Global_infected

August 24, 2020

```python
[23]: 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()
```

```python
[24]: 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'])
```

```python
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()))
```
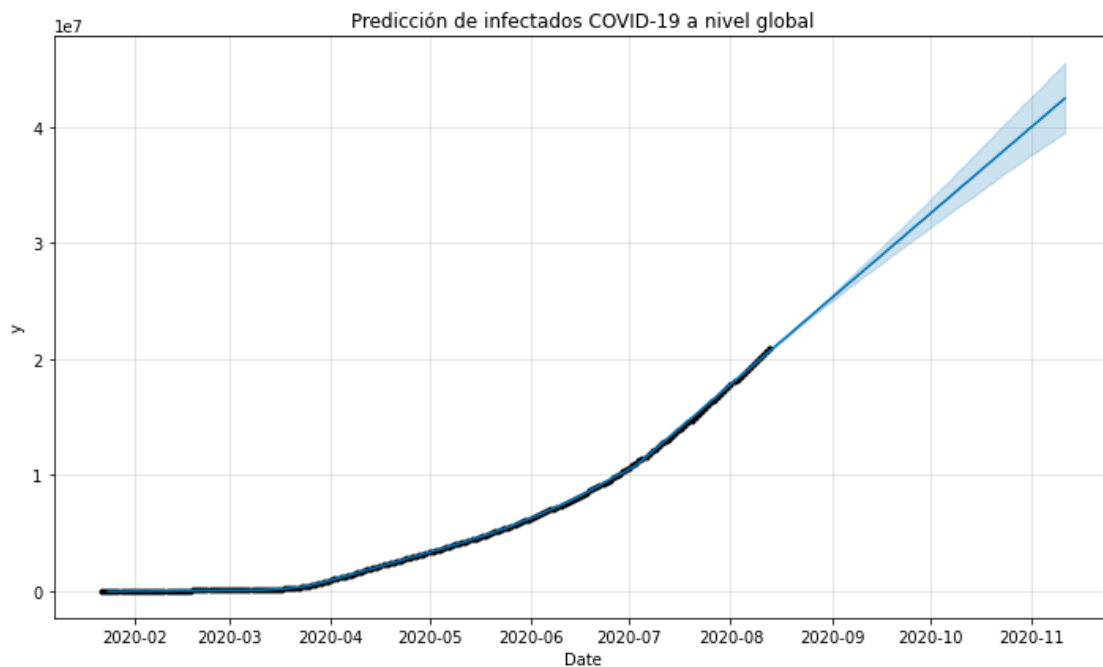
INFO:fbprophet:Disabling yearly seasonality. Run prophet with yearly_seasonality=True to override this.
INFO:fbprophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to override this.

<IPython.core.display.HTML object>

```
[25]: df = get_data()
```

```
[26]: 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
```

```
[26]:                                   Country/Region     PopTotal
      70                                    Afghanistan     38928341
      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]
```

```
[27]: 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
```

[27]:
```
       Country/Region  infected last_day
0         Afghanistan                 79
1             Albania                154
2             Algeria                488
3             Andorra                  4
4              Angola                 53
..                ...                ...
261  West Bank and Gaza              307
262      Western Sahara                0
263               Yemen                6
264              Zambia              162
265            Zimbabwe               97

[266 rows x 2 columns]
```

[28]:
```
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
```

[29]:
```
df_country.sort_values(by='infected last_day', ascending=False).head(10)
```

[29]:
```
               Country/Region  infected last_day     PopTotal  Infected/Million
136                     India              64553   1380004385         46.777388
29                     Brazil              60091    212559409        282.702141
225  United States of America              51443    331002647        155.415676
86                   Colombia              11286     50882884        221.803465
189                      Peru               8875     32971845        269.169044
212                     Spain               7550     46754782        161.480809
6                   Argentina               7498     45195777        165.900456
165                    Mexico               7371    128932753         57.169337
210              South Africa               3946     59308690         66.533252
138                      Iraq               3841     40222503         95.493809
```

[ ]:

[30]:
```
#df_country['Country/Region'].unique()
```

[19]:
```
#populations['Country/Region'].unique()
```

[31]:
```
from datetime import datetime, timedelta
import seaborn as sns
from matplotlib import pyplot as plt
```

```python
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]
    chart_df = df

    pd.plotting.register_matplotlib_converters()
    chart_df.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()

    df.tail(30).style.format ({ c : "{:20,.0f}" for c in df.columns }).
 ↪background_gradient(cmap='Wistia', )
    return plt
```
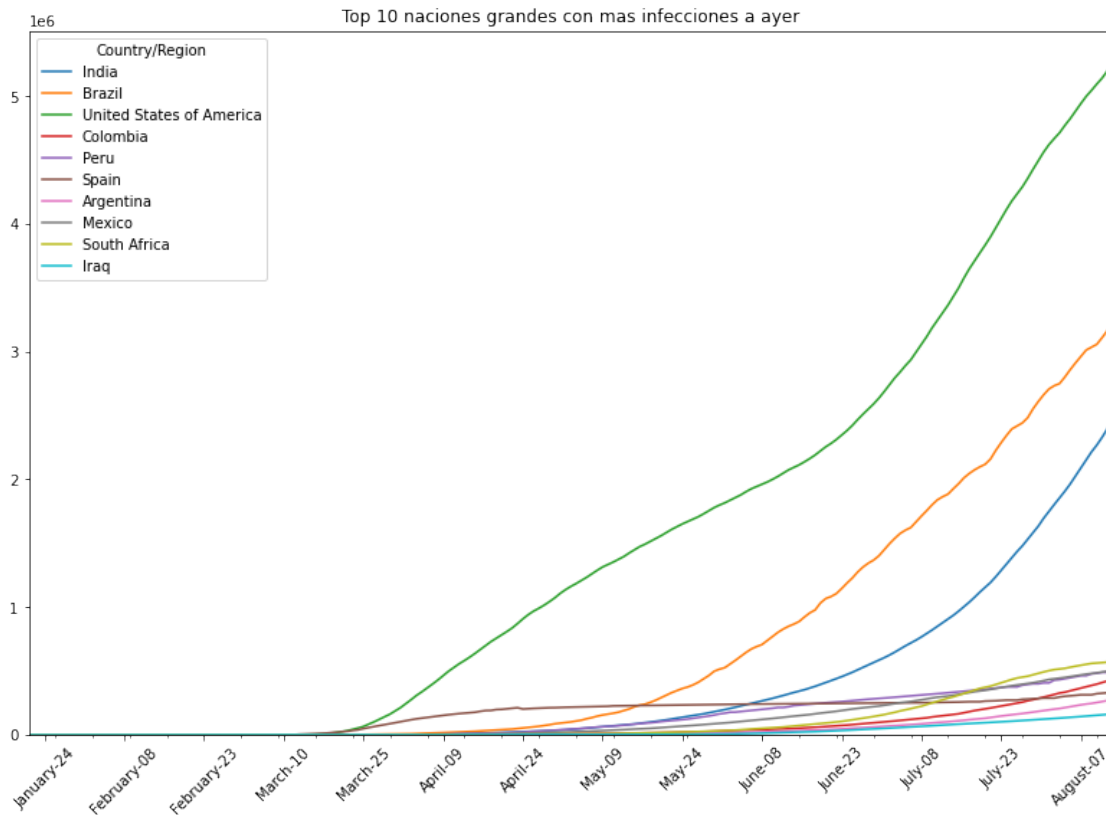
```python
[32]: title="Top 10 naciones grandes con mas infecciones a ayer"
display(HTML(f"""<h1 id='{title}'>{title}</h1>"""))

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>
```

Top 10 naciones grandes con mas infecciones a ayer

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

```python
[45]: 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")
```
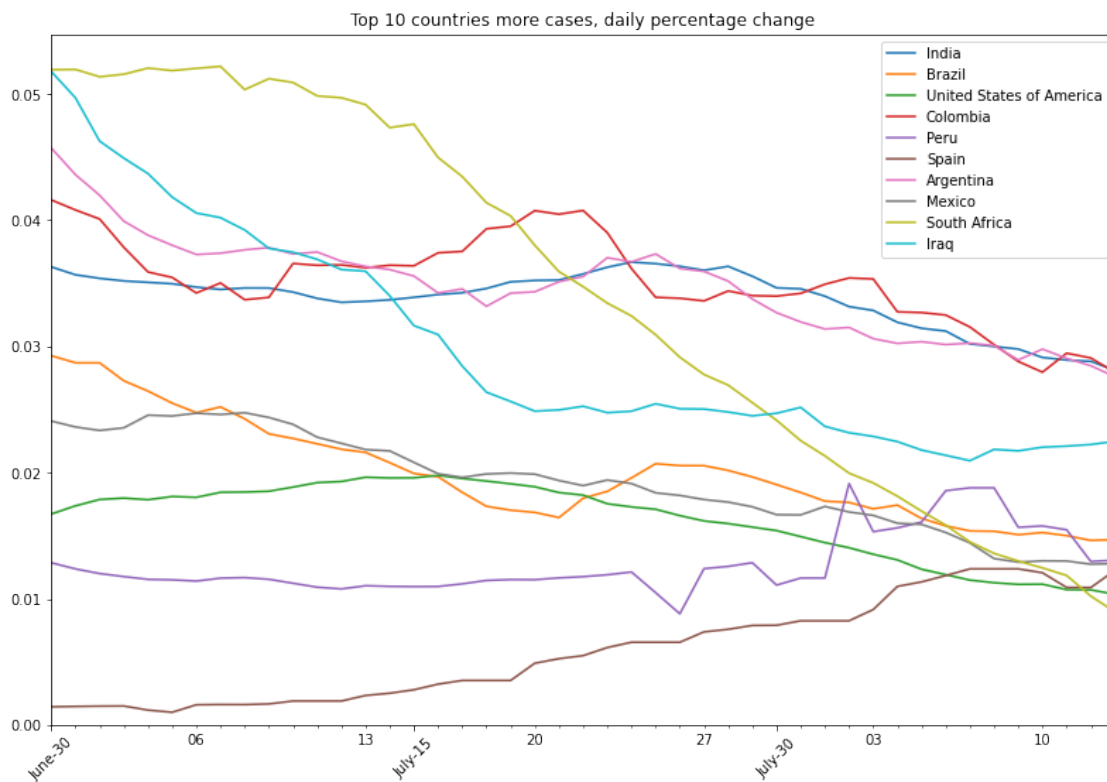

