

# Global\_infected

September 25, 2020

```
[1]: import pandas as pd
import janitor

def get_data():
    URL_CSV="https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/
    ↪csse_covid_19_data/csse_covid_19_time_series/
    ↪time_series_covid19_confirmed_global.csv"
    df = pd.read_csv(URL_CSV)
    df['Country/Region'] = df['Country/Region'].replace({'US': 'United States of
    ↪America'})
    return df
df = get_data()
df
```

```
[1]: Province/State    Country/Region    Lat    Long    1/22/20    \
0      NaN    Afghanistan    33.939110    67.709953    0
1      NaN    Albania    41.153300    20.168300    0
2      NaN    Algeria    28.033900    1.659600    0
3      NaN    Andorra    42.506300    1.521800    0
4      NaN    Angola    -11.202700    17.873900    0
..      ...      ...      ...      ...      ...
261    NaN    West Bank and Gaza    31.952200    35.233200    0
262    NaN    Western Sahara    24.215500    -12.885800    0
263    NaN    Yemen    15.552727    48.516388    0
264    NaN    Zambia    -13.133897    27.849332    0
265    NaN    Zimbabwe    -19.015438    29.154857    0

    1/23/20    1/24/20    1/25/20    1/26/20    1/27/20    ...    9/15/20    9/16/20    \
0      0      0      0      0      0    ...    38815    38855
1      0      0      0      0      0    ...    11672    11816
2      0      0      0      0      0    ...    48734    48966
3      0      0      0      0      0    ...    1438     1483
4      0      0      0      0      0    ...    3569     3675
..      ...      ...      ...      ...      ...    ...      ...
261     0      0      0      0      0    ...    32250    33006
262     0      0      0      0      0    ...      10      10
263     0      0      0      0      0    ...    2016     2019
```

264	0	0	0	0	0	...	13819	13887
265	0	0	0	0	0	...	7576	7598

	9/17/20	9/18/20	9/19/20	9/20/20	9/21/20	9/22/20	9/23/20	9/24/20
0	38872	38883	38919	39044	39074	39096	39145	39170
1	11948	12073	12226	12385	12535	12666	12787	12921
2	49194	49413	49623	49826	50023	50214	50400	50579
3	1483	1564	1564	1564	1681	1681	1753	1753
4	3789	3848	3901	3991	4117	4236	4363	4475
..	...	...	...	...	...	...	...	...
261	33843	34401	35003	35686	36151	36580	37083	37591
262	10	10	10	10	10	10	10	10
263	2022	2024	2026	2026	2028	2028	2029	2029
264	13928	14022	14070	14131	14175	14389	14443	14491
265	7633	7647	7672	7683	7683	7711	7725	7752

[266 rows x 251 columns]

```
[2]: import janitor
import pandas as pd
import pandas_flavor as pf
import fbprophet
from matplotlib import pyplot as plt
from IPython.display import display, HTML

LISTA_COLUMNAS_A_BORRAR = ['Province/State', 'Country/Region', 'Lat', 'Long']

df = get_data()
df = df.remove_columns(LISTA_COLUMNAS_A_BORRAR)
df
df = pd.DataFrame( df.sum())
df.columns=['y']
df.index = pd.to_datetime(df.index)
df['ds'] = df.index
df = df.reset_index()

df = df.remove_columns(['index'])

df_prophet = fbprophet.Prophet(changepoint_prior_scale=0.15)
df_prophet.fit(df)
```

```

df_forecast = df_prophet.make_future_dataframe(periods=90, freq='D')
# Make predictions
df_forecast = df_prophet.predict(df_forecast)
df_forecast

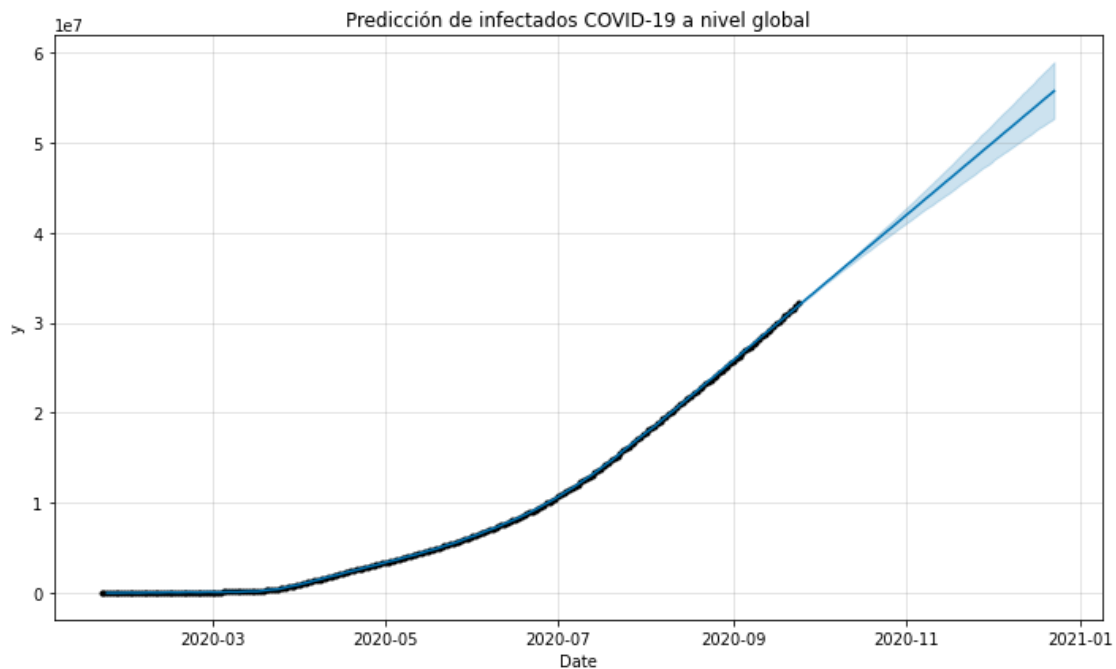
df_forecast = df_forecast[df_forecast["yhat"] >= 0]
df_forecast.loc[df_forecast.yhat_lower < 0, 'yhat_lower'] = 0

df_prophet.plot(df_forecast, xlabel = 'Date' )
plt.title('Predicción de infectados COVID-19 a nivel global' )

display(HTML(pd.DataFrame(df_forecast).to_html()))

