

Global_infected

November 13, 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
..      ...      ...      ...      ...      ...
264    NaN    West Bank and Gaza    31.952200    35.233200    0
265    NaN    Western Sahara    24.215500    -12.885800    0
266    NaN    Yemen    15.552727    48.516388    0
267    NaN    Zambia    -13.133897    27.849332    0
268    NaN    Zimbabwe    -19.015438    29.154857    0

    1/23/20    1/24/20    1/25/20    1/26/20    1/27/20    ...    11/3/20    11/4/20    \
0      0      0      0      0      0    ...    41728    41814
1      0      0      0      0      0    ...    21904    22300
2      0      0      0      0      0    ...    58979    59527
3      0      0      0      0      0    ...    4910     5045
4      0      0      0      0      0    ...    11577    11813
..      ...      ...      ...      ...      ...    ...      ...
264      0      0      0      0      0    ...    55408    56090
265      0      0      0      0      0    ...      10      10
266      0      0      0      0      0    ...    2063     2063
```

267	0	0	0	0	0	...	16661	16698
268	0	0	0	0	0	...	8410	8427

	11/5/20	11/6/20	11/7/20	11/8/20	11/9/20	11/10/20	11/11/20	11/12/20
0	41935	41975	42033	42092	42297	42463	42609	42795
1	22721	23210	23705	24206	24731	25294	25801	26211
2	60169	60800	61381	62051	62693	63446	64257	65108
3	5135	5135	5319	5383	5437	5477	5567	5616
4	12102	12223	12335	12433	12680	12816	12953	13053
...
264	56672	57226	57657	58158	58838	59422	60065	60784
265	10	10	10	10	10	10	10	10
266	2063	2067	2070	2070	2071	2071	2071	2071
267	16770	16819	16908	16954	16971	16997	17036	17056
268	8444	8471	8498	8531	8561	8610	8667	8696

[269 rows x 300 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

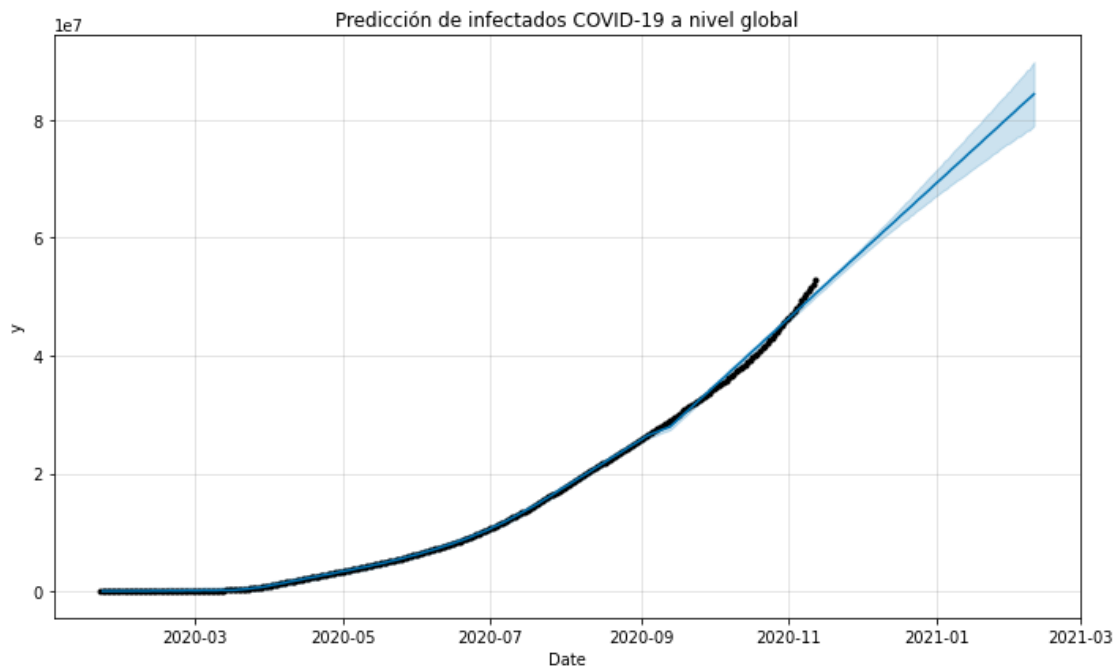
df_forecast = df_forecast[df_forecast["yhat"] >= 0]
df_forecast.loc[df_forecast.yhat_lower < 0, 'yhat_lower'] = 0

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	
..	
264	NaN	West Bank and Gaza	31.952200	35.233200	0	
265	NaN	Western Sahara	24.215500	-12.885800	0	
266	NaN	Yemen	15.552727	48.516388	0	
267	NaN	Zambia	-13.133897	27.849332	0	
268	NaN	Zimbabwe	-19.015438	29.154857	0	

	1/23/20	1/24/20	1/25/20	1/26/20	1/27/20	...	11/3/20	11/4/20	\
0	0	0	0	0	0	...	41728	41814	
1	0	0	0	0	0	...	21904	22300	
2	0	0	0	0	0	...	58979	59527	
3	0	0	0	0	0	...	4910	5045	
4	0	0	0	0	0	...	11577	11813	
..	
264	0	0	0	0	0	...	55408	56090	
265	0	0	0	0	0	...	10	10	
266	0	0	0	0	0	...	2063	2063	
267	0	0	0	0	0	...	16661	16698	
268	0	0	0	0	0	...	8410	8427	

	11/5/20	11/6/20	11/7/20	11/8/20	11/9/20	11/10/20	11/11/20	11/12/20
0	41935	41975	42033	42092	42297	42463	42609	42795
1	22721	23210	23705	24206	24731	25294	25801	26211
2	60169	60800	61381	62051	62693	63446	64257	65108
3	5135	5135	5319	5383	5437	5477	5567	5616
4	12102	12223	12335	12433	12680	12816	12953	13053
..
264	56672	57226	57657	58158	58838	59422	60065	60784
265	10	10	10	10	10	10	10	10
266	2063	2067	2070	2070	2071	2071	2071	2071
267	16770	16819	16908	16954	16971	16997	17036	17056
268	8444	8471	8498	8531	8561	8610	8667	8696

[269 rows x 300 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      Africa  1340598113
      1838  African Group  1338826591
      1989  African Union  1339423920
      2140  African Union: Central Africa  158619638
      ...
      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      186
1      Albania      410
2      Algeria      851

```

3	Andorra	49
4	Angola	100
..
264	West Bank and Gaza	719
265	Western Sahara	0
266	Yemen	0
267	Zambia	20
268	Zimbabwe	29

[269 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
156	Luxembourg	1508	625976	2409.038046
16	Austria	9262	9006400	1028.379819
169	Montenegro	644	628062	1025.376476
209	Slovenia	1931	2078932	928.842309
219	Switzerland	6924	8654618	800.035311
122	Georgia	3120	3989175	782.116603
89	Croatia	3082	4105268	750.742704
92	Czechia	7870	10708982	734.897117
154	Liechtenstein	28	38137	734.195139
7	Armenia	2132	2963234	719.484185

```
[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

```

```

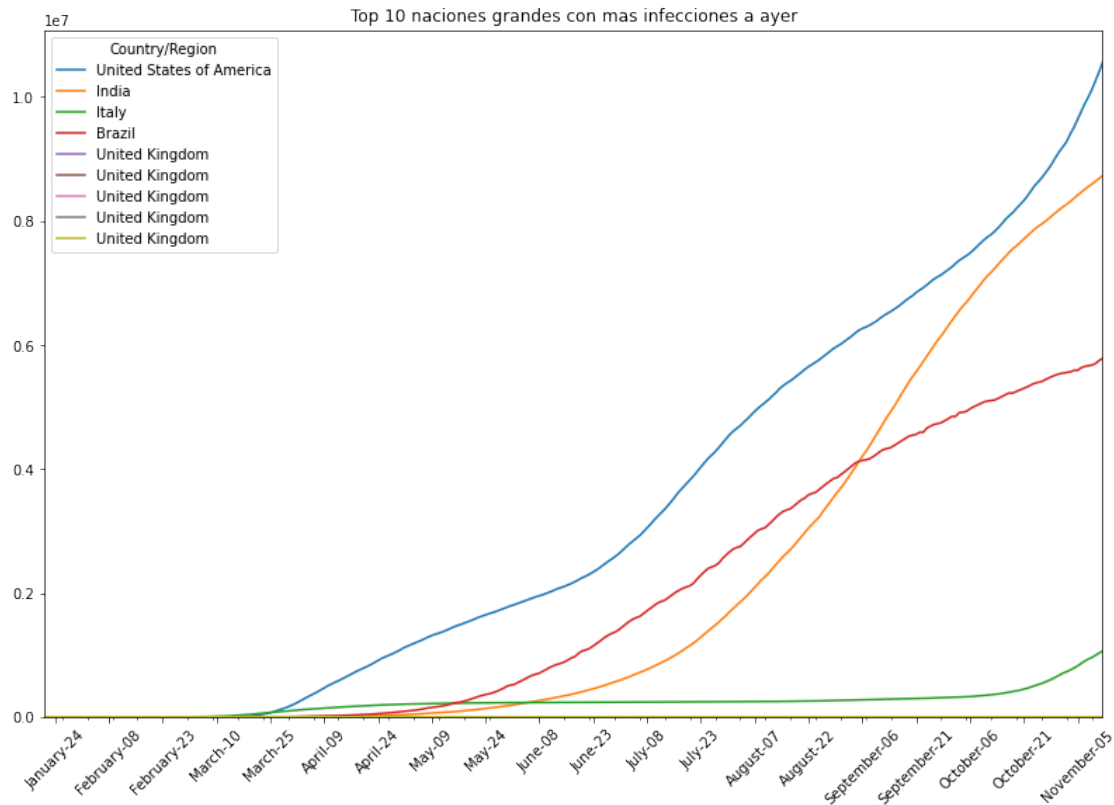
[9]: title="Top 10 naciones grandes con mas infecciones a ayer"
display(HTML(f"""<h1 id='{title}'>{title}</h1>"""))
df = get_data()

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	186	38928341	4.778010
1	Albania	410	2877800	142.469942
2	Algeria	851	43851043	19.406608
3	Andorra	49	77265	634.181065
4	Angola	100	32866267	3.042633
..
244	Vanuatu	0	307150	0.000000
245	Western Sahara	0	597330	0.000000
246	Yemen	0	29825967	0.000000
247	Zambia	20	18383956	1.087905
248	Zimbabwe	29	14862927	1.951163

[249 rows x 4 columns]

```

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

```

```

[ ]:

```