In [1]:
```python
import pandas as pd
```

Estudo Covid-19 PANDEMIA

In [2]:
```python
# Lendo o arquivo CSV e informando que há uma coluna do tipo Data-Tempo
file = 'covid_19_clean_complete.csv'
df = pd.read_csv(file, sep=',', parse_dates=['Date'])
```

In [3]:
```python
# Ignorando os warnings
import warnings
warnings.filterwarnings('ignore')
```

In [4]:
```python
df.head(10)
```

	Province/State	Country/Region	Lat	Long	Date	Confirmed	Deaths	Recovered
0	NaN	Afghanistan	33.0000	65.0000	2020-01-22	0	0	0
1	NaN	Albania	41.1533	20.1683	2020-01-22	0	0	0
2	NaN	Algeria	28.0339	1.6596	2020-01-22	0	0	0
3	NaN	Andorra	42.5063	1.5218	2020-01-22	0	0	0
4	NaN	Angola	-11.2027	17.8739	2020-01-22	0	0	0
5	NaN	Antigua and Barbuda	17.0608	-61.7964	2020-01-22	0	0	0
6	NaN	Argentina	-38.4161	-63.6167	2020-01-22	0	0	0
7	NaN	Armenia	40.0691	45.0382	2020-01-22	0	0	0
8	Australian Capital Territory	Australia	-35.4735	149.0124	2020-01-22	0	0	0
9	New South Wales	Australia	-33.8688	151.2093	2020-01-22	0	0	0

In [5]:
```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 19836 entries, 0 to 19835
Data columns (total 8 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Province/State  6080 non-null   object
 1   Country/Region  19836 non-null  object
 2   Lat             19836 non-null  float64
 3   Long            19836 non-null  float64
 4   Date            19836 non-null  datetime64[ns]
 5   Confirmed       19836 non-null  int64
 6   Deaths          19836 non-null  int64
 7   Recovered       19836 non-null  int64
dtypes: datetime64[ns](1), float64(2), int64(3), object(2)
memory usage: 1.2+ MB
```

In [6]:
```python
# Criando a coluna Casos Ativos = Casos Confirmados - Mortos - Casos Recuperados
df['Active Cases'] = df['Confirmed'] - df['Deaths'] - df['Recovered']
```

In [7]:
```python
# Subistituindo Mainland China por China na coluna Country/Region
df['Country/Region'] = df['Country/Region'].replace('Mainland China', 'China')
```

In [8]:
```python
# Preechendo Missing Values
df[['Province/State']] = df[['Province/State']].fillna('')
df[['Confirmed','Deaths','Recovered','Active Cases']]=df[['Confirmed','Deaths','Recovere
d','Active Cases']].fillna('')
```

In [9]:
```python
# Convertendo DataTypes
df['Recovered']=df['Recovered'].astype(int)
```

In [10]:
```python
# Cabeçalho após tratamento
df.head(10)
```

	Province/State	Country/Region	Lat	Long	Date	Confirmed	Deaths	Recovered	Active Cases
0		Afghanistan	33.0000	65.0000	2020-01-22	0	0	0	0
1		Albania	41.1533	20.1683	2020-01-22	0	0	0	0
2		Algeria	28.0339	1.6596	2020-01-22	0	0	0	0
3		Andorra	42.5063	1.5218	2020-01-22	0	0	0	0
4		Angola	-11.2027	17.8739	2020-01-22	0	0	0	0
5		Antigua and Barbuda	17.0608	-61.7964	2020-01-22	0	0	0	0
6		Argentina	-38.4161	-63.6167	2020-01-22	0	0	0	0
7		Armenia	40.0691	45.0382	2020-01-22	0	0	0	0
8	Australian Capital Territory	Australia	-35.4735	149.0124	2020-01-22	0	0	0	0
9	New South Wales	Australia	-33.8688	151.2093	2020-01-22	0	0	0	0

