

In [1]:

```
import pandas as pd #Data analysis and manipulation
import matplotlib.pyplot as plt #Data visualization
import plotly.offline as py #creates functions both online and offline mode
import plotly.graph_objs as go #tracing objects
import plotly.express as px #easier and faster to create plotly figures
import plotly.io as pio #display the figure using the current default renderer(s)
import csv
```

In [2]:

```
df= pd.read_csv("time-series-19-covid-combined.csv")
```

In [3]:

```
df.columns
```

Out[3]:

```
Index(['Date', 'Country/Region', 'Province/State', 'Lat', 'Long', 'Confirmed',
       'Recovered', 'Deaths'],
      dtype='object')
```

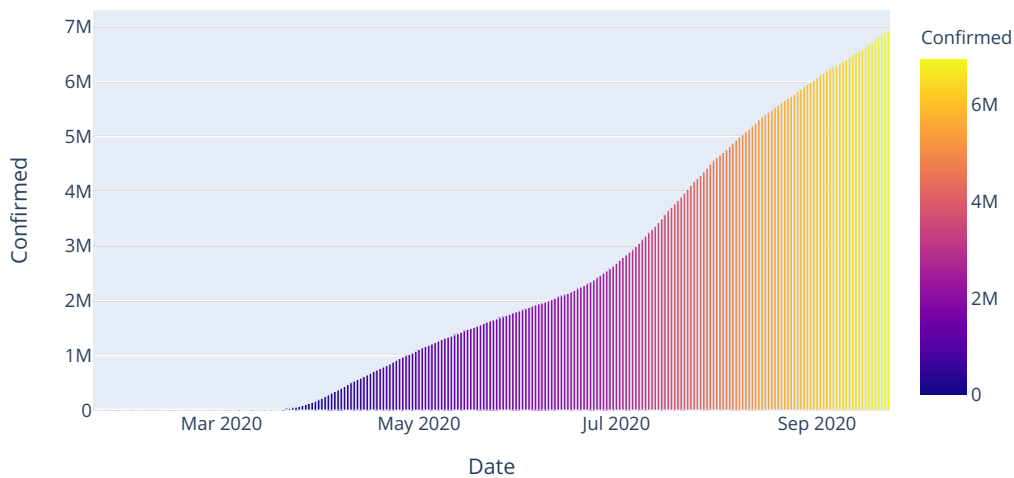
Date vs Confirmed

In [4]:

```
df_US = df.loc[df["Country/Region"]=="US"]
```

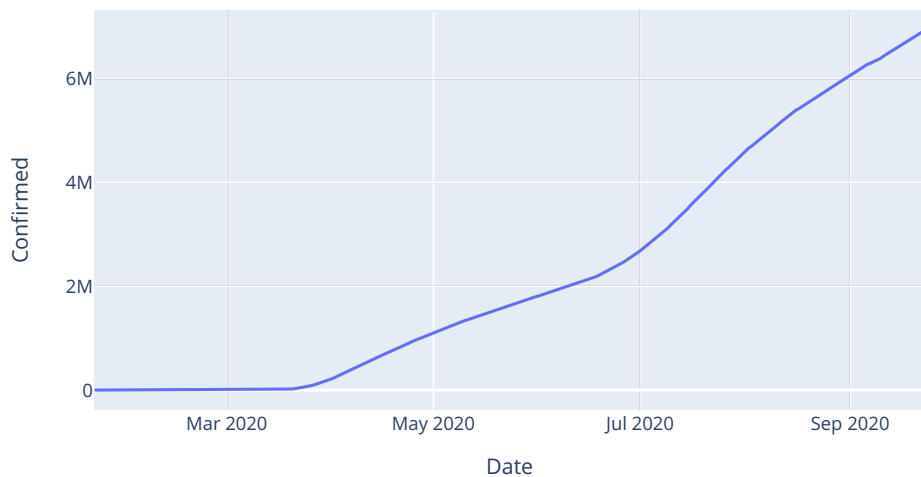
In [5]:

```
px.bar(df_US, x="Date", y="Confirmed", color="Confirmed", height=400)
```



In [6]:

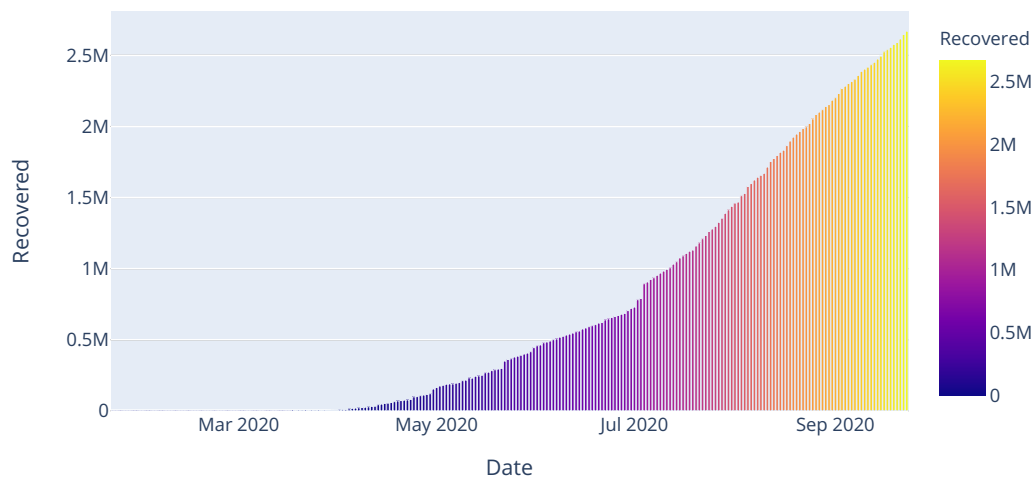
```
px.line(df_US, x="Date", y="Confirmed", height=400)
```



