed']/confirmed_all,4)
         (
             Pie()
             .add("Cases", [list(z) for z in zip(Data_china_per_state['Province/State'].values.tolist(),
                                                  Data_china_per_state['Confirmed_per'].values.tolist())])
             .set_global_opts(title_opts=opts.TitleOpts(title='Confirmed cases in China')
                             ,legend_opts=opts.LegendOpts(type_="scroll", orient="vertical"
                                                              , pos_left="80%", is_show=True,
                                                             ))
             .set_series_opts(label_opts=opts.LabelOpts(is_show=True,
                                                           formatter=JsCode(
                         "function(x){ { return x.name + ':' + Math.round(Number(x.value*10000))/100+'%';} }"
                     )))
             .render_notebook()
         )
```

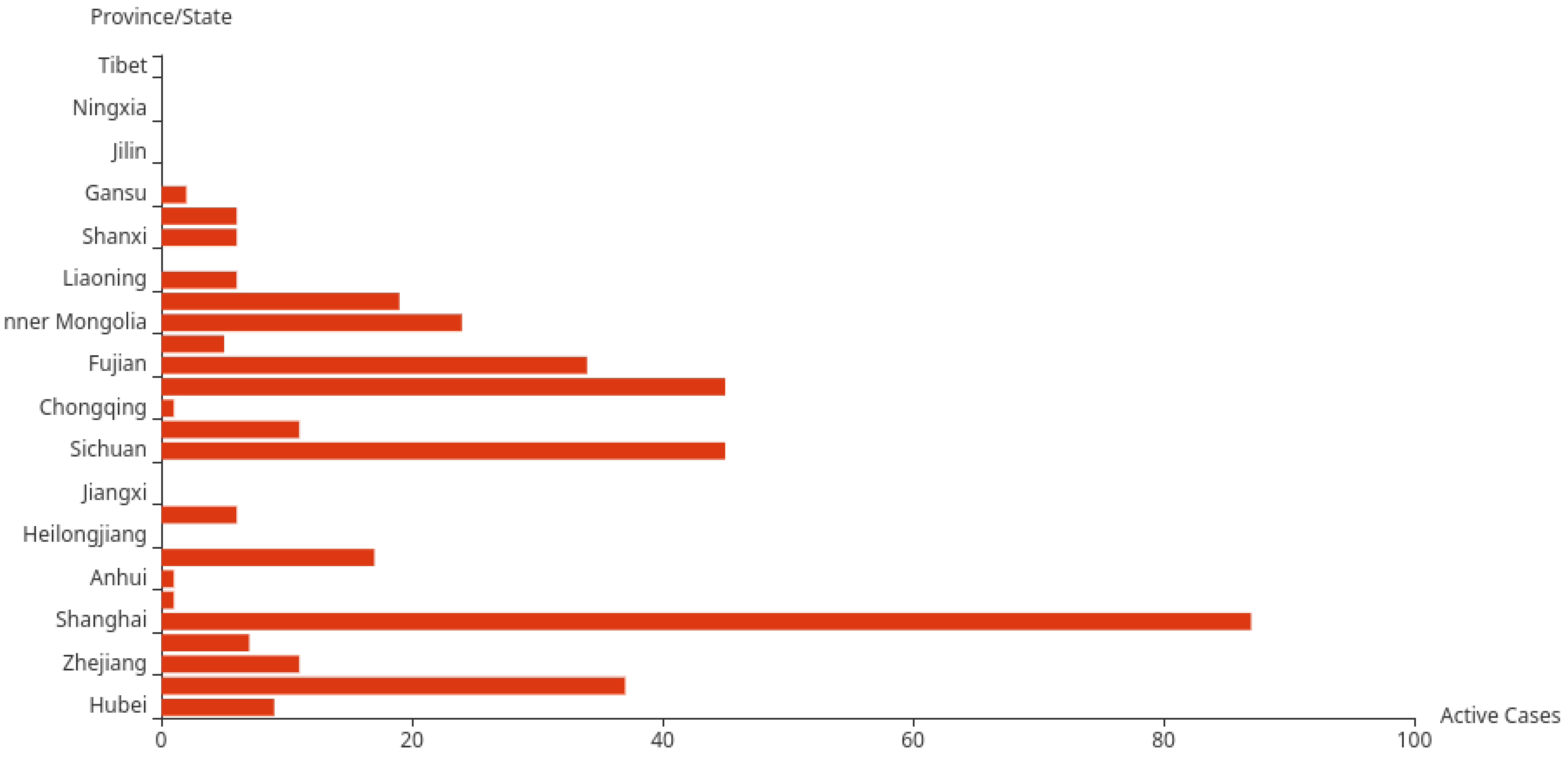
Out[62]:


Confirmed cases in China
Hainan:0.2%
Gansu:0.21%
Yunnan:0.25%
Shanxi:0.25%
Guangxi:0.3%
Liaoning:0.33%
Tianjin:0.33%
Inner Mongolia:0.36%
Hebei:0.43%
Fujian:0.53%
Shaanxi:0.56%
Chongqing:0.68%
Jiangsu:0.78%
Sichuan:0.91%
Shandong:0.98%
Jiangxi:1.08%
Beijing:1.1%
Heilongjiang:1.1%
Xinjiang:1.13%
Anhui:1.15%
Hunan:1.18%
Shanghai:1.48%
Henan:1.49%
Zhejiang:1.5%
Guangdong:2.28%
Hubei:78.92%
Hubei
Guangdong
Zhejiang
Henan
Shanghai
Hunan
Anhui
Xinjiang
Heilongjiang
Beijing
Jiangxi
Shandong
Sichuan
Jiangsu
Chongqing
Shaanxi
Fujian
Hebei
Inner Mongolia
Tianjin
1/2

```
In [63]: (
             Bar(init_opts=opts.InitOpts(theme=ThemeType.LIGHT))
             .add_xaxis(Data_china_per_state['Province/State'].values.tolist())
             .add_yaxis("", Data_china_per_state['Active_case'].values.tolist(),
                        label_opts=opts.LabelOpts(is_show=False),
                        itemstyle_opts=opts.ItemStyleOpts(color="#DC3912"))
             .reversal_axis()
             .set_global_opts(
                 title_opts=opts.TitleOpts(title="Active Cases In Each Province/State"),
                 tooltip_opts=opts.TooltipOpts(is_show=False),
                 xaxis_opts=opts.AxisOpts(type_='value', name="Active Cases"),
                 yaxis_opts=opts.AxisOpts(type_="category", name="Province/State"),
             )
             .render_notebook()
         )
```

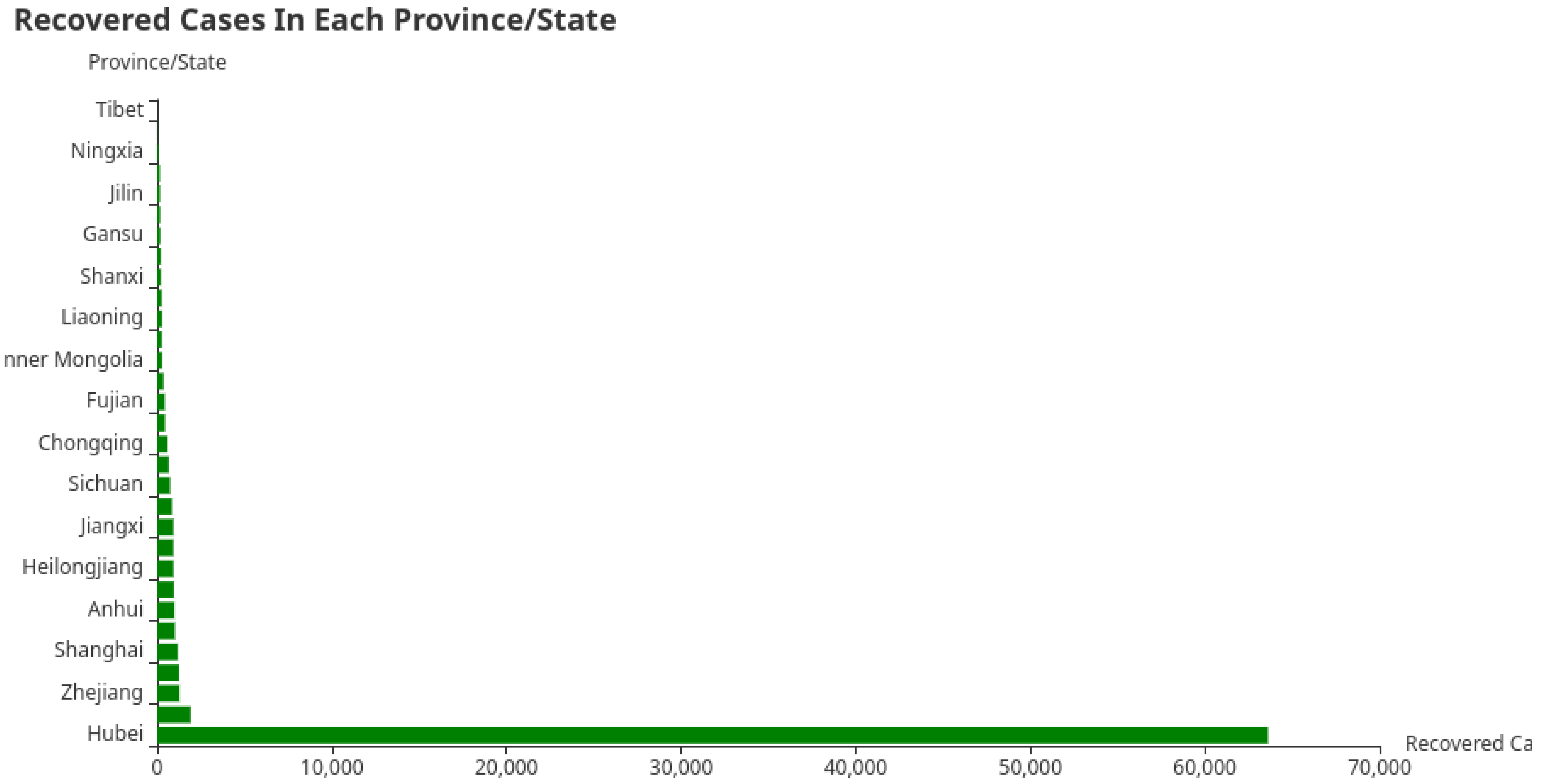
Out[63]:

Active Cases In Each Province/State