Examinando Dados Temporais

In [11]:
```python
df.Date.describe()
```

```
count                   19836
unique                     76
top       2020-01-25 00:00:00
freq                      261
first     2020-01-22 00:00:00
last      2020-04-06 00:00:00
Name: Date, dtype: object
```

Agrupando Dados

In [12]:
```python
# Obtem o número de casos fonfirmados, mortes, recuperados e ativos agrupado por data e p
or região
df_agrupado = df.groupby(['Date','Country/Region'])['Confirmed','Deaths','Recovered','Act
ive Cases'].sum().reset_index()
```

In [13]:
```python
df_agrupado
```

	Date	Country/Region	Confirmed	Deaths	Recovered	Active Cases
0	2020-01-22	Afghanistan	0	0	0	0
1	2020-01-22	Albania	0	0	0	0
2	2020-01-22	Algeria	0	0	0	0
3	2020-01-22	Andorra	0	0	0	0
4	2020-01-22	Angola	0	0	0	0
...	...	...	...	...	...	...
13979	2020-04-06	Vietnam	245	0	95	150
13980	2020-04-06	West Bank and Gaza	254	1	24	229
13981	2020-04-06	Western Sahara	4	0	0	4
13982	2020-04-06	Zambia	39	1	5	33
13983	2020-04-06	Zimbabwe	10	1	0	9

13984 rows × 6 columns

In [14]:
```python
# Odena o DataFrame por mais casos confirmados
df_agrupado.sort_values(by='Confirmed', ascending=False)
```

	Date	Country/Region	Confirmed	Deaths	Recovered	Active Cases
13971	2020-04-06	US	366614	10783	19581	336250
13787	2020-04-05	US	337072	9619	17448	310005
13603	2020-04-04	US	308850	8407	14652	285791
13419	2020-04-03	US	275586	7087	9707	258792
13235	2020-04-02	US	243453	5926	9001	228526
...	...	...	...	...	...	...
8756	2020-03-09	Mauritania	0	0	0	0
4967	2020-02-17	Zimbabwe	0	0	0	0
8754	2020-03-09	Mali	0	0	0	0
4968	2020-02-18	Afghanistan	0	0	0	0
0	2020-01-22	Afghanistan	0	0	0	0

13984 rows × 6 columns

In [15]:
```python
# Obtem o número de casos confirmados, mortes, recuperados e ativos agrupando por região.
df_group_paises = df.groupby('Country/Region')['Confirmed','Deaths','Recovered','Active C
ases'].sum().reset_index()
```

In [16]:
```python
# Ordena por países com mais casos fonfirmados
df_group_paises.sort_values('Confirmed', ascending=False)
```

	Country/Region	Confirmed	Deaths	Recovered	Active Cases
36	China	4683417	165756	2847170	1670491
171	US	2831915	64777	98681	2668457
84	Italy	1942859	206052	266247	1470560
156	Spain	1472568	123441	295573	1053554
65	Germany	1142172	12016	217756	912400
...	...	...	...	...	...
103	Malawi	19	0	0	19
166	Timor-Leste	16	0	0	16
181	Western Sahara	8	0	0	8
143	Sao Tome and Principe	4	0	0	4
155	South Sudan	2	0	0	2

184 rows × 5 columns

In [17]:
```python
# Agrupa quantidade de casos recuperados, mortes e ativos por data
temp = df.groupby('Date')['Recovered','Deaths','Active Cases'].sum().reset_index()
```

In [18]:
```python
# Remodela o dataframe com variável e valor para ter quantidades de recuperados, mortos e
ativos
temp = temp.melt(id_vars='Date', value_vars=['Recovered','Deaths','Active Cases'],
                 var_name='Case',value_name='Count')
```

In [19]:
```python
temp
```

	Date	Case	Count
0	2020-01-22	Recovered	28
1	2020-01-23	Recovered	30
2	2020-01-24	Recovered	36
3	2020-01-25	Recovered	39
4	2020-01-26	Recovered	52
...	...	...	...
223	2020-04-02	Active Cases	751644
224	2020-04-03	Active Cases	813507
225	2020-04-04	Active Cases	889225
226	2020-04-05	Active Cases	945742
227	2020-04-06	Active Cases	997225