```

INFO:numexpr.utils:NumExpr defaulting to 8 threads.  
 INFO:fbprophet:Disabling yearly seasonality. Run prophet with yearly\_seasonality=True to override this.  
 INFO:fbprophet:Disabling daily seasonality. Run prophet with daily\_seasonality=True to override this.  
 <IPython.core.display.HTML object>



```

[3]: df = get_data()
      df

```

```
[3]:
```

	Province/State	Country/Region	Lat	Long	1/22/20	\
0	NaN	Afghanistan	33.939110	67.709953	0	
1	NaN	Albania	41.153300	20.168300	0	
2	NaN	Algeria	28.033900	1.659600	0	
3	NaN	Andorra	42.506300	1.521800	0	
4	NaN	Angola	-11.202700	17.873900	0	
..	...	...	...	...	...	
261	NaN	West Bank and Gaza	31.952200	35.233200	0	
262	NaN	Western Sahara	24.215500	-12.885800	0	
263	NaN	Yemen	15.552727	48.516388	0	
264	NaN	Zambia	-13.133897	27.849332	0	
265	NaN	Zimbabwe	-19.015438	29.154857	0	

	1/23/20	1/24/20	1/25/20	1/26/20	1/27/20	...	9/15/20	9/16/20	\
0	0	0	0	0	0	...	38815	38855	
1	0	0	0	0	0	...	11672	11816	
2	0	0	0	0	0	...	48734	48966	
3	0	0	0	0	0	...	1438	1483	
4	0	0	0	0	0	...	3569	3675	
..	...	...	...	...	...	...	...	...	
261	0	0	0	0	0	...	32250	33006	
262	0	0	0	0	0	...	10	10	
263	0	0	0	0	0	...	2016	2019	
264	0	0	0	0	0	...	13819	13887	
265	0	0	0	0	0	...	7576	7598	

	9/17/20	9/18/20	9/19/20	9/20/20	9/21/20	9/22/20	9/23/20	9/24/20
0	38872	38883	38919	39044	39074	39096	39145	39170
1	11948	12073	12226	12385	12535	12666	12787	12921
2	49194	49413	49623	49826	50023	50214	50400	50579
3	1483	1564	1564	1564	1681	1681	1753	1753
4	3789	3848	3901	3991	4117	4236	4363	4475
..	...	...	...	...	...	...	...	...
261	33843	34401	35003	35686	36151	36580	37083	37591
262	10	10	10	10	10	10	10	10
263	2022	2024	2026	2026	2028	2028	2029	2029
264	13928	14022	14070	14131	14175	14389	14443	14491
265	7633	7647	7672	7683	7683	7711	7725	7752