```
In [64]: (
             Bar(init_opts=opts.InitOpts(theme=ThemeType.LIGHT))
             .add_xaxis(Data_china_per_state['Province/State'].values.tolist())
             .add_yaxis("", Data_china_per_state['Recovered'].values.tolist(),
                         label_opts=opts.LabelOpts(is_show=False),
                         itemstyle_opts=opts.ItemStyleOpts(color="green"))
             .reversal_axis()
             .set_global_opts(
                 title_opts=opts.TitleOpts(title="Recovered Cases In Each Province/State"),
                 tooltip_opts=opts.TooltipOpts(is_show=False),
                 xaxis_opts=opts.AxisOpts(type_='value', name="Recovered Cases"),
                 yaxis_opts=opts.AxisOpts(type_="category", name="Province/State"),
             )
             .render_notebook()
         )
```

Out[64]:



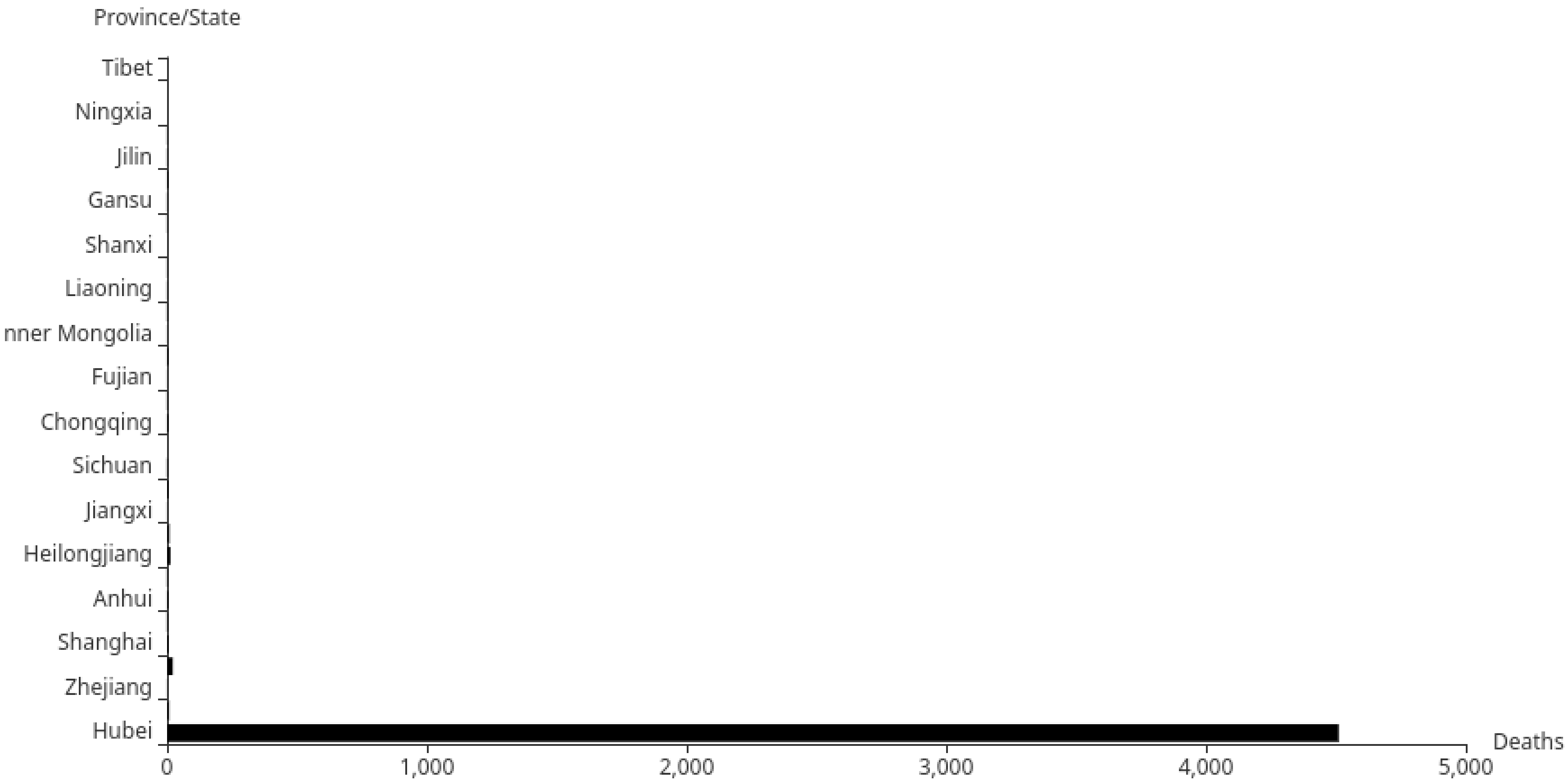
Recovered Cases In Each Province/State

In [65]:

```
(
    Bar(init_opts=opts.InitOpts(theme=ThemeType.LIGHT))
    .add_xaxis(Data_china_per_state['Province/State'].values.tolist())
    .add_yaxis("", Data_china_per_state['Deaths'].values.tolist(),
               label_opts=opts.LabelOpts(is_show=False),
               itemstyle_opts=opts.ItemStyleOpts(color="black"))
    .reversal_axis()
    .set_global_opts(
        title_opts=opts.TitleOpts(title="Deaths In Each Province/State"),
        tooltip_opts=opts.TooltipOpts(is_show=False),
        xaxis_opts=opts.AxisOpts(type_='value', name="Deaths"),
        yaxis_opts=opts.AxisOpts(type_="category", name="Province/State"),
    )
    .render_notebook()
)
```

Out[65]:

Deaths In Each Province/State



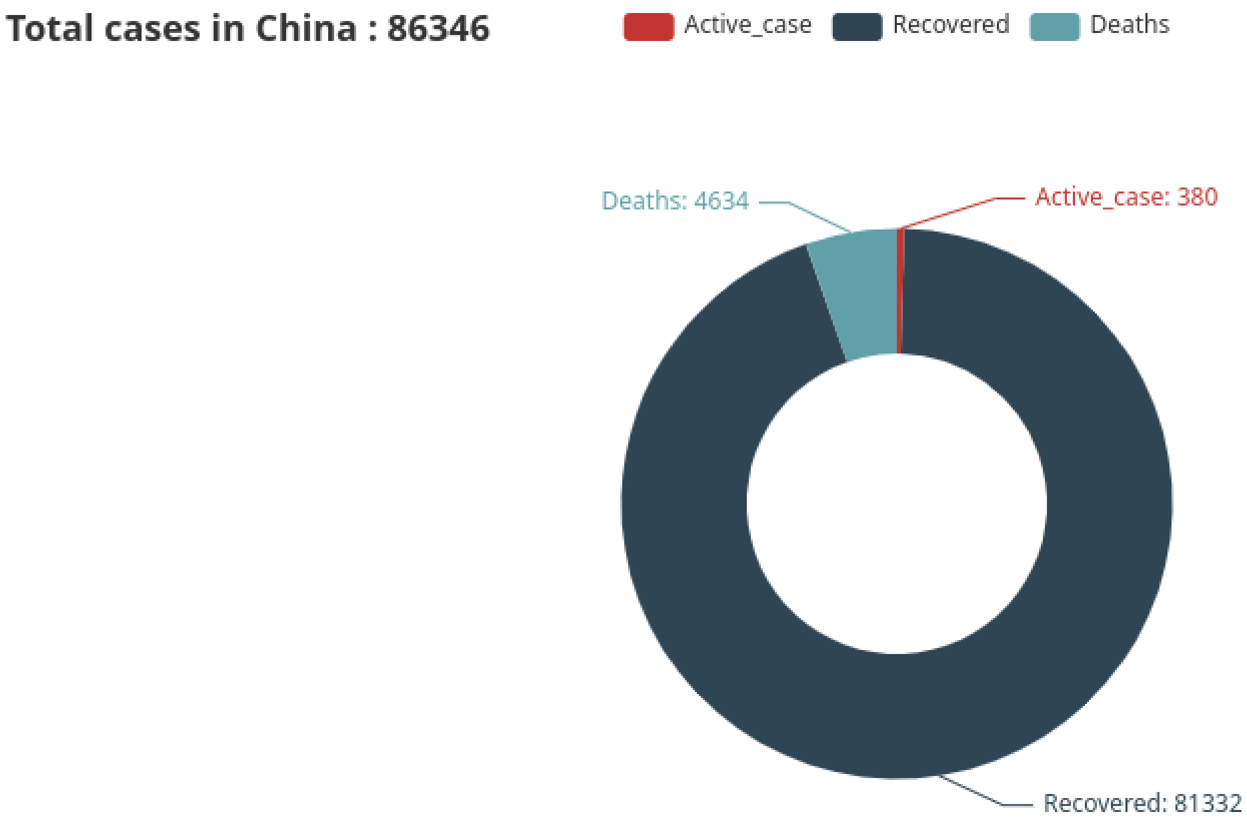
```
In [66]: Data_china_total= Data_china_last.groupby(["Country/Region"])["Confirmed","Deaths","Recovered","Active_case"].sum().
         reset_index().reset_index(drop=True)
         Data_china_total.head()
```

Out[66]:

	Country/Region	Confirmed	Deaths	Recovered	Active_case
0	China	86346	4634	81332	380

```
In [67]: labels = ["Active cases","Recovered","Deaths"]
         idxies = ["Active_case","Recovered","Deaths"]
         data_pair = [ [z[0],int(z[1])] for z in zip(idxies,
                                                    Data_china_total.loc[0, ["Active_case","Recovered","Deaths"]].values.tol
         ist()) ]
         (
             Pie()
             .add("Cases", data_pair, radius=["30%", "55%"])
             .set_global_opts(title_opts=opts.TitleOpts(title='Total cases in China : '+str(Data_china_total["Confirmed"][0
         ])))
             .set_series_opts(label_opts=opts.LabelOpts(formatter="{b}: {c}"))
             .render_notebook()
         )
```

Out[67]:



Evolution of cases over time in China

In [68]:

```
Data_china_op= Data_China.groupby(["ObservationDate","Country/Region"])["Confirmed","Deaths","Recovered","Active_case"].sum().reset_index().reset_index(drop=True)
Data_china_op.head()
```

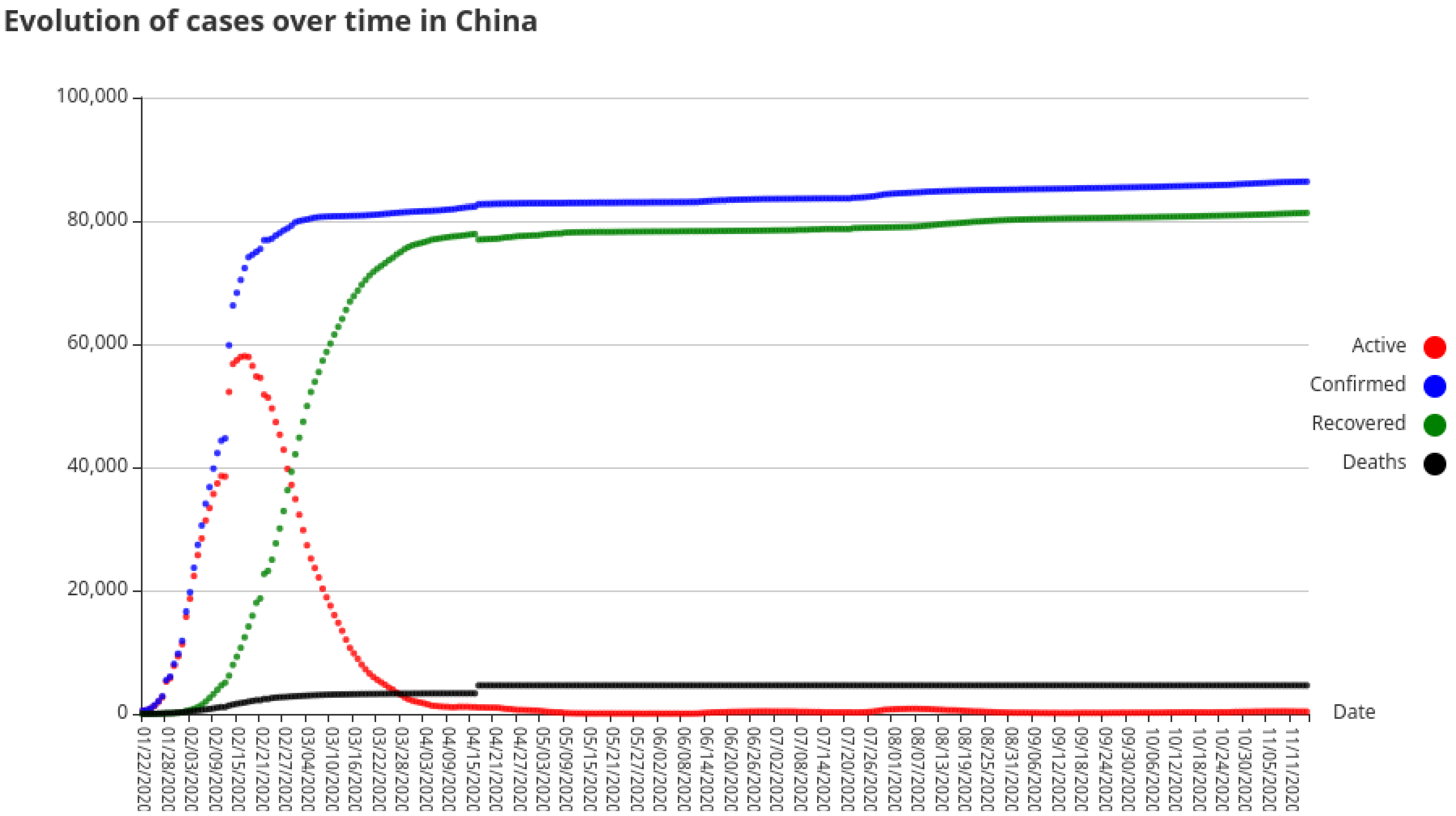
Out[68]:

	ObservationDate	Country/Region	Confirmed	Deaths	Recovered	Active_case
0	01/22/2020	China	547	17	28	502
1	01/23/2020	China	639	18	30	591
2	01/24/2020	China	916	26	36	854
3	01/25/2020	China	1399	42	39	1318
4	01/26/2020	China	2062	56	49	1957

In [69]:

```
(
    Scatter()
    .set_global_opts(
        title_opts=opts.TitleOpts(title="Evolution of cases over time in China"),
        xaxis_opts=opts.AxisOpts(type_='category', name="Date"
                                 ,axislabel_opts=opts.LabelOpts(rotate=-90,font_size="10")),
        yaxis_opts=opts.AxisOpts(
            type_="value",
            axistick_opts=opts.AxisTickOpts(is_show=True),
            splitline_opts=opts.SplitLineOpts(is_show=True),
        ),
        legend_opts=opts.LegendOpts(pos_left="right",orient="vertical",pos_top="middle")
    )
    .add_xaxis(Data_china_op['ObservationDate'].tolist())
    .add_yaxis(
        series_name="Active",
        y_axis=Data_china_op['Active_case'].tolist(),
        symbol_size=4,
        label_opts=opts.LabelOpts(is_show=False),
        itemstyle_opts=opts.ItemStyleOpts(color="red"),
    )
    .add_yaxis(
        series_name="Confirmed",
        y_axis=Data_china_op['Confirmed'].tolist(),
        symbol_size=4,
        label_opts=opts.LabelOpts(is_show=False),
        itemstyle_opts=opts.ItemStyleOpts(color="blue"),
    )
    .add_yaxis(
        series_name="Recovered",
        y_axis=Data_china_op['Recovered'].tolist(),
        symbol_size=4,
        label_opts=opts.LabelOpts(is_show=False),
        itemstyle_opts=opts.ItemStyleOpts(color="green"),
    )
    .add_yaxis(
        series_name="Deaths",
        y_axis=Data_china_op['Deaths'].tolist(),
        symbol_size=4,
        label_opts=opts.LabelOpts(is_show=False),
        itemstyle_opts=opts.ItemStyleOpts(color="black"),
    )
    .render_notebook()
)
```

Out[69]:



Coronavirus in United States

image.png

```
In [70]: Data_US = data [(data['Country/Region'] == 'US') ].reset_index(drop=True)
```

```
In [71]: Data_us_last = Data_US[Data_US['ObservationDate'] == max(Data_US['ObservationDate'])].reset_index()
```

In [72]:

```
Data_us_total= Data_us_last.groupby(["Country/Region"])["Confirmed","Deaths","Recovered","Active_case"].sum().reset_index().reset_index(drop=True)
Data_us_total.head()
```

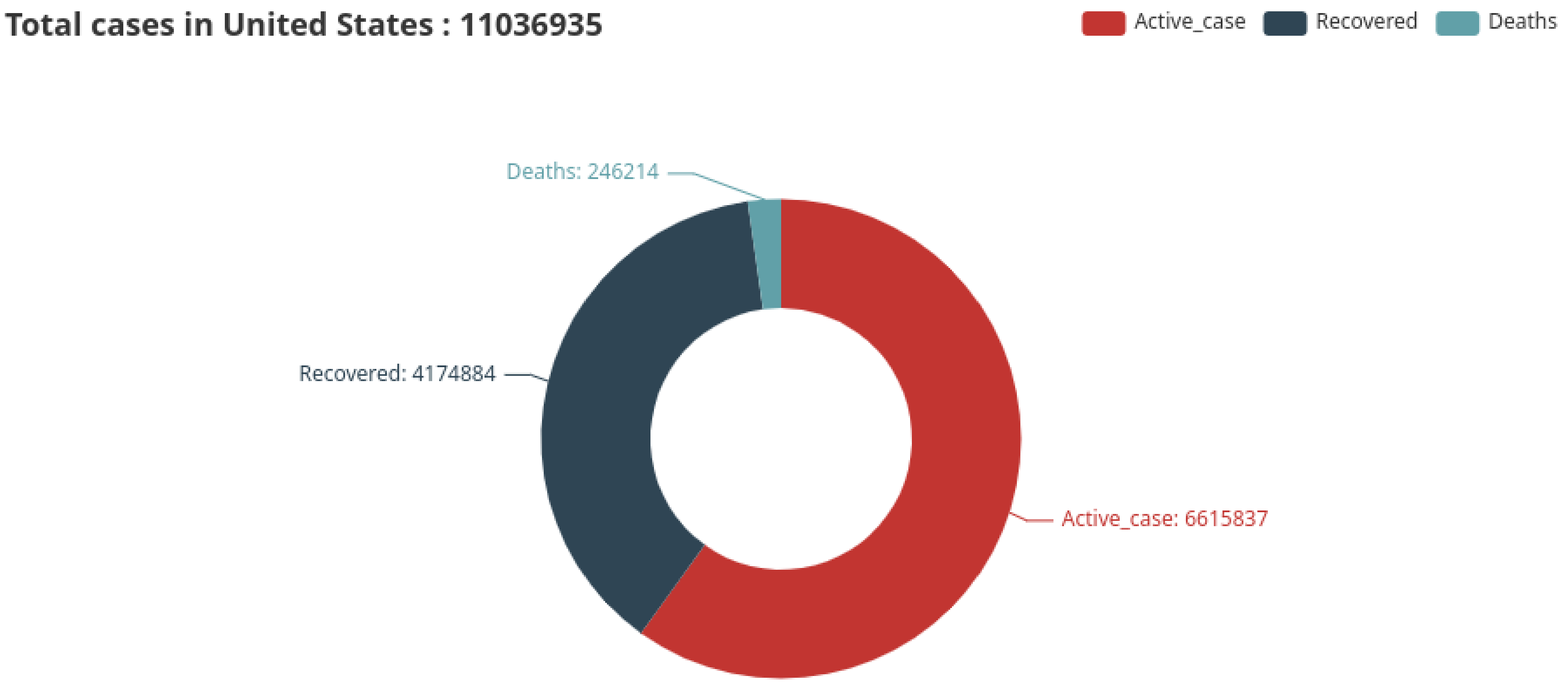
Out[72]:

	Country/Region	Confirmed	Deaths	Recovered	Active_case
0	US	11036935	246214	4174884	6615837

In [73]:

```
labels = ["Active cases","Recovered","Deaths"]
idxies = ["Active_case","Recovered","Deaths"]
data_pair = [ [z[0],int(z[1])] for z in zip(idxies,
                                            Data_us_total.loc[0, ["Active_case","Recovered","Deaths"]].values.tolist
()) ]
(
    Pie()
    .add("Cases", data_pair, radius=["30%", "55%"])
    .set_global_opts(title_opts=opts.TitleOpts(title='Total cases in United States : '+str(Data_us_total["Confirmed"
][0])),
                    legend_opts=opts.LegendOpts(pos_left="right"))
    .set_series_opts(label_opts=opts.LabelOpts(formatter="{b}: {c}"))
    .render_notebook()
)
```

Out[73]:



Cases in every Province/State in US

In [74]:

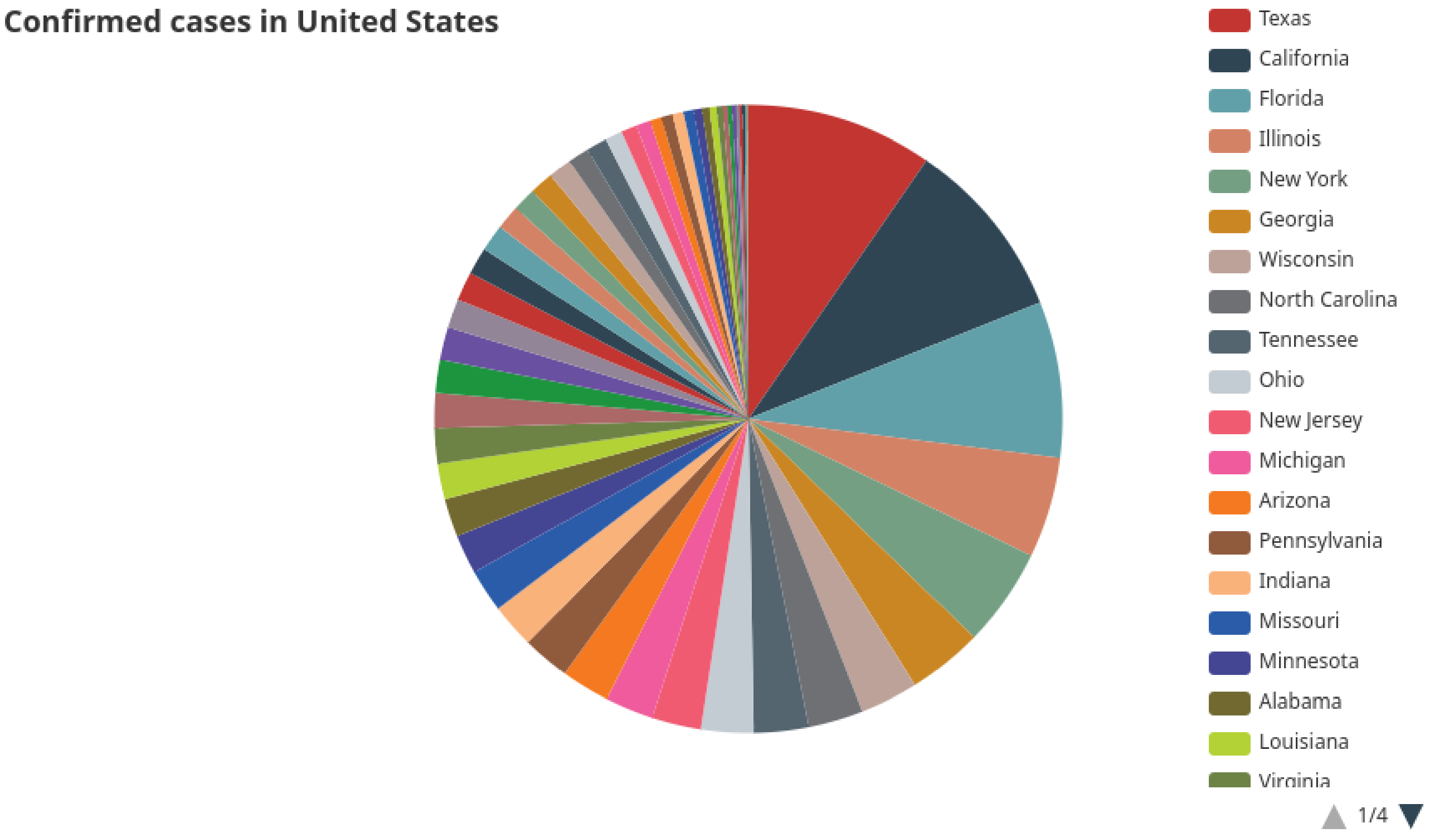
```
Data_us_per_state= Data_us_last.groupby(["Province/State"])["Confirmed","Active_case","Deaths"].sum().reset_index().
sort_values("Confirmed",ascending=False).reset_index(drop=True)
Data_us_per_state.head()
```

Out[74]:

	Province/State	Confirmed	Active_case	Deaths
0	Texas	1059753	1039751	20002
1	California	1032095	1013833	18262
2	Florida	885201	867683	17518
3	Illinois	573616	562454	11162
4	New York	560200	526168	34032

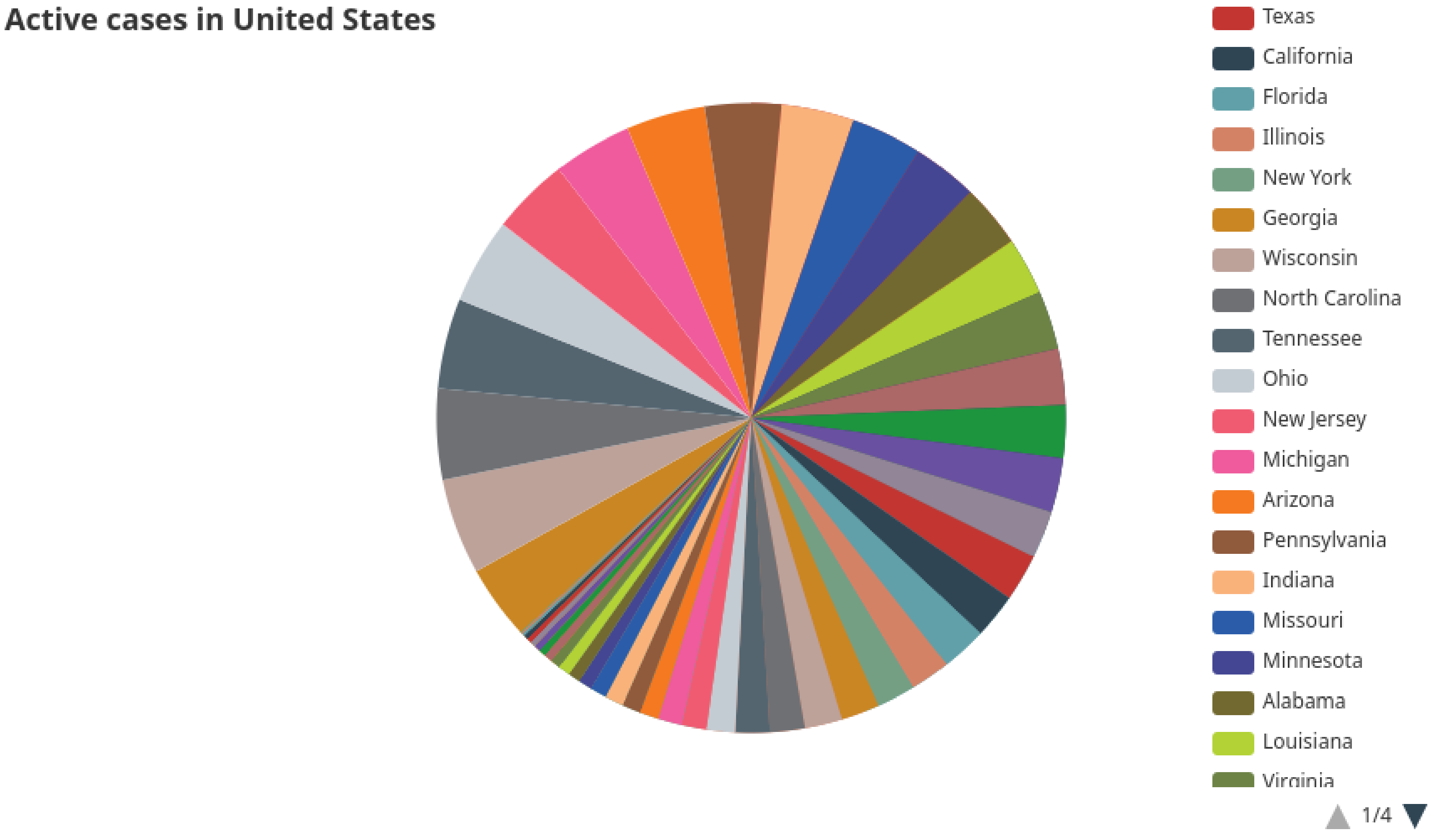
```
In [75]: (
             Pie()
             .add("Confirmed", [list(z) for z in zip(Data_us_per_state['Province/State'].values.tolist(),
                                                  Data_us_per_state['Confirmed'].values.tolist())])
             .set_global_opts(title_opts=opts.TitleOpts(title='Confirmed cases in United States')
                             ,legend_opts=opts.LegendOpts(type_="scroll", orient="vertical"
                                                          , pos_left="80%", is_show=True,
                                                         ))
             .set_series_opts(label_opts=opts.LabelOpts(is_show=False,
                                                      formatter=JsCode(
                         "function(x){ if(x.value>0.1){ return x.name + ':' + Number(x.value);} else {return '';} }"
                     )))
             .render_notebook()
         )
```

Out[75]:



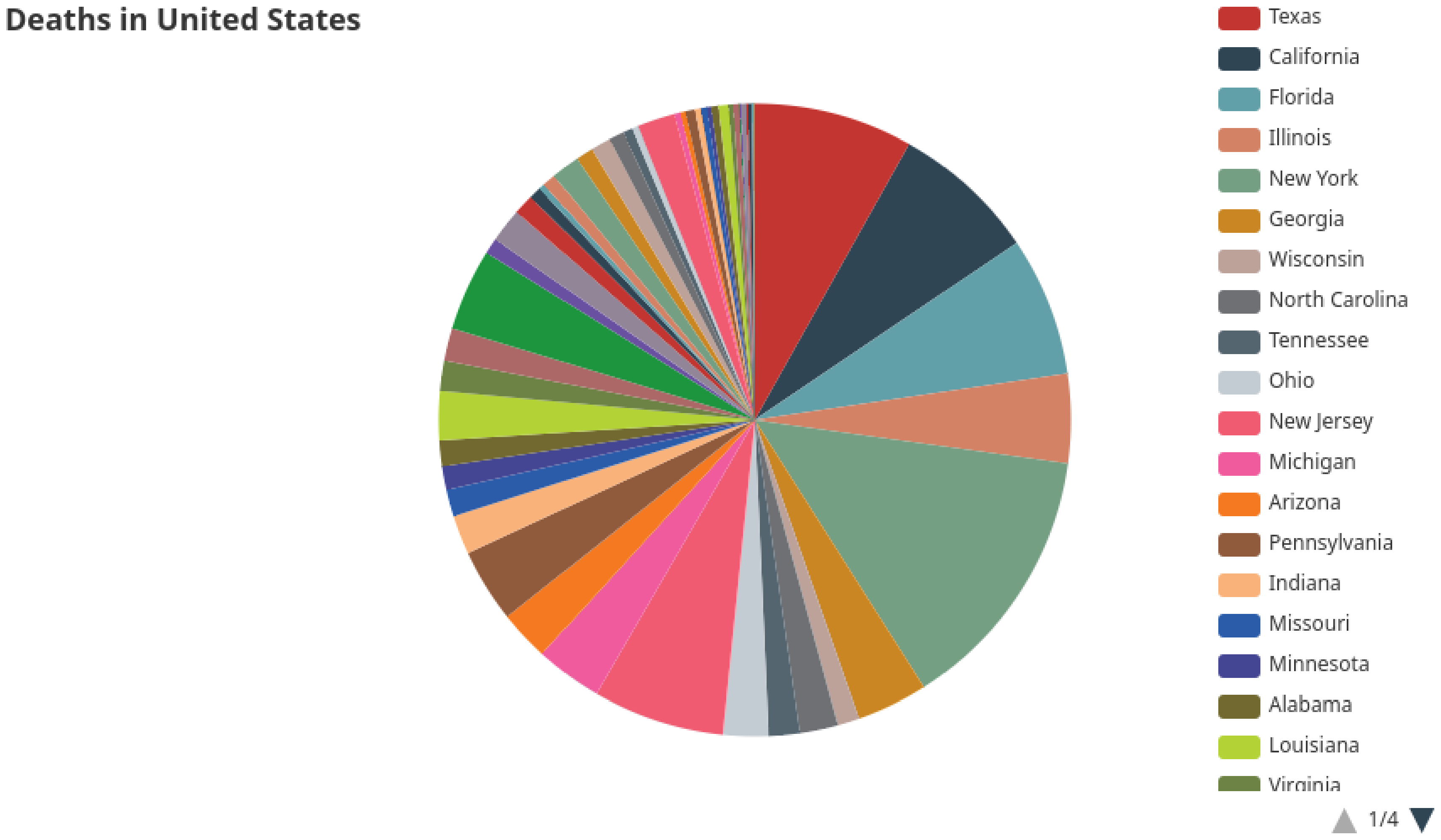
```
In [76]: (
             Pie()
             .add("Active", [list(z) for z in zip(Data_us_per_state['Province/State'].values.tolist(),
                                                  Data_us_per_state['Active_case'].values.tolist())])
             .set_global_opts(title_opts=opts.TitleOpts(title='Active cases in United States')
                              ,legend_opts=opts.LegendOpts(type_="scroll", orient="vertical"
                                                           , pos_left="80%", is_show=True,
                                                          ))
             .set_series_opts(label_opts=opts.LabelOpts(is_show=False,
                                                        formatter=JsCode(
                         "function(x){ if(x.value>0.1){ return x.name + ':' + Number(x.value);} else {return '';} }"
                     )))
             .render_notebook()
         )
```

Out[76]:



```
In [77]: (
             Pie()
             .add("Deaths", [list(z) for z in zip(Data_us_per_state['Province/State'].values.tolist(),
                                                   Data_us_per_state['Deaths'].values.tolist())])
             .set_global_opts(title_opts=opts.TitleOpts(title='Deaths in United States')
                              ,legend_opts=opts.LegendOpts(type_="scroll", orient="vertical"
                                                           , pos_left="80%", is_show=True,
                                                          ))
             .set_series_opts(label_opts=opts.LabelOpts(is_show=False,
                                                        formatter=JsCode(
                         "function(x){ if(x.value>0.1){ return x.name + ':' + Number(x.value);} else {return '';} }"
                     )))
             .render_notebook()
         )
```

Out[77]:



In [78]:

```
Data_US_op= Data_US.groupby(["ObservationDate","Country/Region"])["Confirmed","Deaths","Recovered","Active_case"].su
m().reset_index().reset_index(drop=True)
Data_US_op.head()
```

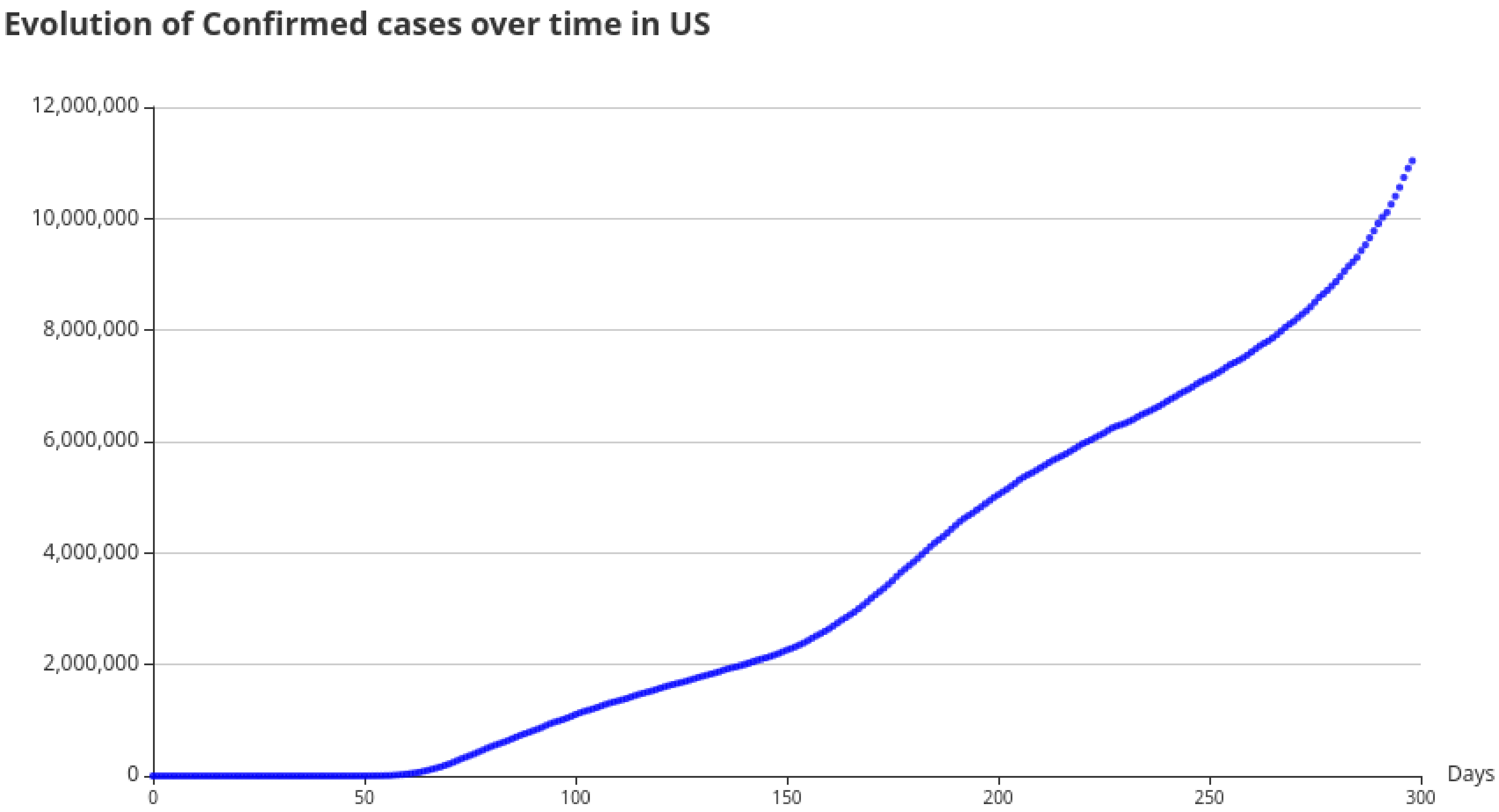
Out[78]:

	ObservationDate	Country/Region	Confirmed	Deaths	Recovered	Active_case
0	01/22/2020	US	1	0	0	1
1	01/23/2020	US	1	0	0	1
2	01/24/2020	US	2	0	0	2
3	01/25/2020	US	2	0	0	2
4	01/26/2020	US	5	0	0	5

In [79]:

```
(
    Scatter()
    .set_global_opts(
        title_opts=opts.TitleOpts(title="Evolution of Confirmed cases over time in US"),
        xaxis_opts=opts.AxisOpts(type_='value', name="Days"
                                ,axislabel_opts=opts.LabelOpts(font_size="10")),
        yaxis_opts=opts.AxisOpts(
            type_="value",
            axistick_opts=opts.AxisTickOpts(is_show=True),
            splitline_opts=opts.SplitLineOpts(is_show=True),
        ),
        legend_opts=opts.LegendOpts(is_show=False)
    )
    .add_xaxis(Data_US_op.index.tolist())
    .add_yaxis(
        series_name="",
        y_axis=Data_US_op['Confirmed'].tolist(),
        symbol_size=4,
        label_opts=opts.LabelOpts(is_show=False),
        itemstyle_opts=opts.ItemStyleOpts(color="blue"),

    )
    .render_notebook()
)
```

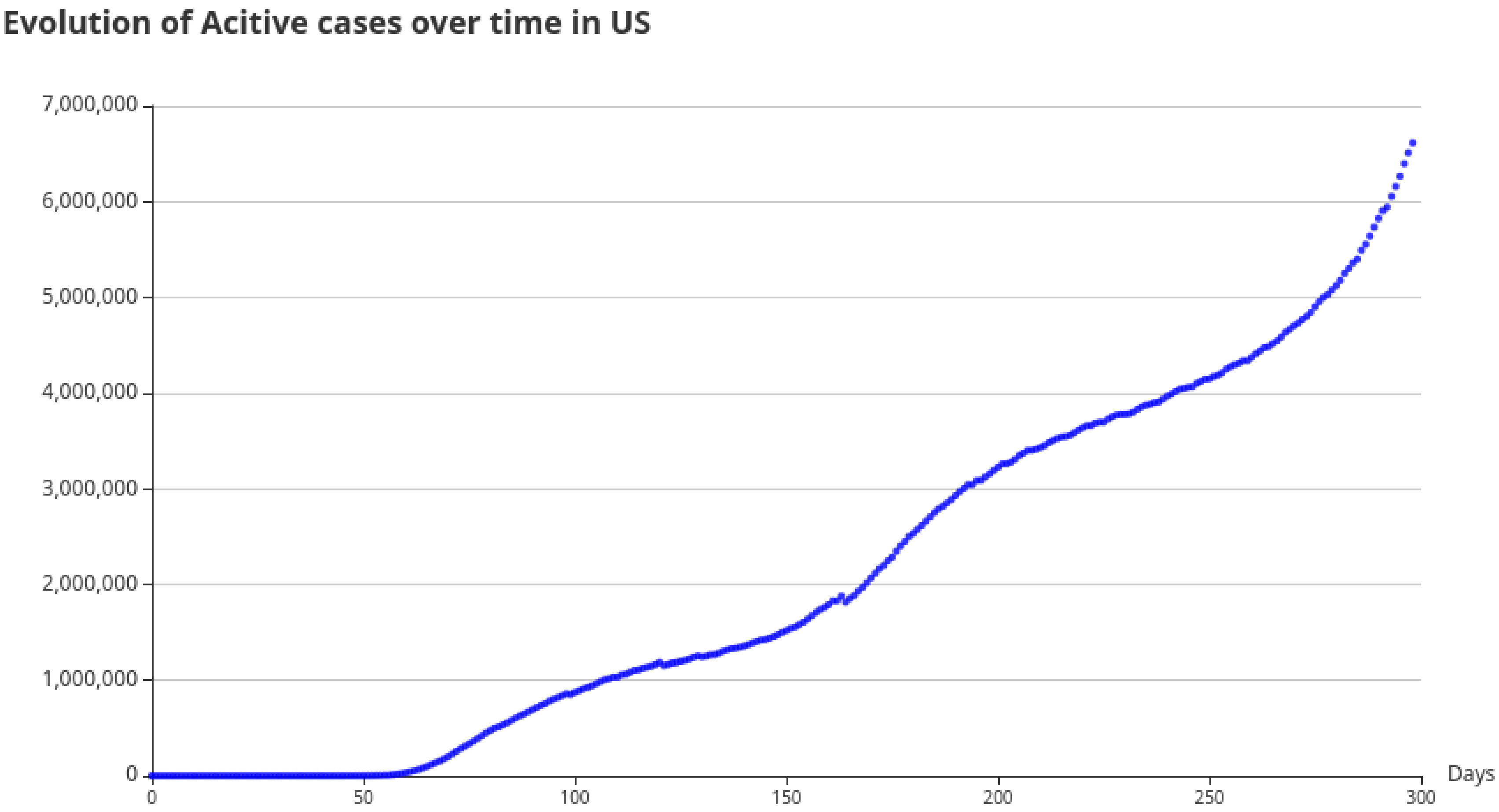
```
Out[79]:
```



Evolution of Confirmed cases over time in US
12,000,000
10,000,000
8,000,000
6,000,000
4,000,000
2,000,000
0
0
50
100
150
200
250
300
Days

```
In [80]: (
    Scatter()
    .set_global_opts(
        title_opts=opts.TitleOpts(title="Evolution of Acitive cases over time in US"),
        xaxis_opts=opts.AxisOpts(type_='value', name="Days"
                                 ,axislabel_opts=opts.LabelOpts(font_size="10")),
        yaxis_opts=opts.AxisOpts(
            type_="value",
            axistick_opts=opts.AxisTickOpts(is_show=True),
            splitline_opts=opts.SplitLineOpts(is_show=True),
        ),
        legend_opts=opts.LegendOpts(is_show=False)
    )
    .add_xaxis(Data_US_op.index.tolist())
    .add_yaxis(
        series_name="",
        y_axis=Data_US_op['Active_case'].tolist(),
        symbol_size=4,
        label_opts=opts.LabelOpts(is_show=False),
        itemstyle_opts=opts.ItemStyleOpts(color="blue"),