228 rows × 3 columns

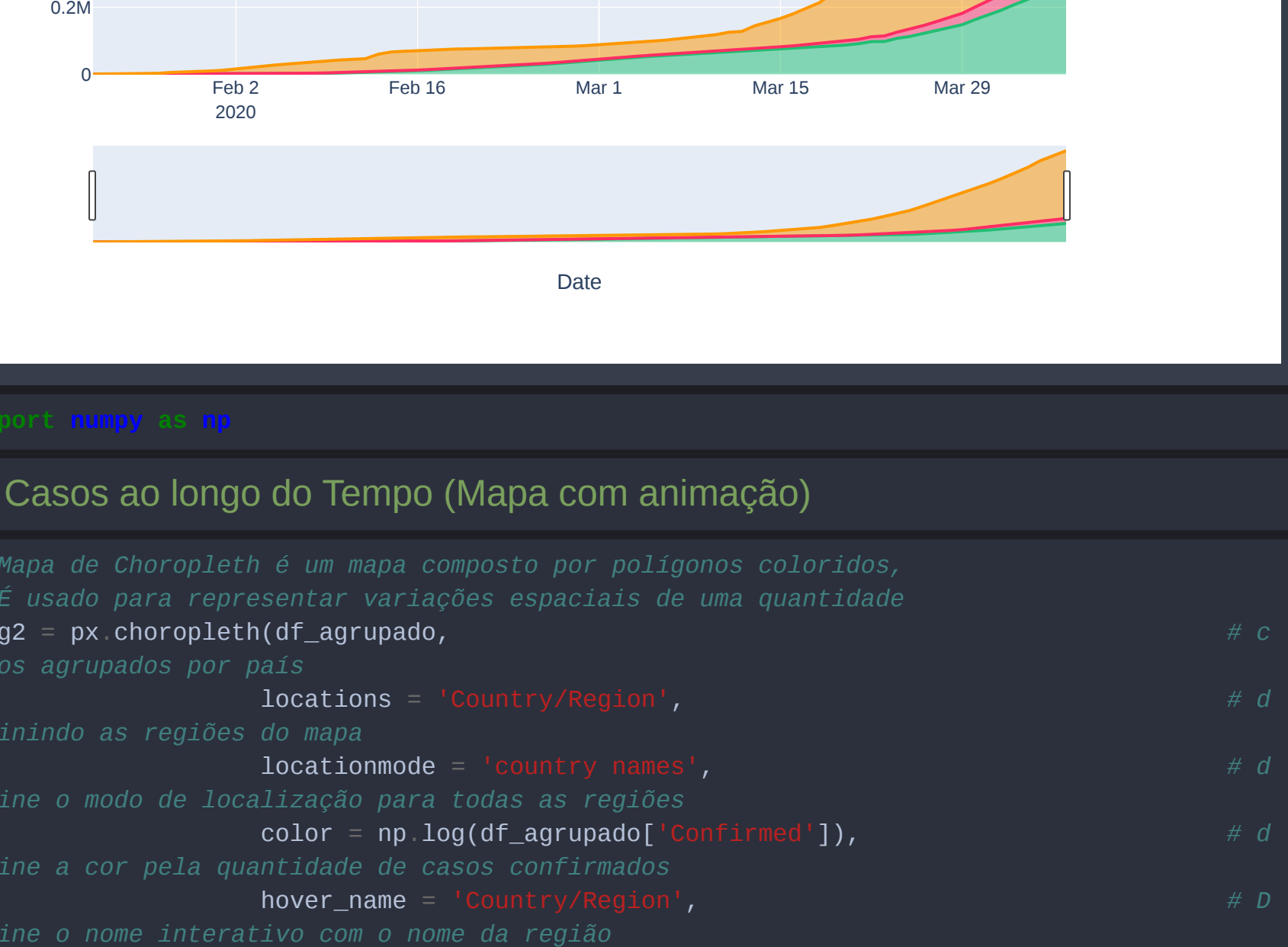
In [20]:
```python
# Habilita modo offline do plot
from plotly.offline import plot, iplot, init_notebook_mode
init_notebook_mode(connected=True)
```