Top 10 countries more cases, daily percentage change

```
[45]:            India      Brazil  United States of America  Colombia       Peru  \
     2020-01-22      NaN         NaN                       NaN       NaN        NaN
     2020-01-23      NaN         NaN                       NaN       NaN        NaN
     2020-01-24      NaN         NaN                       NaN       NaN        NaN
     2020-01-25      NaN         NaN                       NaN       NaN        NaN
     2020-01-26      NaN         NaN                       NaN       NaN        NaN
     ...             ...         ...                       ...       ...        ...
     2020-08-09  0.029785    0.015084                  0.011148  0.028803   0.015665
     2020-08-10  0.029120    0.015252                  0.011163  0.027955   0.015776
     2020-08-11  0.028916    0.015009                  0.010731  0.029458   0.015472
     2020-08-12  0.028811    0.014629                  0.010712  0.029080   0.012960
     2020-08-13  0.028112    0.014686                  0.010358  0.027938   0.013065

                    Spain  Argentina    Mexico  South Africa      Iraq
     2020-01-22      NaN         NaN       NaN           NaN       NaN
     2020-01-23      NaN         NaN       NaN           NaN       NaN
     2020-01-24      NaN         NaN       NaN           NaN       NaN
     2020-01-25      NaN         NaN       NaN           NaN       NaN
     2020-01-26      NaN         NaN       NaN           NaN       NaN
     ...             ...         ...       ...           ...       ...
     2020-08-09  0.012376    0.028923  0.012907      0.012997  0.021723
     2020-08-10  0.012067    0.029780  0.013009      0.012449  0.022013
     2020-08-11  0.010904    0.029051  0.012996      0.011854  0.022099
     2020-08-12  0.010898    0.028465  0.012747      0.010217  0.022225
     2020-08-13  0.012259    0.027590  0.012795      0.008969  0.022481

     [205 rows x 10 columns]
```

```python
[43]: country='India'
      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
```

```
[43]:            India      Brazil  United States of America  Colombia       Peru  \
     2020-01-22      NaN         NaN                       NaN       NaN        NaN
     2020-01-23      NaN         NaN                       NaN       NaN        NaN
     2020-01-24      NaN         NaN                       NaN       NaN        NaN
     2020-01-25      NaN         NaN                       NaN       NaN        NaN
     2020-01-26      NaN         NaN                       NaN       NaN        NaN
     ...             ...         ...                       ...       ...        ...
     2020-08-09  0.029785    0.015084                  0.011148  0.028803   0.015665
     2020-08-10  0.029120    0.015252                  0.011163  0.027955   0.015776
```

```
2020-08-11  0.028916  0.015009            0.010731  0.029458  0.015472
2020-08-12  0.028811  0.014629            0.010712  0.029080  0.012960
2020-08-13  0.028112  0.014686            0.010358  0.027938  0.013065

              Spain   Argentina   Mexico   South Africa      Iraq
2020-01-22      NaN        NaN      NaN            NaN       NaN
2020-01-23      NaN        NaN      NaN            NaN       NaN
2020-01-24      NaN        NaN      NaN            NaN       NaN
2020-01-25      NaN        NaN      NaN            NaN       NaN
2020-01-26      NaN        NaN      NaN            NaN       NaN
...             ...        ...      ...            ...       ...
2020-08-09  0.012376   0.028923  0.012907       0.012997  0.021723
2020-08-10  0.012067   0.029780  0.013009       0.012449  0.022013
2020-08-11  0.010904   0.029051  0.012996       0.011854  0.022099
2020-08-12  0.010898   0.028465  0.012747       0.010217  0.022225
2020-08-13  0.012259   0.027590  0.012795       0.008969  0.022481

[205 rows x 10 columns]
```

[12]:
```
#title="Top 10 naciones grandes con mas infeccionesa ayer en infectados/millon"
#display(HTML(f"""<h1 id='{title}'>{title}</h1>"""))

#naciones_pintar = df_country.filter_on("PopTotal > 25000000").
 ↪sort_values(by='Infected/Million', ascending=False).head(10)['Country/
 ↪Region'].values
#df_country.sort_values(by='infected last_day', ascending=False).head(10)

#pintar_grafico(df_country,naciones_pintar,title)
```

[ ]: