In [27]:
```
import plotly.figure_factory as ff
from plotly.subplots import make_subplots
```

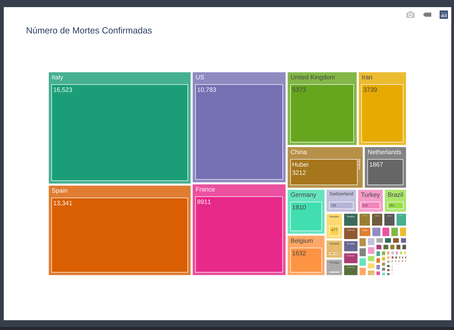
In [28]:
```
# Filtra os dados considerando o ultimo dia da base de dados
completo = df[df['Date'] == max(df['Date'])]
```

In [29]:
```
# Imprime as 5 primeiras linhas
completo.head(5)
```

Out[29]:

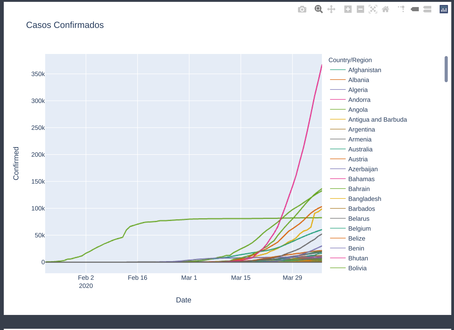
	Province/State	Country/Region	Lat	Long	Date	Confirmed	Deaths	Recovered	Active Cases
19575		Afghanistan	33.0000	65.0000	2020-04-06	367	11	18	338
19576		Albania	41.1533	20.1683	2020-04-06	377	21	116	240
19577		Algeria	28.0339	1.6596	2020-04-06	1423	173	90	1160
19578		Andorra	42.5063	1.5218	2020-04-06	525	21	31	473
19579		Angola	-11.2027	17.8739	2020-04-06	16	2	2	12

In [30]:
```
# Plota painel
fig3 = px.treemap(completo.sort_values(by='Confirmed', ascending=False).reset_index(drop=True),
                  path=['Country/Region','Province/State'],
                  values='Confirmed',
                  height=600,
                  title='Número de Casos Confirmados',
                  color_discrete_sequence = px.colors.qualitative.Dark2)
fig3.data[0].textinfo = 'label+text+value'
fig3.show()
# Plota painel
fig3 = px.treemap(completo.sort_values(by='Confirmed', ascending=False).reset_index(drop=True),
                  path=['Country/Region','Province/State'],
                  values='Deaths',
                  height=600,
                  title='Número de Mortes Confirmadas',
                  color_discrete_sequence = px.colors.qualitative.Dark2)
fig3.data[0].textinfo = 'label+text+value'
fig3.show()
```



Pico de Casos Confirmados e Mortes

In [31]:
```
# Por casos confirmados
fig4 = px.line(df_agrupado,
               x = 'Date',
               y = 'Confirmed',
               color = 'Country/Region',
               height = 600,
               title='Casos Confirmados',
               color_discrete_sequence = px.colors.qualitative.Dark2)
fig4.show()
# Por mortes confirmadas
fig4 = px.line(df_agrupado,
               x = 'Date',
               y = 'Deaths',
               color = 'Country/Region',
               height = 600,
               title='Mortes Confirmadas',
               color_discrete_sequence = px.colors.qualitative.Dark2)
fig4.show()
```



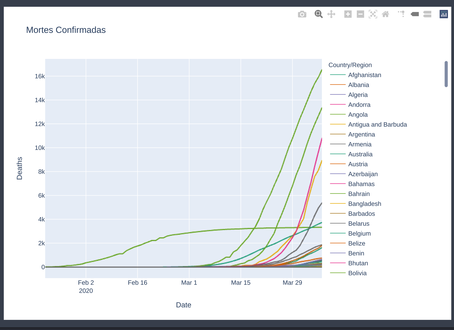


Gráfico com Folium

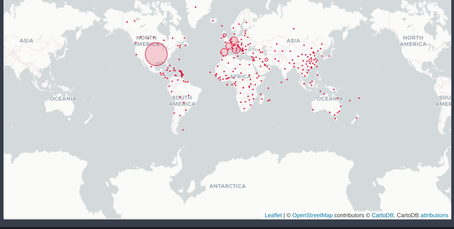
In [32]:
```
import folium
```

In [33]:
```
# Filtra os dados considerando o ultimo dia da base de dados
temp = df[df['Date'] == max(df['Date'])]
```

In [34]:
```
mapa = folium.Map(location=[0,0], tiles='cartodbpositron',
                  min_zoom=1, max_zoom=4, zoom_start=1)

for i in range(0, len(temp)):
    folium.Circle(
        location=[temp.iloc[i]['Lat'], temp.iloc[i]['Long']],
        color = 'crimson', fill='crimson',
        tooltip = '<li><bold>Country : '+str(temp.iloc[i]['Country/Region'])+
                  '<li><bold>Province : '+str(temp.iloc[i]['Province/State'])+
                  '<li><bold>Confirmed : '+str(temp.iloc[i]['Confirmed'])+
                  '<li><bold>Deaths : '+str(temp.iloc[i]['Deaths']),
        radius=int(temp.iloc[i]['Confirmed'])**1.1).add_to(mapa)
mapa
```

Out[34]:



In [ ]:

Pequena demonstração da pandemia global

Por: Leandro M. Torres