In [21]:
```python
# Definindo o renderizador
import plotly.io as pio
pio.renderers
```

```
Renderers configuration
-----------------------
    Default renderer: 'plotly_mimetype+notebook_connected'
    Available renderers:
        ['plotly_mimetype', 'jupyterlab', 'nteract', 'vscode',
         'notebook', 'notebook_connected', 'kaggle', 'azure', 'colab',
         'cocalc', 'databricks', 'json', 'png', 'jpeg', 'jpg', 'svg',
         'pdf', 'browser', 'firefox', 'chrome', 'chromium', 'iframe',
         'iframe_connected', 'sphinx_gallery']
```

In [22]:
```python
# Cores
recuperados = '#21bf73'
mortes = '#ff2e63'
ativos = '#fe9801'
```

In [23]:
```python
import plotly.express as px
fig = px.area(temp,
              x='Date',
              y='Count',
              color='Case',
              height=600,
              title='Casos ao longo do tempo',
              color_discrete_sequence=[recuperados, mortes, ativos])
fig.update_layout(xaxis_rangeslider_visible=True)
fig.show()
```



In [24]:
```python
import numpy as np
```

Casos ao longo do Tempo (Mapa com animação)

In [25]:
```python
# Mapa de Choropleth é um mapa composto por polígonos coloridos,
# É usado para representar variações espaciais de uma quantidade
fig2 = px.choropleth(df_agrupado,                                                       # c
asos agrupados por país
                    locations = 'Country/Region',                                      # d
efinindo as regiões do mapa
                    locationmode = 'country names',                                    # d
efine o modo de localização para todas as regiões
                    color = np.log(df_agrupado['Confirmed']),                          # d
efine a cor pela quantidade de casos confirmados
                    hover_name = 'Country/Region',                                     # D
efine o nome interativo com o nome da região
                    hover_data = ['Confirmed', 'Deaths'] ,                             # D
efine o texto interativo com o número de confirmados e mortos
                    animation_frame = df_agrupado['Date'].dt.strftime('%d-%m-%Y'),     # D
efine o animate frame com as datas
                    title = 'Casos ao longo do Tempo',                                 # D
efine o título
                    color_continuous_scale = px.colors.sequential.Magenta)             # D
efine a paleta de cores
fig2.update_layout(autosize=False, width=800, height=600)                              # D
efine o tamnho da figura
fig2.show()                                                                            # E
xibe a figura
```

Mortes ao longo do Tempo (Mapa com animação)

In [26]:
```python
fig2 = px.choropleth(df_agrupado,                                                       # c
asos agrupados por país
                    locations = 'Country/Region',                                      # d
efinindo as regiões do mapa
                    locationmode = 'country names',                                    # d
efine o modo de localização para todas as regiões
                    color = np.log(df_agrupado['Deaths']),                             # d
efine a cor pela quantidade de Mortos
                    hover_name = 'Country/Region',                                     # D
efine o nome interativo com o nome da região
                    hover_data = ['Confirmed', 'Deaths'] ,                             # D
efine o texto interativo com o número de confirmados e mortos
                    animation_frame = df_agrupado['Date'].dt.strftime('%d-%m-%Y'),     # D
efine o animate frame com as datas
                    title = 'Mortes ao longo do Tempo',                                 #
 Define o título
                    color_continuous_scale = px.colors.sequential.Magenta)             # D
efine a paleta de cores
fig2.update_layout(autosize=False, width=800, height=600)                               #
 Define o tamnho da figura
fig2.show()
```