[266 rows x 251 columns]

```
[4]: import janitor
import datetime
import numpy as np

def pipeline_populations():
    """ Cogemos un dataframe de poblaciones """
```

```

URL="https://population.un.org/wpp/Download/Files/1_Indicators%20(Standard)/
↳CSV_FILES/WPP2019_TotalPopulationBySex.csv"
THIS_YEAR = datetime.datetime.now().year

return (
    pd.read_csv(URL)
    .filter_on( f"" Time == {THIS_YEAR} & Variant == "Medium" "" )
    .select_columns(['Location','PopTotal'])
    .join_apply(lambda x: x['PopTotal'] * 1000 ,
↳new_column_name="PopMillions" )
    .remove_columns(['PopTotal'])
    .rename_column('PopMillions' , 'PopTotal')
    .transform_column('PopTotal',np.int64)
    .rename_column('Location' , 'Country/Region')
)

populations = pipeline_populations()
populations

```

```

[4]:
      70      Country/Region  PopTotal
      954      Afghanistan  38928341
      1838      Africa  1340598113
      1989      African Group  1338826591
      2140      African Union  1339423920
      ...      ...
      277315      World  7794798728
      278199  World Bank Regional Groups (developing only)  6528762227
      278350      Yemen  29825967
      279234      Zambia  18383956
      280118      Zimbabwe  14862927

[477 rows x 2 columns]

```

```

[5]: df = get_data()
df_country = pd.DataFrame()
df_country['Country/Region'] = df['Country/Region']
df_country['infected last_day'] = df.iloc[:, -1] - df.iloc[:, -2]
df_country.set_index('Country/Region')
df_country

```

```

[5]:
      Country/Region  infected last_day
0      Afghanistan      25
1      Albania      134
2      Algeria      179

```

3	Andorra	0
4	Angola	112
..	...	...
261	West Bank and Gaza	508
262	Western Sahara	0
263	Yemen	0
264	Zambia	48
265	Zimbabwe	27

[266 rows x 2 columns]

```
[6]: df_country_enrich = pd.merge(df_country, populations, on="Country/Region")
df_country_enrich['Infected/Million'] = 1000000 * df_country_enrich['infected_
↳last_day'] / df_country_enrich['PopTotal']
df_country = df_country_enrich
```

```
[7]: df_country.sort_values(by='Infected/Million', ascending=False).head(10)
```

[7]:	Country/Region	infected last_day	PopTotal	Infected/Million
140	Israel	7425	8655541	857.831995
168	Montenegro	289	628062	460.145654
119	France	27801	65273512	425.915492
19	Bahrain	687	1701582	403.741930
29	Brazil	66338	212559409	312.091572
6	Argentina	13467	45195777	297.970317
88	Costa Rica	1400	5094114	274.826987
92	Czechia	2910	10708982	271.734512
212	Spain	10653	46754782	227.848351
18	Bahamas	81	393248	205.976890

```
[8]: from datetime import datetime, timedelta
import seaborn as sns
from matplotlib import pyplot as plt
import matplotlib.dates as mdates

def pintar_grafico(df, array_naciones_pintar ,title):
    df = df.T
    df = df.iloc[1:]
    new_header = df.iloc[0] #grab the first row for the header
    df = df[1:] #take the data less the header row
    df.columns = new_header #set the header row as the df header
    df = df.iloc[2:]
    df.index = pd.to_datetime(df.index)
    df = df[array_naciones_pintar]
    df = df.iloc[:, : 9]
    chart_df = df
```

```

pd.plotting.register_matplotlib_converters()
chart_df.plot(legend=True,figsize=(13.5,9))

plt.gca().axis.set_major_formatter(mdates.DateFormatter('%B-%d'))
plt.gca().axis.set_major_locator(mdates.DayLocator(interval=15))
plt.xticks(rotation=45)

ax = plt.gca()

ax.set_title(title)
ax.set_ylim(ymin=0)

plt.show()

#df.tail(30).style.format ({ c : "{:20,.0f}" for c in df.columns }).
↪background_gradient(cmap='Wistia', )
return plt