    )
    .render_notebook()
)
```

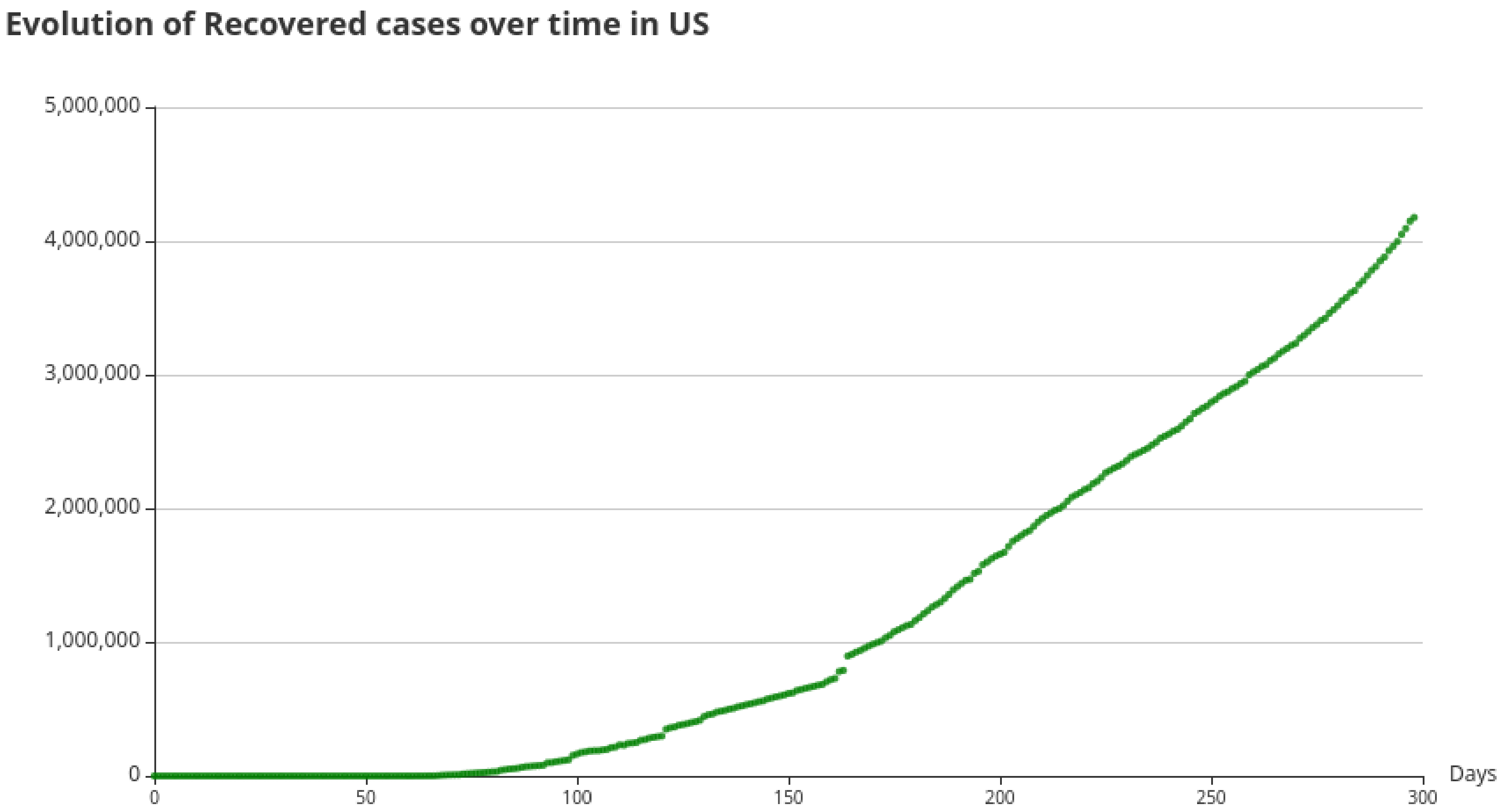
```
Out[80]:
```


Evolution of Acitive cases over time in US
7,000,000
6,000,000
5,000,000
4,000,000
3,000,000
2,000,000
1,000,000
0
0
50
100
150
200
250
300
Days

In [81]:

```
(
    Scatter()
    .set_global_opts(
        title_opts=opts.TitleOpts(title="Evolution of Recovered cases over time in US"),
        xaxis_opts=opts.AxisOpts(type_='value', name="Days"
                                 ,axislabel_opts=opts.LabelOpts(font_size="10")),
        yaxis_opts=opts.AxisOpts(
            type_="value",
            axistick_opts=opts.AxisTickOpts(is_show=True),
            splitline_opts=opts.SplitLineOpts(is_show=True),
        ),
        legend_opts=opts.LegendOpts(is_show=False)
    )
    .add_xaxis(Data_US_op.index.tolist())
    .add_yaxis(
        series_name="",
        y_axis=Data_US_op['Recovered'].tolist(),
        symbol_size=4,
        label_opts=opts.LabelOpts(is_show=False),
        itemstyle_opts=opts.ItemStyleOpts(color="green"),

    )
    .render_notebook()
)
```

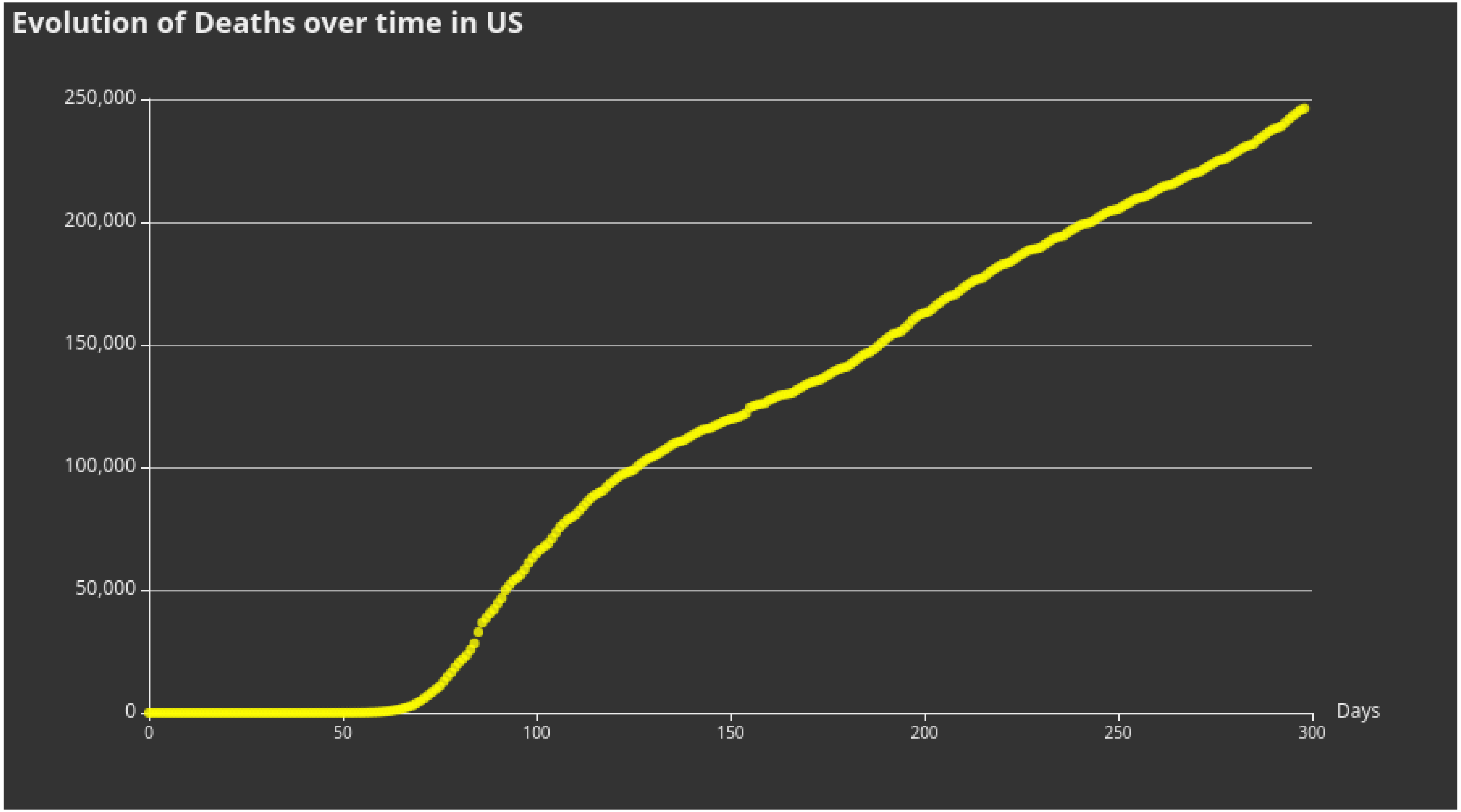
```
Out[81]:
```


Evolution of Recovered cases over time in US
5,000,000
4,000,000
3,000,000
2,000,000
1,000,000
0
0
50
100
150
200
250
300
Days

In [85]:

```
(
    Scatter(init_opts=opts.InitOpts(theme=ThemeType.DARK))
    .set_global_opts(
        title_opts=opts.TitleOpts(title="Evolution of Deaths over time in US"),
        xaxis_opts=opts.AxisOpts(type_='value', name="Days"
                                 ,axislabel_opts=opts.LabelOpts(font_size="10")),
        yaxis_opts=opts.AxisOpts(
            type_="value",
            axistick_opts=opts.AxisTickOpts(is_show=True),
            splitline_opts=opts.SplitLineOpts(is_show=True),
        ),
        legend_opts=opts.LegendOpts(is_show=False)
    )
    .add_xaxis(Data_US_op.index.tolist())
    .add_yaxis(
        series_name="",
        y_axis=Data_US_op['Deaths'].tolist(),
        symbol_size=6,
        label_opts=opts.LabelOpts(is_show=False),
        itemstyle_opts=opts.ItemStyleOpts(color="yellow"),

    )
    .render_notebook()
)
```

```
Out[85]:
```


Evolution of Deaths over time in US
250,000
200,000
150,000
100,000
50,000
0
0
50
100
150
200
250
300
Days