Date vs Recovered

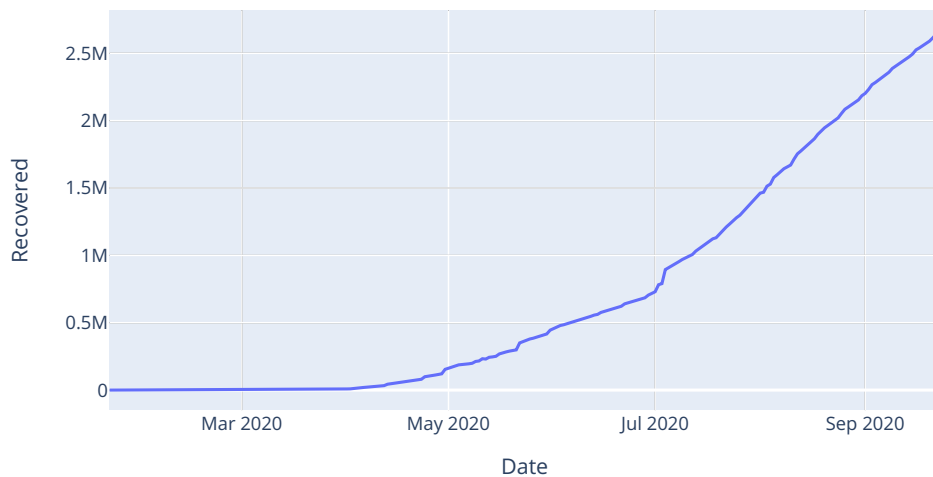
In [7]:

```
px.bar(df_US, x="Date", y="Recovered", color="Recovered", height=400)
```



In [8]:

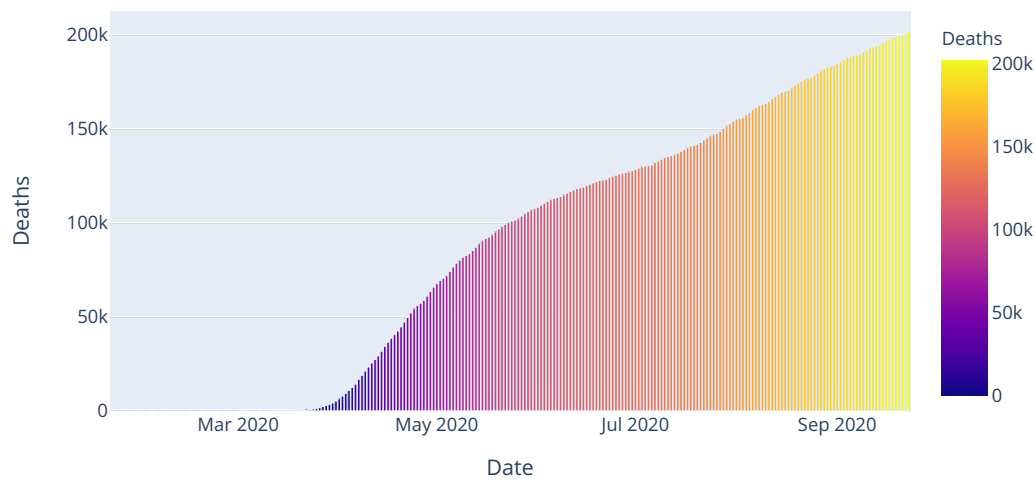
```
px.line(df_US, x="Date", y="Recovered", height=400)
```



Date vs Death

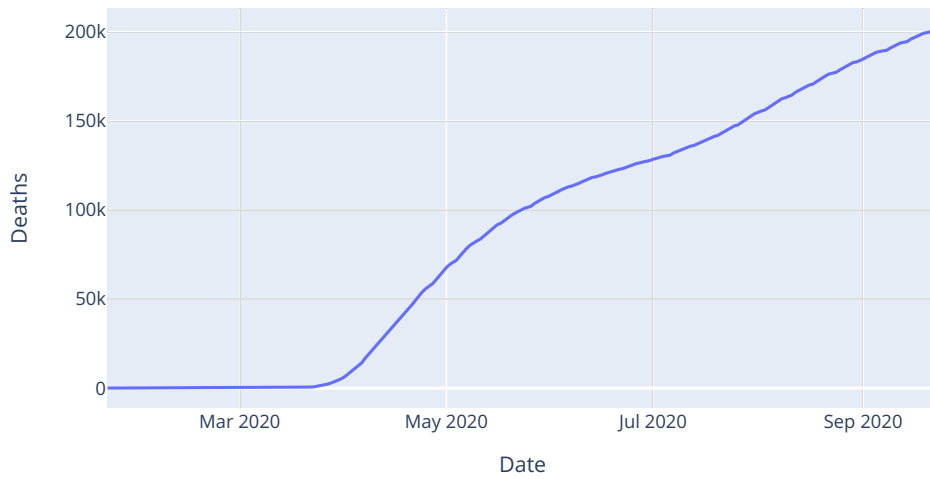
In [9]:

```
px.bar(df_US, x="Date", y="Deaths", color="Deaths", height=400)
```



In [10]:

```
px.line(df_US, x="Date", y="Deaths", height=400)
```



Confirmed vs Deaths

In [11]:

```
px.scatter(df_US, x="Confirmed", y="Deaths", height=400)
```