```

```

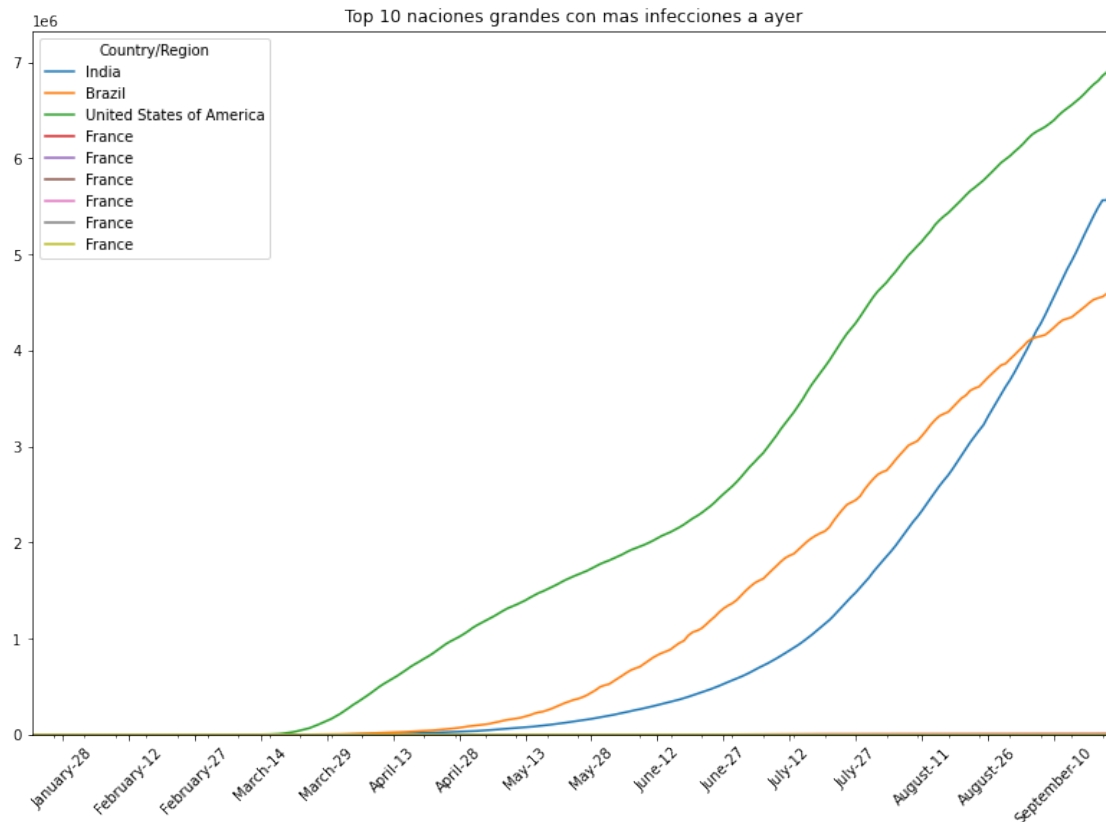
[9]: title="Top 10 naciones grandes con mas infecciones a ayer"
display(HTML(f"""<h1 id='{title}'>{title}</h1>"""))
df = get_data()

naciones_pintar = df_country.sort_values(by='infected last_day',
↪ascending=False).head(10)['Country/Region'].values
df_country.sort_values(by='infected last_day', ascending=False).head(10)

pintar_grafico(df,naciones_pintar,title)

```

<IPython.core.display.HTML object>



```
[9]: <module 'matplotlib.pyplot' from
      '/root/anaconda2/envs/jupyter/lib/python3.6/site-packages/matplotlib/pyplot.py'>
```

```
[10]: def pintar_grafico_daily(df, array_naciones_pintar ,title):
        df = df.T
        df = df.iloc[1:]
        new_header = df.iloc[0] #grab the first row for the header
        df = df[1:] #take the data less the header row
        df.columns = new_header #set the header row as the df header
        df = df.iloc[2:]
        df.index = pd.to_datetime(df.index)
        df = df[array_naciones_pintar]
        chart_df = df

        pd.plotting.register_matplotlib_converters()

        df_daily_increments = pd.DataFrame()
        for country in chart_df.columns:
            df_daily_increments[country] = chart_df[country].pct_change().
            ↪rolling(window=7).mean()
```



```

df_daily_increments

chart_df = df_daily_increments
chart_df.tail(45).plot(legend=True,figsize=(13.5,9))

plt.gca().xaxis.set_major_formatter(mdates.DateFormatter('%B-%d'))
plt.gca().xaxis.set_major_locator(mdates.DayLocator(interval=15))
plt.xticks(rotation=45)

ax = plt.gca()

ax.set_title(title)
ax.set_ylim(ymin=0)

plt.show()

chart_df.tail(30).style.format ({ c : "{:20,.2f}" for c in df.columns }).
↳background_gradient(cmap='Wistia', )

return chart_df

#pintar_grafico_daily(df,naciones_pintar,"Top 10 countries more cases, daily
↳percentage change")
df_country

```

```

[10]:
Country/Region  infected last_day  PopTotal  Infected/Million
0      Afghanistan          25  38928341      0.642206
1      Albania            134   2877800     46.563347
2      Algeria            179  43851043      4.082001
3      Andorra              0     77265      0.000000
4      Angola             112  32866267      3.407749
..      ...                ...      ...
241     Uzbekistan         559  33469199     16.701923
242  Western Sahara          0    597330      0.000000
243      Yemen              0  29825967      0.000000
244      Zambia             48  18383956      2.610972
245     Zimbabwe           27  14862927      1.816600

```

[246 rows x 4 columns]

```

[11]: #pintar_grafico_daily(df,naciones_pintar,"Top 10 countries more cases, daily
↳percentage change")

```

```

[ ]:

```